/u/fortune* news

The Newsletter for Users of the Fortune 32:16 Computer — January 1987/Volume 4 Number 1

The Fortune:Word Glossary

The Green Box - Dealing with Glossary Errors

by Richard Alan Dow

Richard Alan Dow is owner of Dow Information Services which is a technical publications firm located in Rhode Island and is contributing to this publication independently.

You've just finished keying in a really great glossary, one that should save you a lot of time doing routine tasks in the future. Now for the moment of truth. You press the **CANCEL/DEL** and **EXECUTE** keys, wait out the "**(Verifying)**" message on the otherwise blank screen ... and then ... the dreaded green box appears, listing the number of errors detected and asking you just what you want to do about them!

No matter how long I work with **Fortune:Word** glossaries, I still get strong feelings of apprehension and insecurity when I see that green box. In fact, whether you are a novice or an old hand at glossaries, the problem usually isn't a major one, and a calm, organized approach can usually solve it quickly.

Just why the box, anyway? What does "verification" mean? Glossaries are different from other **Fortune:Word** documents because they are compiled each time you finish creating or editing them. That is, they are converted to binary form so that they will execute faster when you recall them while working on a document. During the time the screen is showing the message "**Verifying**," your Fortune System is in the process of compiling the glossary. And when you get a verification error message (the green box) it means that one or more problems have prevented the glossary from compiling properly.

What do you do? You cannot use the glossary until the errors are corrected, so the first step is to return to the glossary document (press the **RETURN** key). If you choose to save the problem for another day, you can press the **DELETE** key (the gray key, not the **CANCEL/DEL**) to return to the glossary menu.

Deal with the verification errors by looking for the simple mistakes first.

Back in the glossary document, go to the work page (**GOTO PAGE** and **w**) where you will find a listing of the errors which the system detected, including the page and line where they were found. If it's a long list, you might want to print a hard copy for reference as you check your glossary. You can't print the work page, but you can copy its contents to another page and print that.

See the inset box (on page 22) for a glossary that won't verify, with the kind of simple errors I usually make. Also, you will find a list of errors which would be found on the work page.

See **Glossary,** page 22

The BASIC Advisor

Ray Wannall is President of BaSiC Software Corporation in Balimore, MD, and is contributing to this publication independently.

This month we feature our second annual **IDOL/BAS** support company list. Actually, I prefer to call this a general Fortune Systems support company list, because most of the companies included here are involved in most aspects of support for Fortune users from hardware through all types of software.

In the process of compiling this listing I picked up on a few things which may be of interest to our readers. For example, costs for support have

See **BASIC Advisor**, page 2

In This Issue...

Laser Printers with the Fortune Computer...**Page 4**.

The Fortune-Mac Connection-Making /u/fortune news...**Page 8**.

UNIX Basics–Three more commands are explained...**Page 12**.

The Unix Directory–Part 6 in our series on shell scripts...**Page 16**.

The BASIC Advisor–Need service or support?...**Page 1.**

Fortune:Word Glossary–Dealing with verification errors...**Page 1.**

News from Fortune...Page 3.

*/u/fortune *news* is published by The Cambridge Consortium, Inc. It is not affiliated with Fortune Systems Corporation.

/u/fortune *news...*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz, Ph. D.

Technical Editor
Scott Bradner

Advertising Manager
Mark Palmerino

/u/fortune *news...* is published 12 times per year by The Cambridge Consortium, Inc. Neither /u/fortune news... nor The Cambridge Consortium are affiliated with Fortune Systems Corporation.

Subscriptions cost $48 for one year within the U.S.A. For foreign subscriptions, add $27 each year for added postage and handling costs.

Back issues are offered at:

Vol 1, Numbers 1 - 3	$1/each
Vol 1, Number 4	$.50/each
Vol 1, Numbers 5 - 8	$3/each
Vol 2, Numbers 1 - 7	$3/each
Vol 3, Numbers 1 - 12	$4/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $10 each. Please add $5 per order for foreign destinations.

Although every attempt is made to insure that our contents are error-free and without risk, The Cambridge Consortium is not responsible for losses that occur due to correct or incorrect interpretation of **/u/fortune** *news*.

For more information, call or write:

/u/fortune *news*
73 Jason Street
Arlington, MA 02174
(617) 663-7700, (617) 646-5603

The text for this newsletter was prepared using a Fortune 32:16 for text input. The information was then transferred to an Apple Macintosh Plus with kermit and formatted for typesetting using the Aldus Pagemaker program.

UNIX™ is a trademark of Bell Laboratories, Inc. Fortune 32:16™, Fortune:Word™ and FOR:PRO™ are trademarks of Fortune Systems Corporation. Multiplan™ and MS-DOS™ are trademarks of Microsoft. IBM-PC™ is a trademark of IBM. Macintosh Plus™ is a trademark of Apple Computer. Pagemaker™ is a trademark of Aldus Corporation.

©1987 by the Cambridge Consortium, Inc. 1/87

Basic Advisor

continued from page 1

tended to remain generally unchanged over the past year. Most of the companies reported good growth during this period with many branching into other hardware systems. Some, in fact, have dropped Fortune as a product but continue to support existing Fortune installations. A few indicated to me that they are looking into or have turned to other business software packages and database management systems for the Fortune users. (This could have something to do with the fact that **Concept Omega**, as of January 1, increased prices and lowered dealer commissions on **Thoroughbred** Software products.) Others seem to be faithful to Fortune and/or Thoroughbred and plan to stay with them for many years. If any of you end users enjoy shopping, this certainly seems to be the year for it.

I will present the companies this year as I did before, in random order with no editorial comments. Again, I suggest that you contact several before making any decisions. The variety of services and products may amaze you. Many have some inteesting proprietary products for sale. Some specialize in specific problem solving such as damaged file repair, communications links, custom modifications, and new software development. You may also find that one company is oriented to your vertical application needs than another.

KRETA TECHNOLOGIES, INC., So. Bound Brook, NJ. Contact Joseph Kreta at (201) 469-4347. $60.00 per hour billed in fifteen minute blocks.

WESTERN BLUE, San Mateo, CA. Contact Jackie Biggs at (415) 345-2583. $60.00 per hour billed in fifteen minute blocks with quarterly and yearly contract rates available.

TIM WAGNER (Independent), Columbus, OH. Contact Tim Wagner at (614) 291-3337. $50.00 per hour billed in fifteen minute blocks.

INTERNATIONAL SYSTEMS CORPORATION, Milwaukee, WI. Contact Dave Meister at (414) 327-5809. $89.00 per hour plus expenses billed in fifteen minute blocks.

UNI-KOMP, Houston, TX. Contact Dave Kloes at (713) 937-7782. $50.00 per hour.

BEACON SYSTEMS, Palm Springs, CA. Contact Mike Eisen at (619) 323-4555. $75.00 per hour billed in tenths of an hour.

SUPERIOR COMPUTER SYSTEMS, St. Louis, MO. Contact Steve Rosenfeld at (314) 837-7111. Support sold on a subscription basis, but $75.00 per hour provided for first time callers.

DAY PROM COMPUTER, INC., Dayton OH. Contact Judy Potter at (513) 299-8555. $60.00 per hour billed by the minute.

GILL & PIETTE, INC., Washington, DC. Contact Richard Gill at (202) 546-6170. $50.00 to $100.00 per hour with flat fee arrangements under some circumstances.

BaSIC SOFTWARE CORPORATION, Baltimore, MD. Contact Lynne Crawford at (301) 448-9460. $60.00 per hour billed in fifteen minute blocks.

HANDS COMPUTER SERVICES, Wheaton, IL. Contact Russell Hands at (312) 510-0088. $45.00 per hour with a $10.00 minimum.

COMPUTER SERVICES OF ATLANTA, Atlanta, GA. Contact Dana Klest at (404) 392-9938. $45.00 per hour billed in 6-minute blocks, fifteen minute minimum, with quarterly contract rates available.

Correction: In the December, 1986, issue of **/u/fortune news** (volume 3, number 12) we were discussing forcing a form feed at the end of your IDOL-defined reports by editing the program **CUTFIB**. You were told to add line 200 to the program as follows:

200 PRINT (6,ERR=201) <FF> <CR>

As any BASIC programmer worth his/her salt knows, the <FF> will create an error 20 when you try to type it in. The correct line of code to add is:

200 PRINT (6,ERR=201) 'FF' <CR>

We regret any confusion which may have been caused as a result of the typographical error. □

Re-dealing in Fortune Stock and Re-stocking Fortune Dealers...

News From Fortune Systems

To: JohnBass DMS

Fortune Systems Announces Stock Repurchase

Fortune Systems Corp. announced that its Board of Directors has authorized the repurchase from time to time of up to one million shares of its common stock. The Company may discontinue purchases of its common stock at any time that management determines additional purchases are not warranted.

"At recent public trading prices, we think the Company's stock is undervalued," stated president and chief executive officer **James S. Campbell**. "With a continued strong cash position and confidence that we have implemented the strategies necessary to position Fortune for growth in 1987, we believe it is appropriate to pursue such a program at this time."

Fortune Systems Acquires Equity Interest In Software System

Fortune announces that it purchased an equity interest in **MIT, Inc.** (Needham Heights, MA), owner of the 21-module **Manufacturing In Time** (MIT) software system. **Fortune Systems** will market and support the **MIT** package on the full range of **Fortune** products worldwide. Other terms of the purchase were not disclosed by either **Fortune Systems** or **MIT**, Inc.

First delivered in 1981, **MIT** is now installed and operating in a number of job shops, build-to-order and discrete manufacturers. The package was specifically designed for such users in the $1 million to $20 million annual sales range. **MIT** currently run under the **Qantel, Wang** and **UNIX** operating systems.

"We see a tremendous market among our dealers and end users for **MIT**," said **Fortune Systems** executive vice president, **Robert A. Davis**. "The sale of such turn-key manufacturing automation systems is expected to grow at a rate in excess of 32% per year through 1990, according to a recent study by Frost and Sullivan, Inc. **MIT** makes the kind of supermicrocomputers we sell more than competitive with several minicomputer-based systems in that market today. **MIT**'s 21 full integrated applications, its modular design and its attention to human engineering combine with **Fortune System**'s powerful office automation applications to deliver a comprehensive system spanning customer service, cost estimation, financial analysis, manufacturing control, production planning, and engineering."

Fortune Systems Signs New Dealers

Fortune announced that it has added two of its former dealers and one new dealer to its new master dealer network. According to executive vice president, **Robert A Davis**, this brings to 16 the number of master dealers that **Fortune Systems** has signed since October, nearly one third of the Company's June 1987 goal of having 50 master dealers in place across the country.

In early October, **Fortune Systems** cancelled all of its aggreements with its former dealers. The new master dealer agreement, according to Davis, is much more specific and demanding, on both sides, than the earlier dealer contracts.

The three new master dealers are: **Datamax Office Systems, Inc.**, of St. Louis, MO, for the state of Missouri; **Jones Business Automation Center**, of Charlotte, NC, for North and South Carolina (both are former **Fortune Systems** dealers); and **Computer Partners, Inc.**, of Boise, ID, for the State of Idaho. In total, the three new dealers have committed to purchases of Fortune Systems' 32:16 and new Formula systems of more than 6.4 million over the next three years.

Current List of Fortune Master Dealers

APEX Business Computers
New Orleans, LA 70112

ATEK Computer Distributors
Sacramento CA

Boston Business Computers Inc.
Needham Heights, MA 02194

Business Computer Systems
Oakbrook, IL 60521

CTD
Birmingham, MI

Computer Partners, Inc.
Boise, ID

Computer Sales Co, Inc.
Sante Fe Springs, CA 90670

Datamax Office Systems, Inc.
St. Louis, MO

JFM Business Systems
Chicago, IL

Jones Bus. Automation Center
Charlotte, NC

Micro Resources
Dublin, OH

New Jersey Bus. Computers
Rutherford, NJ 07070

QBC Systems Inc.
North Tonawanda, NY 14120

Ridding-Oullette, Inc.
Montreal, Quebec CANADA H3C-3R5

Sierra Systems
Sacremento, CA 95827

United Business Solutions, Inc.
Houston, TX 77092 □

Realists, Dreamers and Desk Top Publishing

Roger Wyse is President of InfoText, Inc. of New Fairfield and Danbury, Connecticut and is contributing to /u/fortune news independently. InfoText produces a product called Laser/TYPE.

Desk-top Publishing is rapidly becoming a bona fide buzzword. Just ask any author who has a desk-top laser printer about publishing and you will be lulled to sleep by opinion, technological definition, and reference. But, the question still remains, *What is Desk-top Publishing*? Thankfully, it's a good thing that the purpose of this article is to describe "How to" rather than "What is".

The fact remains that most businesses are purchasing desk-top laser printers for very simple reasons. Laser printers are fast, quiet, flexible, reliable, and inexpensive.

Some innovations fill a need so completely that they become widely accepted almost overnight. Electronic calculators have replaced slide rules and adding machines; word processing has relegated the typewriter to the most fundamental tasks in many offices. Electronic spread-sheets didn't replace a tool as much as they made possible a new way of thinking, analyzing and planning. So, what are laser printers and how will they impact our way of operating?

Laser printers are fast, flexible printers that use the *electro-photographic* process that's been common in office copiers since the 1950's. This is a non-impact printing process which is still black-and-white. These printers are page printers–which means that the information for at least one complete physical page must be prepared and stored before the printing process can be started.

So, what was the delay in the advent of the laser printer? The simple answer is probably the best. Do the words personal computer, cheap memory, floppy disk drives, winchester disk drives, and twenty years of research and development mean anything to you? Well, it may not be the whole answer–but it's close.

What's in a Name!

Laser printers derive their name from the process of character generation. A modulated laser beam is used to produce an electrostatic charge pattern that generates the desired information on the surface of a photoconductor drum. In laser printing, the photoconductor drum rotates with a constant speed and is positively charged by the charge corotron. The laser beam and laser optics focus onto a polygon mirror which directs the beam to the photoconductor drum which discharges the drum in response to the data buffered in the laser printer controller.

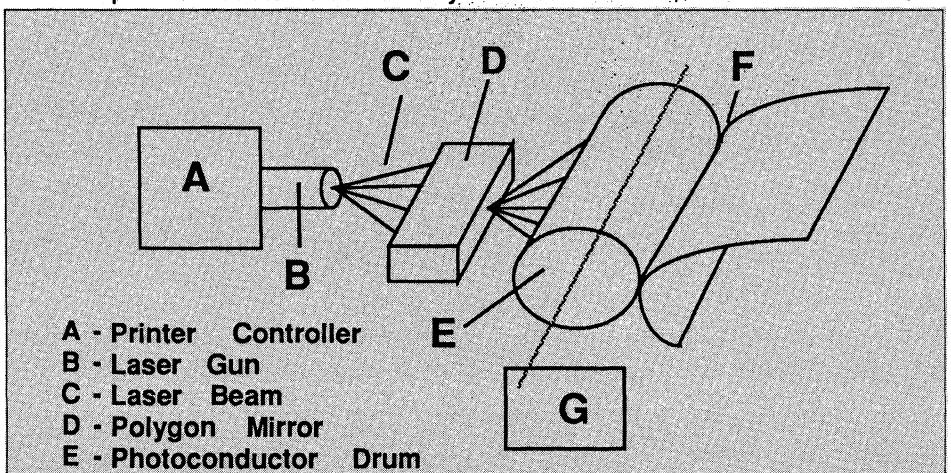
Positively charged toner is applied to the photoconductor drum in only those areas that have lost their positive charge by the action of the laser beam. A negative voltage causes the image to adhere electrostatically. At the fuser station, heat and pressure melt the print image into the paper. After the transfer process, light discharges the photoconductor drum and the remaining toner particles are removed in the brush cleaning system. In some "toner-drum-cartridge" systems this entire mechanism is changed every 3,000 to 5,000 pages and is mess free, so an office worker can do this quickly.

This printing process–from the electronic page preparation to the fusing of the printed image–will occur at a rate from between 6 pages per minute to as high as 26 pages per minute. However, there is more to selection than just the *speed* of printing. This brings us to the features, function, and benefits of laser printers.

Features, Functions and Benefits

A features, functions, and benefits

Continued on next page



Idealized schematic drawing of the basic components of a laser printer.

comparison is a starting point to sort out various laser printers. However, the laser printer itself is only 1/3 of the entire system. We must also consider the interface of the word processor to the laser printer. In our case, how does **Fortune:Word** facilitate the use of the laser printer features? Does the presence of more memory in the laser printer mean we can effectively use this additional memory with **Fortune:Word**? Will the laser printer allow us to use the graphics? If so, how? These and many other questions will be addressed in this article.

Chicken and Egg Problem

The broad range of features available in **Fortune:Word** are not directed toward laser printing functions, but rather toward inputting of information, manipulation of information (i.e., text editing), formatting, merging, records processing, etc. For example, *page* **makeup** with **Fortune:Word** is still a non-operator intervention process, whereby the header and footer (and soon fotnote) pages are merged together during the physical printing process. A page preview capability is simply not available on a standard Fortune Teminal. Also, this process is based on mono-spaced character characteristics. That is, each character takes up the same space as each other character. This creates a problem when we use a "page-printer" such as a desk-top laser that has the ability to change fonts. Different pitches, fonts, leading, and inter-character spacing (to mention a few) create a simple problem. What you see on the screen is not what you get when you print on a page-printer using these features. This makes editing text more difficult for the operators, not to mention the requirement of having to print a galley or pre-view copy of the document to check for page layout.

***(Editor's Note:** In all fairness we should point out that pre-view copies are realistically a necessity even with systems that provide **WYSIWYG** (what you see is what you get) capabilities. The fact of the matter is, it is almost always necessary to proof your work. However, **WYSIWYG** systems generally reduce the number of cycles that are necessary.)*

Laser printers derive their name from the process of character generation.

The features and functions of a laser printer/page printer were simply not available when **Fortune:Word** was programmed. The **Print Summary** menu is directed toward daisy-wheel printer technology. Proportional spacing has always been a feature on the **Print Summary** menu, but similar difficulties present themselves when we review the printed copy and the CRT version. Printing with proportionally spaced print wheel or font in **Fortune:Word** highlights some potential limitations, such as right-justified hyphenated text. Extreme difficulties present themselves when we use multiple fonts in a document, page, sentence, or word.

continued on next page

SST DATA INC.

Introduces

HANDSHAKE™ BLAZER

for high speed data communications. The newest member of the HANDSHAKE family offers error free transmission of binary and ASCII files between UNIX systems using low cost, dial-up lines. HANDSHAKE BLAZER achieves throughput speeds up to 9600 bps with these features:

* **End to End Error Correction** utilizing CRC algorithm
* **Auto Restart** permits restart from point of interruption
* **Send, Receive and Mail** supports commands from all nodes
* **User Friendly** menus, function keys Autodial, Autolog
* **Automatic Send Capability** with easy to use menus
* **Speed Optimizer** insures maximum speed compatible with line conditions
* **Audit Trail of File Transfers** complete history of transactions
* **Interactive Terminal Option** interactive mode also supports text file transfer

SST Data, Inc. 4701 W. Schroeder Dr. Milwaukee, WI 53223 (414) 355-6990 Telex 590597

How can "we" (**Fortune:Word** users) use the functions and features of laser/page printers with **Fortune:Word**? Well, I thought you would never ask! The laser printer is only 1/3 of the puzzle. The other two parts are the word processor and the interface software that addresses the laser printer controller features. With **Fortune:Word** as it now exists, how can we use the advanced features of a laser/page printer? There currently exist several "page description" languages that are **ASCII** based and can be used with excellent word processors such as **Fortune:Word**. For many years, the typesetting industry has had the capability of accepting **ASCII** based text with formatting commands to produce high quality output. The **UNIX** community for many years has been using **troff** and similar capabilities for that specific purpose.

*(**Editor's Note:** We used to produce/**u/fortune news** using **troff.** For more information on what we use to do and what we do now, see the article entitled "The Fortune-Mac Connection" in this issue.)*

With the new generation of laser/page printers, several new page description languages have been developed and are currently fighting to be accepted as a de facto standard for interfacing with desktop laser/page printers. Adobe Systems' **Postscript** (Page Description Language) and Imagen Corporation's **DDL** (Document Description Language) are two of the most well known languages that are currently being promoted as de facto standards. Each has its champion; Adobe has Apple Computer and Imagen has Hewlett Packard, Inc. So, the battle lines are drawn. Where does this leave **Fortune:Word** users who have neither **Postscript** or **DDL** capabilities?

Fortune:Word's Possibilities

In order to use an HP Laserjet, or similar laser printer with **Fortune:Word**, it is necessary to purchase the **Laserprinter Interface** from Fortune. This is a software product which makes it possible for your Fortune to communicate with the laserprinter. Once it is installed, you will be able to have your laserprinter emulate your letter quality printer, and also take advantage of several special fonts that are available in font cartridges. Typically the fonts supported are `courier`, which is like a typewriter, Times which is a newspaper font, and Helvetica, which is what use for /u/fortune. By modifying the information found in the /usr/lib/wheels files, you can use other fonts as well.

If you would like to produce more complex printouts, then read on to discover what the **Laser/Type** product from **InfoText** offers.

Laser/page printers have their own printer-contained software, called firmware, that resides on the controller of the printer. Some laser/page printer manufacturers have very well documented printer controller manuals for controlling the printer features. Hewlett-Packard has their **PCL** (printer command language) with its various levels defined for standard data processing to word processing

continued on next page

LASER/TYPE . . . THE FINAL TOUCH!

SAMPLES OF SOME FONTS AVAILABLE

HEWLETT PACKARD 92286T TAX 1 Font Cartridge

THIS IS HELVETICA 14 POINT BOLD FONT

THIS IS HELVETICA 12 POINT BOLD FONT

THIS IS HELVETICA 10 POINT BOLD FONT

THIS IS HELVETICA 8 POINT BOLD FONT

THIS IS HELVETICA 8 POINT MEDIUM FONT

THIS IS HELVETICA 8 POINT MEDIUM FONT

TAX LINE DRAW

HEWLETT PACKARD 92286B Font Cartridge

THIS IS HELVETICA 14.4 POINT BOLD FONT

THIS IS THE TMS RMN MEDIUM FONT

THIS IS THE TMS RMN BOLD FONT

THIS IS THE TMS RMN ITALIC MEDIUM FONT

THIS IS THE TMS RMN LIGHT 8 POINT FONT

These fonts and ***many more are all available*** within one document and on a single page. In addition, several different fonts can be selected at any given time for use within the Fortune:Word document. The selection of fonts is accomplished by using Fortune:Word glossary entries.

DESK-TOP PUBLISHING WITH FORTUNE:WORD

THIS ENTIRE PAGE WAS PRINTED FROM FORTUNE:WORD, LASER/TYPE AND AN HP LASERJET WITH A LASER/TYPE CONTROLLER

AT&T Computer Systems☐PENRIL Data Communications☐FORTUNE SYSTEMS☐HEWLETT-PACKARD Laser-printers☐
QMS Laser-printers☐ATI Laser-printers☐LASERPRO Laser-printers☐XEROX Laser-printers☐POST SCRIPT LASERS☐
DATA LANGUAGE CORPORATION - PROGRESS Fourth Generation Database Systems☐LEGAL MANAGEMENT SYSTEMS
DESKTOP PUBLISHING SOFTWARE & LASERS - WANG SYSTEMS☐SOFTEST INC. - SOFTYPE DESKTOP PUBLISHING☐

Examples of laser printer output from Laser/Type and the HP Laser Jet.

INFOTEXT INCORPORATED
217 CANDLEWOOD ISLE • NEW FAIRFIELD, CONNECTICUT 06812 • 203-746-4620
72 LAKE AVENUE • DANBURY, CONNECTICUT 06810 • 203-746-4620

across various printers, from daisy wheel, matrix, plotters, to page printers. **Fortune:Word** uses a small subset of **PCL**, with its laser printer software, when using the HP Laserjet printer. The commands used are only for font changing and do not include some of the more powerful capabilities. In addition, these commands or escape characters are actually hidden from the user with the use of special glossary that inserts selected foreign language symbols into the **Fortune:Word** document. Prior to the actual printing of the document, a formatting program is used, again hidden from the user to essentially globally replace the foreign language symbols with the proper PCL escape sequences. This process is used universally by many other program products that are interfacing with laser/page printers. The logic employed is no different than underlining or highlighting a string of characters. There is a beginning marker and an ending marker, that is to turn a characteristic on and turn it off. The background formatting program does the substitution in conjunction with the specific wheels file in the **/usr/lib/wheels** directory.

This process is limited and provides only for the substitution of one set of special characters for another. So, how can we use the more extended capabilities?

Some laser/page printers have extensive command languages which are printer specific and reside on the controller of the printer. The graphic primitives, font generation, line drawing, shading and other commands constitute a page description language not very different from the **Postscript** and **DDL** languages. As an example, we at **InfoText**, Inc., have a product called **Laser/TYPE**. We have modified the standard Hewlett-Packard laserjets to incorporate advanced printer controllers and integrate the standard **Fortune:Word** software and wheel files so as to take advantage of these advanced printer features.

Laser/TYPE Express Command Language, a proprietary English-like page composition language with a font generating system, simplifies page composition tasks. Access to fonts within a **Fortune:Word** document can be accomplished in several ways. The font generating system allows the user to vary typographical variations such as point size (from 2 to 50 points), character skewing, rotation and pitch, as well as separate x-axix and y-axis bolding through simple English commands that specify the font transformation parameters, thereby reducing the need for additional font cartridges. As many as one hundred fonts can be maintained in the laser printer memory and can be called for use within a **Fortune:Word** document.

Conclusion

Laser printers offer fast, quiet, reliable and flexible printing. More variety (e.g., italics, point size changes) can be attained with this technology. In addition, products are available that will work in conjunction with the powerful word processor **Fortune:Word** to allow high quality output. □

UNISOFT INC.

Experts in Computer Consulting - Sales - Service

Presents

FORTUNE SYSTEMS HELP

We Provide:

- Expert Trouble-shooting
- Years of programming and accounting experience
- *Free Initial Consultation*
- Training Available for:
  - * BASIC/IDOL/BAS and Multiplan
  - * UNIX and Ext. Fortune:Word
- Hardware Repairs & Upgrades
- FORTUNE trained experts at a cost that will surprise you!
- *Trade-Ins Welcome*
- Specializing in:
  - * Hard Disk Trade-In
  - * Hard Disk Upgrades

Authorized Fortune Master Dealer

Authorized Fortune Systems Repair Depot

CALL: (303) 671-8600 EMERGENCY HELP: (303) 691-HELP

UNISOFT INC. 14100 East Jewell Avenue Suite 27 Aurora, Colorado 80012

We have had numerous inquiries about...

The Fortune-Mac Connection

Over the last several months, you may have noticed a change in the appearance of **/u/fortune news** -- our typeface as changed, our layout is different, and the information block no longer says that we typeset it with **troff**. This is because we now produce the newsletter on a **Mac Plus** computer. When we made the switch, we included a short note explaining our reasons, primarily citing the "*right tool for the right job*" axiom. Since then, we have had numerous inquiries about how we connect a **Fortune** to a **Mac**, and why we decided to move (slightly) away from the Fortune for our production needs.

(As an aside, the "right tool for the right job" means that most of the day we spend banging on the Fortune terminals on our desks. We use a **Progress** database to maintain our subscription list, accounting information, phone lists, etc. We program, process words, send electronic mail, do remote logins--all with our Fortune. In a future issue, we will explain further how **The Cambridge Consortium** uses Fortune computers in our business.)

How we used to produce the newsletter

When we first started publishing **/u/fortune news**, we entered all of our text into our Fortune computer. Then we used a **UNIX** program called **troff** to format our text for a phototypesetting machine. The **troff** program is a powerful text formatter that takes care of font sizes, justification, hyphenation, etc. Once the text was formatted, the phototypesetter would change our articles into long columns of text, about 3 1/2 inches wide. The phototypesetter actually used little pictures of each letter which were then magnified and positioned by a series of complex lenses, and then exposed on photographic paper. Our direct cost for doing this was about $4.00 per page, or 12-inch column. Since we were only setting one column with the typesetter, and our pages were two columns, the cost would be about $8.00 for each finished page. (Of course, that figure doesn't include corrections, etc.)

Once the type was set in a single column, we would begin the process of laying out the newsletter. **The Glossary Entry** might begin on page one, and then continue on pages six, seven and eight. This involved cutting and pasting and retypesetting and repositioning, etc. until we would finally arrive at our final design. This was a laborious and expensive process.

At the beginning of last year, we obtained access to an **Imagen** laser printer. This was a huge breakthrough. With the **Imagen**, we could take our same documents, format them with **troff**, and obtain a printout of the article for about $.15 per page. The quality wasn't nearly as good as the phototypesetter, but at $.15 compared to $4.00 and the ease of use of the laserprinter (no chemicals, fonts to change, etc.), it seemed like the sensible route.

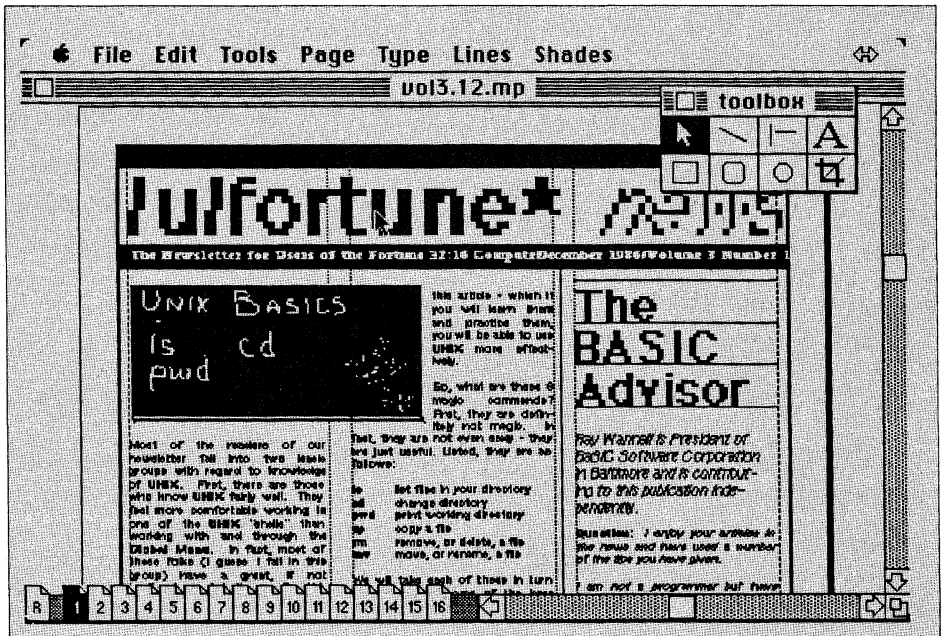
We were excited about laser printers and explored what we could do with just the Fortune and a laser printer.

Fortune and Laser Printers

Elsewhere in this issue, there is an article explaining how laser printers can be used with **Fortune:Word**. When you read that article and look at the examples, you will see that it is possible to create some pretty sophisticated output with the Fortune/laser combination. For correspondence and reports, it offers the flexibility of a dot-matrix printer together with the quality of a daisy-wheel printer. As a replacement for traditional printers, the laser printer offers many advantages.

However, it does not offer the

continued on next page



The above is a representation of how Aldus' Pagemaker looks on the Macintosh. With this program we can easily, and with more variety, design each page of /u/fortune news.

capability of producing sophisticated graphics, viewing them on the terminal screen and then printing them. Graphics were never Fortune's strong suit -- they have designed their equipment to be powerful and easy to use in more traditional arenas. When viewing text on the terminal, the user is restricted to the single font (or typestyle) that comes with the terminal; true "**W**hat **y**ou **s**ee **i**s **w**hat **y**ou **g**et" (**WYSIWYG,** pronounced wiss-ee-wig) formatting is not possible.

For certain applications, **troff** on the Fortune is a good choice. If you need to produce manuals or directories from a database, the formatting commands are relatively easy to insert; and the quality of the output will be good. One company that markets **troff** for the Fortune is **ETP Developers** in Portland, Oregon. Their phone number is 503 639-4024.

Enter Desktop Publishing and the Macintosh

Although **Fortune:Word** could produce an attractive newsletter, we felt that we would like more flexibility. A client of ours invited us to come to an **Apple/Pagemaker** demonstration session to see what was possible with a Macintosh. After a morning session, it was clear that the Macintosh would open up new possibilities for ease and flexibility with **/u/fortune news**. The Macintosh employs a bit-mapped display, which enables it to display graphics at fairly high resolution. Bit-mapped means that the computer has control of all of the individual dots, or pixels, on the screen. The Fortune uses a character display, which limits it to a fixed set of characters and graphic shapes.

The **Pagemaker** program is tailored to newsletter production. With it, you can start an article on page one, continue it on page four and finish it on page six. If you add more text to page one, it snakes the remainder of the article through all of the appropriate pages. If you decide you would like to drop in a chart or a picture, it can be easily imported from **MacDraw** or **MacPaint.** Headlines can be enlarged and reduced, fonts can be changed from one to another. Make this bold, make that underlined. Put this story on page two, that one on page one.

*We used to use **troff** to format the newsletter. With the advent of Pagemaker, we found that page layout and design became greatly simplified.*

Once we began using the Mac, complete pages (as opposed to single columns of text) could easily be produced on the laser printer and then just sent directly to our printer.

As you can imagine, this simplified our lives. Our razor blades and cans of glue have gotten buried at the back of my top left drawer. We have also been trying to improve the appearance and readability of the newsletter, which hopefully is apparent.

How we do it now

These days, we either type our articles directly into the Fortune, or have them electronically transferred from our guest authors. In either case, they end up on our trusty 32:16 and in **Fortune:Word**. We have attached our Macintosh to one of the ports on the Fortune, and using **MacKermit** we can sign into the Fortune and transfer the files from Fortune to Mac. **Microsoft Word** can read the files, although with a direct transfer like this, not only do we lose any formatting, but each line has a hard return at the end of it, so we lose the ability to word-wrap. To solve this, we have written a very short C program which erases these returns and just places them at the end of each paragraph. (If you are interested in obtaining a copy of this program, please call us.)

Once the articles are on the Macintosh, we format them with **Microsoft Word**, putting in the appropriate bold facing, fonts, etc. Finally, they are imported directly into **Pagemaker**, formatted as pages, and printed on an Apple Laserwriter.

Attaching a Mac to a Fortune

This process is really quite simple. If you are using a Mac Plus, you will need a wire with a 9-pin male round connector on one end (which goes into the *phone* connection on the back of the Mac) and a 25-pin male rectangular connector on the other end. Unfortunately, you may have to do this in two steps with 9-pin rectangular plugs in between. We use a mini DIN-8 (male) to DE-9 (female) (Part #590-0341). Then you'll need a DE-9 (male) to DB-25 (male) connector.

Next you'll need a communications program. As mentioned, we use **kermit** on the Fortune and **MacKermit** on the Mac. These are both available from us.

Finally, you need to configure a port on your Fortune for a terminal. This is exactly the same as you would do for any other terminal, although instead of entering Fortune as the terminal type, you will enter **Other** and then **vt100**. This is because the Mac emulates a **DEC VT100**.

Once that is done, if you fire up **MacKermit**, you should get the Fortune login **Type in your name and press RETURN**. If you do, you're all set. To actually transfer files, you'll have to have **kermit** running on the Fortune also. If you need help with this, please call us. A word of caution: since the **Global Menu** does not support terminals other than Fortunes, be sure to log into an account which will place you directly in the **UNIX** shell.

Josh Lobel

/u/help

Have you ever been faced with your terminal screen being full of garbage? Sometimes, after the running of a program, the terminal may not only be filled with extraneous material, but sometimes the cursor is at the top of the screen. What can be done about this?

Bob Ellsworth from the Compleat Winemaker (St. Helena, CA) wrote with the following advice:

In playing around in **UNIX**, I found that an easy way to clear the "**$**" or "**#**" at the top line is:

```
cd /bin
```
(pwd should read /bin)

```
echo "echo CTRL-L" > EXECUTE
chmod 775 EXECUTE
```

*What Bob is doing is creating a new shellscript which sends a **CTRL-L** (that is, hold the gray **CTRL** key down and simultaneously depress the **L** key) to your screen. This is accomplished by the **echo "echo CTRL-L"** part of his advice. The name of the new command is actually what is output by the **EXECUTE** key on your Fortune keyboard. The **> EXECUTE** (**EXECUTE** means to depress the **EXECUTE** key) is what names the command. Finally, you must make this command executable. The **chmod 775 EXECUTE** line does that. If you follow Bob's advice, you can just hit the **EXECUTE** key whenever you are at the **UNIX** shell and your screen will be cleared.*

Problems with cron

Dear /u/fortune news,

We believe we are having an unusual problem with our Fortune 32:16. After loading the disk from you containing **cron** we immediately experience intermittent system hangs. This occurred late in 1985 and after removal of **cron** and its associated files and directories the hangs ceased to occur entirely. We recently had reason to reinstall **cron** and try it again and within 24 hours all of our terminals hung. No error messages on the console, no pattern to hang ups. This occurred two more times forcing me to remove **cron** again. Do you know of any problems with **cron**?

*We do not know of any problems with **cron**. We have had **cron** installed on our systems for years and have never experienced the kinds of crashes you describe. If any other users have experienced this problem, please write us and tell us how you solved the problem. Without having your system in front of us it is hard to diagnose the problem. However, **cron** is a program that periodically runs other programs. Our first guess is that **cron** is running a program or programs that are causing the crash. You might inspect carefully the **crontab** file which exists in the **/usr/lib** subdirectory. What programs are being run? How often? Is the syntax for each line in the **crontab** file correct? However, barring some obvious problem with **crontab**, you will probably have to reinstall **cron** and then monitor the activity of your system carefully.*

A subscriber from GNS Systems, Inc. (E. Northport, NY) asks:

Would you know if anyone has done anything with regard to connecting the Fortune computer with cash registers?

*Not with cash registers per se. However, a user in the Massachusetts area has, in essence, made his terminal into a cash register. The basic tack he took was to control a cash drawer and a small printer from the ports on the back of a Fortune terminal. The cash drawer was controlled from the **auxilliary port** (he needed to custom build a device which would recognize the appropriate signal from a RS232 connection and then open the drawer). The printer was controlled from the **printer port**. Special software was written in **Business Basic** that controlled the entry of the sale information into the appropriate data structures and also open the cash drawer and print a sales receipt.*

The following is a response from Ray Esposito at The Brass Ring Society (Tulsa, OK) concerning a question Michael Smith had in last month's issue:

We use a QMS laser printer along with our NEC's. For the NEC's, the output command line is:

```
report to pipe "lpr -hF51 -p 3"
```

The **F51** is a length because we use unusual paper sizes and send the report to printer number 3. When we send only to the laser printer we use:

```
report to pipe "lpr -hp"
```

In this case, the laser printer is our default printer and a number is not needed. In both cases, the printing

continued on next page

works fine. I do not know if the quotes are the key but we have no problems.

In a third instance we use a program to allow us to select the printer:

```
acego -q pmen
echo "Which printer do you want to send
the output (1,2,3)?"
read p
case $p in
      "1") lpr -hp 1 tempfile;;
      "2") lpr -hp 2 tempfile;;
      "3") lpr -hF51 -p 3 tempfile;;
esac
rm tempfile
```

This third option may be of little use to Michael. This fits our needs as we create temporary files for review and printing out of **Fortune:Word**.

Hopefully, this will be of help to Michael Smith and other Fortune users who have had similar problems. Thanks, Ray.

Question: I recently spent several hours creating a **Multiplan** spreadsheet. I am very cautious about saving my work, so every 15 minutes or so I did the **Transfer Save** option to save the file on the hard disk. At the end of the session I quit out of **Multiplan** and everything seemed fine. The next morning I went to work on my file, and it was gone! Any ideas about what might have gone wrong.

Answer: Well, there is one likely possibility. If you happened to choose a name for your spreadsheet that was longer than 11 characters, and you have a version of **Multiplan** that is older than release **1.4**, your spreadsheet has fallen into a black hole. Why, you may ask. The answer is simple.

UNIX has certain restrictions on file names. The one that may have gotten you in trouble is a limit of 14 characters for a file name. All **Multiplan** files have a **".MP"** added to the end of their name so that they can be found easily. Suppose you named your file **4thfinancials**. That's 12 characters. When the **".MP"** is added on it exceeds 14 characters and the end of the file is truncated, so rather than the name becoming **4thfinancials.MP**, it ends up **4thfinancials.M**. Since it doesn't end in **.MP Multiplan** won't be able to find this file in its index, so as far as it knows, it's not there. Fear not, it is an easy problem to fix.

You'll have to leave **Multiplan** and enter the **UNIX** shell. The easiest way to do this is to type **!sh** from the **Global Menu**. This will give you a **$** or a **#** sign depending on who you are. Next you will look for your missing file by type the beginning of the file name followed by an asterisk, in our case we type:

```
ls 4thfina*  <CR>
```

As described in the **UNIX Basics** articles, this will list all files that begin **4thfina**. You only need to include enough of the file name to make it so that the list command doesn't list so many files that you can't find the one you seek. With any luck you will see **4thfinancials.M** listed there. In order to use it with **Multiplan**, you will need to rename the file with the following command:

```
mv 4thfinancials.M 4thfinan.MP  <CR>
```

The idea here is to pick a new name that is shorter than your bad name and is not a duplicate of any files you already have on your disk. If you are unsure about that latter point, use the **ls** command to look for the new file name, or use the **mv -i 4thfinancials.M 4thfinan.MP** version as described in **UNIX Basics** elsewhere in this issue. This command is very easy to use.

Once you've renamed the file, **Multiplan** will be able to find it and your troubles will be over.

If you have the version **1.4** of **Multiplan**, the program will protect you from choosing a name which is too long. □

Expand your Fortune® without spending a fortune.

New equipment costs big bucks. The Good Fortune Trader sells fully tested burned-in equipment, refurbished to Fortune® standards at prices that won't make waves with your cash flow. Call or write today for a listing of our specials.

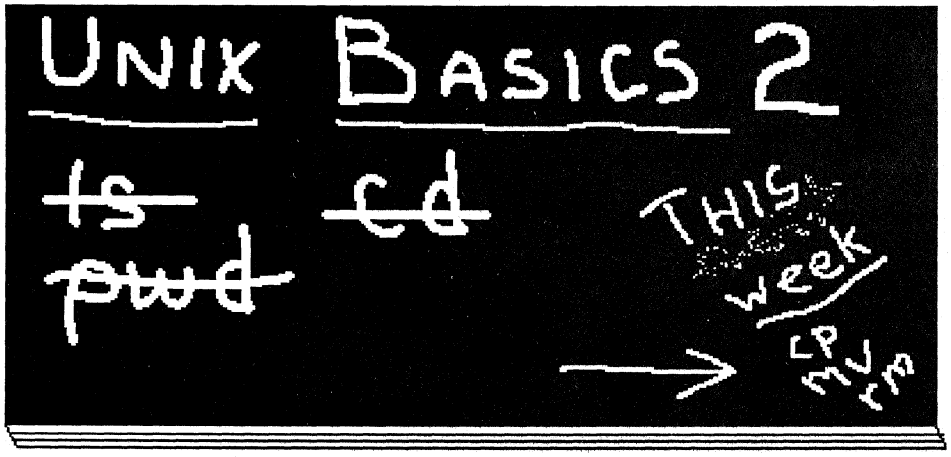
THE GOOD FORTUNE TRADER

60 DAY WARRANTY

a division of Computer Services of Atlanta

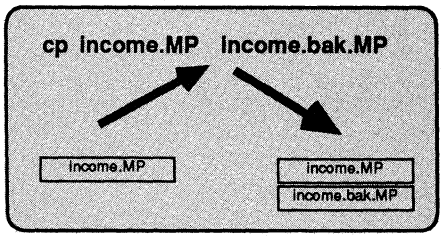
(404) 392-9938

1534 Dunwoody Village Parkway/Atlanta, Georgia 30338



Last month we began this series by introducing you to three basic **UNIX** commands. These commands were **ls**, **pwd** and **cd**. In that article, we also showed you how to go from the Global Menu to the **UNIX** shell. All of the commands that we talked about last month, and the commands that we talk about this month, are executed from the **UNIX** shell. However, they each have counterparts on the Global Menu. We hope you spent the time learning how to use the **ls**, **pwd** and **cd** commands last month and we hope you will join us for a basic look at three more useful **UNIX** commands. These commands are **cp** (for copying files), **mv** (for renaming or moving files) and **rm** (for removing or deleting files).

The *cp* command



Let's first look at the **cp** command. **cp** stands for "copy" and is the command you use for various activities. The most basic function of **cp** is to make a copy of a file. For example, suppose you had just finished creating a Multiplan file that contained some projections for next year's income and let's say this file was called **Income.MP**. Now, since this is an important file, you want to copy it so that if something were to happen to the original, you would still have something as a backup. The **cp** command is perfect for this maneuver. At the **UNIX** dollar sign prompt you might type the following:

```
cp income.MP income.bak.MP
```

This command would create a copy of the file **Income.MP** in the directory in which you are working and this copy would be called **Income.bak.MP**. After you issue the above **cp** command, you might wish to issue an **ls** command just to convince yourself that a new file has indeed been created.

Thus, one common use of **cp** is to create a copy of a file in the directory you are in. Another common use is to copy one or more files from the directory you are in to some other directory on the system. A good example of this is when you copy a file from your directory to the floppy. This is often done to make a backup of important files on a floppy so that if something were to happen to your hard disk, you would be able to recover your files. It is also useful to have backups on floppies in the event that you accidentally destroy the contents of the file. We will say more about that later.

Let's say we really want to protect those income projections and we decide to make a copy of the file **Income.MP** on a floppy. To do this we would first put a diskette in the floppy drive, shut the floppy drive door and then mount the floppy (**mount /dev/fd02 /f**). Assuming the floppy is mounted, we would use the **cp** command in the following fashion:

```
cp income.MP /f
```

This **UNIX** command directs the file **Income.MP** to be copied to the directory **/f**.

So far, then, we have seen **cp** used with two different forms. The first form was, syntactically,

cp *file1 file2*

where **cp** is used to create a copy (**file2**) from an original (**file1**).

The second form was

cp *file(s) directory*

where **cp** was used to create a copy of a file in another directory. Note that the copy of the file in this other directory would have the same name as the original file.

So, in our second example we would find a file called **Income.MP** on the floppy. (Again, the **ls** command could be used to verify this).

The differences in these forms are sometimes important. For example, if you had ten files in your directory and you wanted to make a copy of each one *in that same directory* (analogous to **cp Income.MP Income.bak.MP**), then you would have to type ten separate **cp** commands. On the other hand, if you wanted to copy all the files from your directory to the floppy, then you could type one **cp** command as follows:

```
cp * /f
```

The * is interpreted by the **UNIX** shell to mean all the files in your directory. That means you don't have to type out each individual file name - which should be appealing to most of you.

The * is called a *meta-character*. Meta-characters are characters that are interpreted by the **UNIX** shell in a special way. Whenever the **UNIX** shell sees a * it, in essence, puts in the place of the * the names of all the files in your directory. It is a place holder. Another way of thinking about the * is to think that it means "all the files in my directory." Thus, the above **cp**

continued on next page

command would be understood by the **UNIX** shell to mean copy (**cp**) all the files in my directory (*) to the floppy (**/f**).

There are actually two main uses of the * in the **UNIX** shell. When the * appears alone, as it does in the above command, it means "all the files in my directory." However, notice the following command:

```
cp *.MP /f
```

In this instance, the * is acting as a place holder again. However, the construction ***.MP** means "all the files that end with the three characters **.MP**." Since we know that all Multiplan files end with those three characters (**.MP**), this **cp** command can be interpreted to mean copy (**cp**) all the Multiplan files in my directory (***.MP**) to the floppy (**/f**).

As you might guess, the * can be a powerful place holder. The more you use the **UNIX** shell, the more you will become comfortable with using this meta-character.

There are some things of which to be wary when using **cp**, however. Keep in mind, for example, that **cp** will overwrite files. This means that if you type **cp file1 file2** and a file by the name of **file2** exists, **cp** will *silently* destroy the original contents of **file2** before copying **file1** into **file2**. This is a common way of losing files. There are several ways of protecting yourself against this possibility. These ways require the use of flags. For example, if you type:

```
cp -i income.MP income.bak.MP
```

and **Income.bak.MP** already exists, then **cp** will inquire whether you really want to overwrite the file **Income.bak.MP**. If you do want to overwrite it, you type a **y <CR>**. Another flag which controls whether a file is overwritten is the **-n** flag. Thus, if you were to type:

```
cp -n income.MP income.bak.MP
```

and **Income.bak.MP** exists, then **cp** will not overwrite **Income.bak.MP**. However, **cp** is totally silent about the fact that it did not overwrite **income.bak.MP** and this could cause problems if you thought you were successful in backing up **Income.MP**. Thus, the **-i** flag is the best flag for preventing accidental overwriting of files.

One common use of cp is to copy files from your directory to the floppy drive, mainly as a means of making a backup.

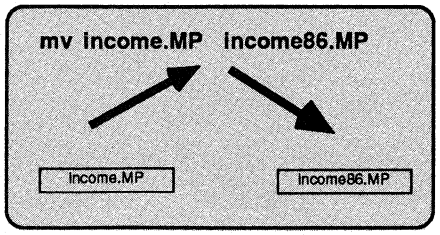
There are many other flags for **cp** and we direct you to the **Introduction to the FOR:PRO** book for more information on the flags. There is one other major use of **cp** that should be mentioned. This is the command that is primarily used for backing up the system. You may have seen, or even issued, a command like the following:

```
cp -Broust /dev/fd02 770 /
```

This is the command that will backup your entire system. Using **cp** in this capacity does not fall under the heading of "UNIX Basics", so we will not discuss it further, although it is also discussed in some detail in the **For:Pro** book.

As we explained last month, many **UNIX** commands will have their counterpart on the Global Menu. Such is the case with **cp**. For example, to make a copy of a file using the Global Menu, you would type **S1** and then **11**. You would then be prompted for the file you wanted to copy and the filename of the copy. Be careful, however, because this is like the unprotected **cp** command. In other words, you will overwrite anything in the file that you are copying to if a file by that name already exists.

The *mv* command



The **mv** command is used to move files from one place to another. This capacity is most often utilized to rename a file. For example, if you had a file called **Income.MP** that you created last year, you might decide this year that you want to rename that file as **Income86.MP**. To do that you would type:

```
mv income.MP income86.MP
```

This command is different from the **cp** command because after the **mv** command is executed the file **Income.MP** will no longer exist. You need to be careful with **mv** in the same way as with **cp**, since **mv** will normally overwrite the destination file if it exists. As with the **cp** command, the **-i** flag will guard against this inadvertent overwriting of files. Also, in a fashion similar to **cp**, you can say **mv *file directory*** and **mv** will move the named file from your current directory into the named directory. The file in the new directory will have the same name as it did in the old directory. In this case, the action of the **mv** command is much like its name, move, suggests. That is, you have moved the file from one directory to another while retaining its name.

The other form of the **mv** command is **mv *directory1 directory2*** and, as you may guess, this form moves all of the files in the first directory to the second directory. There is one limitation to this form, however. You can not move directories across a file system.

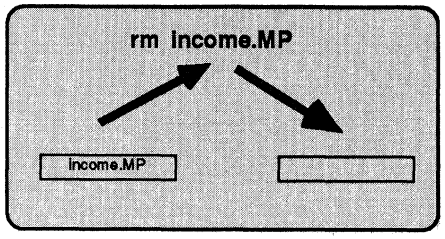
What does "across a file system" mean, you ask? Generally, file systems correspond to actual devices. Thus, the hard disk is usually mounted as one file system on the Fortune. When you mount the floppy drive to **/f**, you are actually attaching another file system. The **mv** command could not be used to move an entire directory from your hard disk to a floppy. To actually do this, i.e. move an entire directory including sub-directories to the floppy, you would have to do it in two steps. First you would have to use the **cp** command to copy all the files from the hard disk to the floppy.

continued on next page

Second, you would simply delete all the files from the hard disk.

Of course, the **mv** command can be issued from the Global Menu by choosing the **S1** screen and then typing either **16** or **26**. Choose **16** if you want to move a file and choose **26** if you want to move a directory.

The *rm* Command



The **rm** command, which stands for remove, is used to delete files. If for some reason you decided to remove your **income.MP** file, you would do it as follows:

```
rm income.MP
```

Naturally, removing a file is a *totally* irreversible action. Once it is done, there is no way to "undo" the remove. So, you must be careful and be absolutely sure that you really want to remove a file. A certain amount of precaution is afforded by the **-I** flag which acts similarly to the **-I** flag for the **cp** and **mv** commands. Thus, if you type:

```
rm -i income.MP
```

the Fortune would respond with:

```
rm: remove income.MP?
```

If you want to really delete **income.MP**, then type a **y** and then a **<CR>**. You can see that using the **-I** flag for each of these commands produces a much safer command in that you will help guard against the inadvertent destruction of a file or files. This is good. However, it is often the case that a user will not go to the extra trouble of typing the **-I** flag. Nevertheless, there is something that is often done on many systems. The system administrator can install something that we call "a safe rm command." For more information about this command see **Volume 3 Numbers 1** and **3** of **/u/fortune news**.

One other caveat about the **rm** command is in order. Be very careful when you use the * with the **rm** command. Remember that a * is interpreted by the **UNIX** shell as meaning all the files in your current directory. Thus, the following is a particularly hazardous use of **rm**:

```
rm *
```

since it will delete all of the files in the current directory. Of course, the best defense against wholesale accidental deletions of files are regular backups of your system. Finally, just for the record, the **rm** command can be accessed through the Global Menu by choosing option **13** from the **S1** screen.

Conclusion

In this article, we discussed the basics of three often-used **UNIX** commands. The first command was the **cp** command which is used to copy files. The second command was the **mv** command which is used to move files from one location to another. This command is most often used to rename files. The third command was the **rm** command which is used to remove or delete files.

These three commands, in conjunction with the **ls** command, the **cd** command and the **pwd** command discussed last month, comprise six fundamental **UNIX** commands. If a user learns and becomes proficient in their use, then he or she will have a good basis for interacting with the Fortune computer. We welcome your comments or criticisms on this article. Tell us what we did well and what we didn't do so well. In addition, if you would like to see some more information on certain **UNIX** commands, please write us.

Mark Palmerino

Vantine Computer Services

Professional Solutions for Business Microcomputer Users

Sales Service Consulting
Software Customization Support
Software/Hardware Upgrades

Over Three and a half years supporting Fortune Users

Vantine Computer Services
70 Walnut Street, Wellesley, MA 02181
(617) 239-8084
Authorized Fortune Dealer

** Don't Miss Out On Some NEW Offerings **

SOFTWARE FOR THE FORTUNE 32:16

MULTIPLAN LINK - can pass data from BAS/IDOL/BASIC files to Multiplan and back. For example, it could be used to send data from any of the BAS files (Chart of Accounts, Customer Master, Inventory Master, etc). Allows selection of fields to be sent and provides logical retrieval and key range selection. **$295.00**

RECORDS PROCESSING LINK - can pass data from BAS/IDOL/BASIC files to Records Processing and back. For example, it could be used to take customer address data from the Customer Master and will automatically make the Records Processing List document for mailouts. Allows selection of fields to be sent and provides logical retrieval and key range selection. **$295.00**

ASCII LINK - can pass data from BAS/IDOL/BASIC files to ASCII and back. For example, it could be used to send data from any of the BAS files (Chart of Accounts, Customer Master, Inventory Master, etc). Allows selection of fields to be sent and provides logical retrieval and key range selection. **$295.00**

TERMINAL/PRINTER LINK - provides capability to print Multiplan, and BAS/IDOL reports on a printer connected to a workstation view any report on the terminal before printing, convert any report to FortuneL:Word format; accumulate multiple BAS/IDOL reports before printing, backup reports onto diskette; copy or rename report files; print multiple copies of the report on either the primary and/or terminal printer. Also has option to print BAS/IDOL reports on HP LASER printer ($50 extra). ** ***Also supports printing off auxiliary workstation with Fortune:Word*** ($50 extra) **. **$195.00**

TELEPHONE LINK - on line telephone message, telephone directory and inter-office mailbox/memo system which can also be used as a simple calendar reminder system. Provides control, printing and viewing of on-line data for each user. **$295.00**

FINANCIAL LINK - stores BAS Income Statement and Balance Sheet data in files. Once stored, it can be sent to Multiplan or instantly printed. **$195.00**

MENU LINK - provides software and documentation for adding items to the Global Menu as well as a sub-menu for your use Includes backup script for user directories and BAS/IDOL/BASIC programs and data. **$95.00**

CALCULATION LINK - provides programs for amortization, depreciation, loan repayment, averaging (with graph), linear correlation, and breakeven analysis. **$95.00**

KOMPACT PERSONNEL ACCOUNTING - provides for data capture of personal, wage/salary, job, education, salary/wage, appraisal training, dental, medical and life insurance data. Includes over 80 pre-defined reports/worksheets. **$495.00**

CRT LINK - provides information for using other terminals on your Fortune (WYSE, TV, IBM 3101). **$95.00**

AUTOMATIC REPORT LINK - provides capability to print numerous IDOL defined reports automatically based on frequency or application without having to select each from a different menu. You identify frequency/application codes. **$95.00**

PRINTER LINK - provides printcaps for Okidata 84, 2410, TI 855, and Anadex 6500. Includes shell scripts for changing pitch and LPI. Allows these printers to be used in word processing. **$95.00**

MODEM LINK - allows Hayes 1200 Smartmodem or compatible to be used on one port for both incoming and outgoing calls. **$95.00**

INSTALL LINK - documentation and software example for installing your product/menu through "S5". **$95.00**

REPORT LINK - modifies IDOL Report Generator Program to allow stacking of up to 8 fields in an IDOL defined report. Also corrects problem with displaying subtotals for all stacked fields. **$95.00**

WESTERN LINK - provides instructions on how to use your Fortune for sending/receiving telex messages using Western Union's Easylink. Can replace your telex! Fortune ITE software is required. **$95.00**

UNI-KOMP 9803 Rocktree Houston, Texas 77040 (713) 937-7782

The UNIX Directory

Introduction to Shell Scripts - Part 6

by David E. Kloes

David E. Kloes is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the Fortune Computer, and is vice-president of the Houston UNIX User's Group. He contributes independently to ***/u/fortune*** *news.*

In part 5, we completed our discussion of the **until do done** command sequence and discussed the topic of executing multiple shell scripts. We also talked about two methods of passing data (variables) from one shell script to another. Part 6 will discuss some finer points about shell programming that we had not mentioned; and, so, we will continue with our topic of the last issue.

First of all, there is one point that we need to expand on from our discussion of variables. Whenever you use a variable name in combination with regular text, a good rule to follow is to always enclose the variable name in quotes. Look at example 1.

In that example, we set the variable **prefix** equal to **/u/**. If we echo the variable either with or without quotes by itself, it echoes correctly. If we echo the variable in combination with text (davek), nothing is echoed. This is because the shell treats **prefixdavek** as the name of the variable and since we have not established such a variable, nothing is echoed. If we enclose the variable name in quotes, the resulting echo line is correct. Quotes are handled differently, depending on which version of **UNIX** you use. However, if you follow the above rule, you will stay out of trouble, regardless of which version of **UNIX** you are on.

Since we are using more complex examples of shell scripts, there is another point we need to discuss. Whenever we use the **echo** and **read** statements to get input from a user, we need to include logic in our programs to handle the possible

Continued on next page

Example 1:

```
$ prefix="/u/"
$ echo "$prefix"
/u/
$ echo $prefix
/u/
$ echo $prefixdavek
        <==== nothing is echoed!
$ echo "$prefix"davek
/u/davek <==== echoes correctly
```

Example 2:

```
finish="n"
while test "$finish" = "n"
do echo -n "Enter name of file(s) to backup: "
   read filenames
   if test -f $filenames
      then cp -routVB /dev/fd02 790 $filenames
   else echo -n "No permission or file does not exist.
Depress <RETURN> to continue: "
        read garbage
   fi
done
```

Example 3:

```
finish="n"
while test "$finish" = "n"
do echo -n "Enter name of file(s) to backup (or 'exit'):
"
   read filenames
   if test "$filenames" = "exit"
      then exit
   fi
   if test -f $filenames -a "$filenames" != ""
      then cp -routVB /dev/fd02 790 $filenames
   else echo -n "No permission or file does not exist.
Depress <RETURN> to continue: "
        read garbage
   fi
done
```

things a user might do. Let's look at example 2.

There are two operator input possibilities that we have not programmed for here. Can you figure out what they are?

One is that we have not given the user an option to exit the program. What if this were a "backup" selection on a menu that they selected accidently and they do not want to perform the action? Or what if after selecting the option, they realized they did not have a formatted diskette? In this situation, the user might also depress the **<RETURN>** key without entering any filename(s). This second action would result in the user seeing our error message, but they would also see the shell syntax error *"test: argument expected"*. In some cases, this might result in the user dropping directly into the shell! Good shell scripts should have the logic to handle both of these situations. Let's look at an enhanced version displayed in example 3.

Here we have changed the echo prompt to include an "exit" option and we have entered the logic to "exit" if the user enters the word "exit". The line that begins "`if test -f`" tests to see if the name(s) the user enters are valid file names (**-f**). It also now includes the "and" (**-a**) condition which also tests to see if the user hit the **<RETURN>**. If the user does hit the **<RETURN>** key by itself, the value of the variable **"filenames"** would be equal to **""** and thus would fail the test. Now the only error message the user will see is our error message.

Another item we need to mention has to do with the use of *meta-characters* especially when used with the **echo** command. Whenever we include a **UNIX** meta-character (i.e. **$**, **`** , *****,etc) in an **echo** line, it should be preceded with a **** (backslash) character so that the shell will not treat it as a meta-character. Consider the extreme example shown in example 4.

In this example, the quotes around **cost** did not display; we got a shell error message; the **$** in front of 295 did not display; and the word **program** does not appear. The line that was echoed does not look anything like what we wanted. This is because the shell was interpreting the special characters as meta-characters. For example, it thought we wanted to display *the result of* the **program** command. (remember what the backquotes mean to the shell??). By using the backslash character in front of shell meta-characters, we are telling the shell to treat the character that follows literally as the character and not the meta-character. Example 5 looks more like what we wanted.

Example 4:

```
$ echo "The "cost" of the `program` is $295.00"
program: not found
The cost of the   95.00.
```

Example 5:

```
$ echo "The \"cost\" of the \`program\` is \$295.00."
The "cost" of the `program` is $295.00.
```

Example 6:

```
Program 1:   main.sh                          Program 2:
export.sh

name="David"; export name                     echo "$name"
echo "$name"                                  name="Sharon"
export.sh                                     echo "$name"
echo "$name"
```

Here we have told the shell to treat certain characters (**"**,**`**, and **$**) literally so they are not responded to as meta-characters.

We now continue on with our discussion of *passing variables* from the last issue. Remember we said that variables can be passed from one shell script to another by either **exporting** them or by passing them as arguments ($1-$9). First of all, an important point needs to be reemphasized and expanded. We stated that variables that have been exported are only available for use by the current shell script and to subsequent shell scripts. *Even though we export a variable, however, any changes to the "value" of the variable are not recognized by previous shell scripts.* This is an important point so we provide example 6 to demonstrate:

When **main.sh** is run, the variable **name** is set to "David" and exported. The first **echo** in **main.sh** would echo "David". The **export.sh** program is then run and the first **echo** would echo "David" since this is the value we passed to the program. We then set **name** equal to "Sharon" and **export.sh** correctly echoes "Sharon". The **export.sh** program is then exited and the second **echo** line in **main.sh** is executed. We would expect this **echo** line to echo "Sharon" since this is the value we gave it in **export.sh**. What is echoed; however, is "David". This demonstrates the point that value changes to variables in subsequent shell scripts are not recognized by the calling, or original, shell script.

What if we absolutely must know the value of what is entered in the **export.sh** shell script? Where there's a will, there's a way, they say. As is usually the case with **UNIX** (why do you think it has become so popular?), there is a way. Example 7 shows you one "way":

The **main.sh** shell script sets **name** to "David" and exports it. The **export.sh** script is then executed and sets **name** equal to "Sharon". The **export.sh** script then echoes the value of **name** and redirects the

See **UNIX Directory**, page 19

MAKE YOUR FORTUNE GROW

One of the greatest assets of your Fortune Systems 32:16 is its built-in ability to expand to match your need.

Whether you want to increase your number of workstations, upgrade system capability, or tie-in your Fortune Systems cluster to a mainframe, Micro Resources can give you the prompt, complete service you need.

As *the* Fortune Systems Master Dealer in Ohio, we have the hardware and software you need available for immediate shipment. From Fortune:Link to uninterruptible power supplies, from storage expansion systems to Development Utilities, we have the exact blend of hardware, software and systems expertise needed to make your Fortune Systems 32:16 do exactly what you want it to in the most reliable and cost-efficient manner possible.

And our genuinely friendly staff programmers and systems analysts are available to provide you with whatever additional support you may need, from software consultation and customization to integrated systems design and specification.

So whether you need a quick question answered or a new service agreement, Business Applications Update or just some networking cable, call Micro Resources.

We'll help your Fortune Grow.

Micro Resources

535 Metro Place South Dublin, Ohio 43017 (614) 766-2335

UNIX Directory

continued from page 17

output to the file **/tmp/tmpfile** instead of to the terminal screen. Now that the variable value has been sent to a **UNIX** file, the name variable can be captured with the name=`cat /tmp/tmpfile` command which says: set the variable **name** to the results of the **cat /tmp/tmpfile** file which holds the value of what the variable was set to in the **export.sh** script. The value of the variable at the end of the **main.sh** script is now "Sharon".

One point to be made here is that the **tmpfile** file name should have the terminal id as part of the file name. If we did not name our temporary files this way, what would happen if two people ran this shell script at the same time? They would both be using the same file name which would cause some problems. We should always use *unique* file names when creating temporary files. The user name would be sufficient as long as we could guarantee that the same user name is only logged in one time which may usually not the case. The terminal id is normally the only unique name to use since we know only one user can be logged in on a terminal at one time. Example 8 shows a helpful way of getting this piece of information (there are more sophisticated ways using the **grep** and **awk** commands but these are not available on all systems):

This may look complicated, but when we break it down, the concept is quite simple. Of course, we have also introduced a new concept here, which we will explain. The program **main.sh** first sets the value of **name** to "David". Then, we set a variable called **dir** which is a combination of the literal string "/tmp" plus "the result of" the **tty** command. Supposing in this example, we are logged into the **console**. The result of the **tty** command would be **/dev/console** so our **dir** variable is now set to **/tmp/dev/console**. This is also a unique file/pathname since only one person can be logged in on the **console**. If we had been logged into **tty02**, the **dir** string would be equal to **/tmp/dev/tty02** which is also a unique name. We then export the **name** and **dir** variables.

The next line makes a directory **/tmp/dev**. If you take a close look at what we are doing here, you will see that we are actually using the output of the **tty** command as the filename/pathname in which we are going to store the value given to the **name** variable in **export.sh**. For example, if the user is at the **console**, the value of the **name** variable will be stored in the file **/tmp/dev/console**. If the user is at **tty02**, the value will be stored in **/tmp/dev/tty02**, etc. If we do not create the **/tmp/dev** directory first, however, we will get a syntax error because we cannot redirect output to a file in a directory that does not exist. We have also introduced a new concept here - the ***UNIX garbage can.***

Anytime we do not want the user to see the output of a command, we can divert the output to the file **/dev/null**, which is the **UNIX** garbage can. This is normally used in shell programming to purposely keep the user from seeing the output of a command. The **rm -r "$dir > /dev/null** at the end of **main.sh** will recursively remove the temporary directory we have created. What would the user see if the directory did not exist and we did not redirect the error message out **/dev/null**? He would see the message **rm: /tmp/dev/console nonexistent**. This message has nothing to do with anything the user did and it would certainly confuse him. By redirecting it to **/dev/null**, the user would not see the message.

We also can divert diagnostic messages (called **standard error**) to **/dev/null**. By putting **2>&1**

continued on next page

Example 7:

Program 1: main.sh

```
name="David"; export name
export.sh
name=`cat /tmp/tmpfile`
```

Program 2: export.sh

```
echo "$name"
name="Sharon"
echo "$name" > /tmp/tmpfile
```

Example 8:

Program 1: main.sh

```
name="David"
dir="/tmp"`tty`
export name dir
mkdir /tmp/dev > /dev/null 2>&1
export.sh
name=`cat "$dir"`
rm -r "$dir" > /dev/null
```

Program 2: export.sh

```
echo "$name"
name="Sharon"
echo "$name" > "$dir"
```

Example 9:

Program: shift.sh

```
# This shell script gives an
# example of the "shift"
for name in $1 $2
do echo $1
   shift
done
```

How it would work:

```
$ shift.sh apples oranges
apples
oranges
```

after the redirect to **/dev/null**, we are telling the shell to divert the standard output and any diagnostic messages out to **/dev/null** which guarantees that the user will not see any output from the command. In the example above, if we do not divert the diagnostic messages when we do the **mkdir** command, the user will see a message if the directory already exists. (This situation could occur, for example, if we had a power failure while the user was executing this shell script–the **rm -r** line would not have had a chance to be executed). You will find the need to redirect standard output and diagnostic messages when you are writing shell scripts.

To continue with our example, after we create the **/tmp/dev** directory, we execute the **export.sh** script which echoes the value of **name** (David), changes the value to "Sharon" and stores this in the **/tmp/dev/console** file. Control returns to the **main.sh** script and we make **name** equal to the result of the "cat /tmp/dev/console" command. We then remove our temporary file. Temporary files should always be removed when they are no longer needed.

We need to discuss two other commands before we leave you for this issue.

THE *exec* COMMAND

The **exec** command is used whenever we want to execute another shell script (or process) but responds differently the response that occurs when we than simply enter the script name by itself. When we use this command, the newly executed shell script replaces the current shell script process. When this is done, control will not pass back to the current shell script. This is normally done when we know that we are finished with the current shell script and want to execute another shell script from which the user will exit. This can be demonstrated in the above shell script. Change the line that reads **export.sh** in the **main.sh** script to read **exec export.sh**. If you recall, before we made this change, control passed back to **main.sh** after the **export.sh** script was finished. When we use the **exec** command, however, the **main.sh** process is *killed* and the **export.sh** process replaces it. If you make this change, you will see that the last two lines of **main.sh** will not be executed because after the **export.sh** script is finished, the user will exit back to whatever process was running before **main.sh** was executed.

THE *shift* COMMAND

Remember last time we said that positional arguments **$1** to **$9** were available for passing variables from one shell script to another. At first glance it appears, with this method, that you can only pass 9 arguments. Actually, you may pass more than 9 arguments. When you do this, you can use the **shift** command to shift each of the line arguments to the left. After the **shift**, **$2** becomes **$1**, **$3** becomes **$2**, etc.

SHELL COMMENT LINES

We also neglected to mention that any line that starts with a **#** is a comment line in shell scripts. The shell will not try to execute these lines as commands. See example 9.

When this shell script is run, the first two lines will be ignored by the shell. The first time through the loop "apples" would be displayed because it is the value of **$1**. We then **shift** the value of **$2** (oranges) into **$1** so the second time through the loop **$1** would now be equal to "oranges" and that is displayed. This would continue for as many positional parameters as were passed.

As you can see from the last two articles, we are now fine tuning our shell script writing abilities and covering a number of concepts and individual commands. The next issue will continue on the same course. After we have covered all of the relevant commands and concepts of shell programming, we will present a complete menu type shell script that will serve as a summary of everything we have discussed in this series of articles. See you next time!

UNIX BOOK STORES

In the past we have mentioned **UNIX** and **Multiplan** books that may be of interest to our readers. Most of these books are available at large computer or book stores. However, many of our readers have had difficulty obtaining them. One excellent source of **UNIX** books is the **Cucumber Bookshop, Inc.** at 5611 Kraft Drive in Rockville, MD 20851. They can be called at **(301) 881-2722**. If you contact them, they will be glad to send you a list of the books they have available. They call themselves the leading **UNIX & C** Language bookstore in the nation, and are specifically set up for mail order trade.

Unfortunately, they do not carry the **Ruth K. Witkin's** book called **Managing your Business with Multiplan** that has been the basis of our recent series on Multiplan. That book was published by **Microsoft Press** in 1984. If you need to contact Microsoft to obtain a copy, they are located in **Bellevue, Washington**. You might also try finding this book in a local computer bookstore. Although this book is not written specifically for the Fortune, it provides a very useful introduction to Multiplan. □

ATTENTION LAW FIRMS

You made a smart decision when you bought your Fortune computer for word processing and office automation.

Now take another giant step forward and install Gill & Piette's

TIME & FINANCIAL MANAGEMENT SYSTEM FOR LAWYERS (TFM)

on your existing Fortune system.

The product of over 25,000 hours of research, development and testing, TFM is a total law office accounting and management information system featuring:

- *autoflex billing in multiple, user-definable formats*
- *integrated client accounting and general ledger*
- *flexible management reporting for work in progress, write offs, write ups, time history, billing realization, etc.*
- *integrated modules for Accounts Payable, Docketing, Conflict checking, Trust accounting, Finance Charges*
- *multiple realization measure, including attorney billing realization and fee receipts by working attorney*

Support? We have a national software support and maintenance program that covers everywhere from Bermuda to Washington State. Nobody can support your installations as well as we will, and we've got the track record to prove it.

HOW MANY PROFESSIONAL HOURS CAN YOU AFFORD TO SPEND DESIGNING YOUR OWN SYSTEM OR TINKERING WITH SOMEBODY ELSE'S UNSUPPORTED SOFTWARE?

MAKE THE RIGHT BUSINESS DECISION WITHOUT FURTHER DELAY AND INSTALL TFM, THE FINEST LAW OFFICE MANAGEMENT SYSTEM AVAILABLE TODAY!

FOR MORE INFORMATION CALL

GILL & PIETTE, INC

(202) 546-6170

Glossary

continued from page 1

Note that "*syntax error*" is the most common, and this is usually (but not always) amplified. In fact, all errors detected during verification are syntax errors of one sort or another, including mis-spelled function or keywords, incorrect construction of a statement, incorrectly identified variables, or the wrong number of expressions in a function argument. Appendix D of the **Fortune:Word Advanced Glossary** manual list all the glossary error messages.

In looking through the glossary for errors, it's important to remember that the line referenced by the error may not actually be where your problem is. An error may not be detected by the system until several lines later. Our first error above is noted as line 6, but in fact the problem is a missing } at the end of the entry. That missing character is also responsible for the second error message. You can't start a new entry when the old one hasn't been closed.

The "*unexpected variable*" in line 15 is related to the "*unmatch character detected*" message. By failing to close the quotes around the string variable in line 14, the system picks up the next pair of quotes to close it, which makes what should have been a string variable appear to be a keyword, and the system doesn't recognize it.

Listing of Glossary

```
line 1      entry a
line 2      {
line 3      search "second" execute
line 4      cancel insert "first and " execuet
line 5
line 6      entry b
line 7      {
line 8      merge "name" MERGE
line 9      call keysin return
line 10     }
line 11
line 12     entry a
line 13     {
line 14     goto "e
line 15     insert "this is the end" execute
line 16     }
```

Output on Work Page

```
**********************************
Tue Jan 20, 1987 at 16:55:40
**********************************

page 1 , line 6 : syntax error :  Cannot start another
entry here
page 1 , line 12 : syntax error :  Cannot start another
entry here
page 1 , line 15 : syntax error :  Unexpected variable
page 1 , line 15 : unmatch character detected( " )
```

Second Set of Verification Errors

```
**********************************
Tue Jan 20, 1987 at 17:10:29
**********************************

page 1 , line 5 : syntax error :  '}'
page 1 , line 12 : Multiply defined Entry name
```

So we fix those problems and verify the glossary. We should be all set -- after all, we corrected all the listed errors, didn't we? Wrong. Once again there's a green box, telling us we have 2 errors detected. Back again to the work page (which is shown in the third part of the inset box).

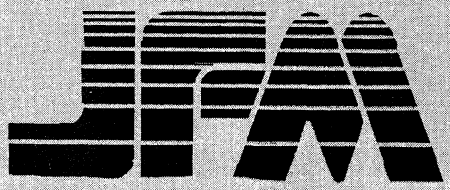
What in the world is "syntax error:'}'" on line 5? We know that's supposed to go there. The real culprit is in the *previous* line where we've mis-spelled a keyword ("execuet"). Just why this particular error message is produced is beyond me. Let's just fix it, and, of course, use a new name for the third entry. Verify it again and ... sigh of relief! We're back at the menu.

This has been an example of dealing with verification errors. You should also know that you can experience other "bugs" with your glossary as well: execution and logic errors. They are related to the way you actually write your glossary and don't show up until you try to run it.

Here are some suggestions. Deal with the verification errors by looking for simple mistakes first. A whole series of errors, as we have seen, can result from forgetting a single character. Copy the workpage to another page and print it for reference if the list of errors is long. If you frequently have trouble with mis-spelled keywords, make up a glossary of them! You can use a glossary in preparing a glossary, and you could also let it help you remember to put in the opening and closing curly brackets.

When using sequences of keywords, you can go through the operation you want the glossary to perform manually at first, writing down each step in sequence. Creating a "Glossary by Example" may be useful. If your glossary uses loops, make sure you get the results you expect after the first pass before letting the loop continue.

Above all, relax. Debugging glossaries is an unavoidable part of programming. Take time to read the **Fortune:Word Advanced Glossary manual.** It has some great tips for successful glossary writing and error correcting and is one of the better instruction manuals around. □



Dear Fortune Owner:

As you have probably read in */u/fortune news,* Fortune Systems SX models are significantly more powerful and faster than the PS and XP models. If you have a PS or XP model system, you should take advantage of our special promotion on SX upgrades. For a limited time, and while our supplies last, we will upgrade your system, including the new SX Cabinet at no additional charge for the cabinet. With this special offer, you may choose any combination of the following upgrades and/or expansions:

- *SX Processor with Current Release of Multi-User For:Pro*
- *60 MB Internal Streaming Tape*
- *30 MB, 45 MB, 70MB High Speed Hard Disk*
- *6-Port Intelligent Asynchronous Controller*
- *Upgrade to Current Release of all Fortune Software*
- *Fortune:Windows (Included at NO Charge)*

Additionally, we can perform these upgrades within 24 hours of receiving your system. In fact, if you ship your system overnight on a Friday for Saturday delivery, we will have it back to you on Monday morning, all ready to plug in and run; not losing the use of your system for a single work day. And, we will issue you a full 90-day warranty on the entire system.

JFM provides services to Fortune Systems owners either directly or through our authorized Affiliated Dealers. If you haven't used our services, I invite you to discover JFM; you will be glad you did.

Sincerely,

Morris Frydman
President

Talk to us, then talk to our clients

JFM
BUSINESS
SYSTEMS
The Merchandise Mart
Lobby Suite 144
Chicago, Illinois 60654
Call (312) 644-0870

Authorized Fortune Systems Master Dealer

BACKUP AND RESTORE UTILITY

PROTECTION MADE EASY!

- o Back up all or selected user directories
- o Restore all or selected user directories or files
- o Format diskettes
- o Review backup volumes
- o Menu-driven, easily installed
- o Complete documentation

Also copies the files necessary to preserve your system configuration--a tremendous help if your system ever needs to be cold booted!

$150.00 Backup/Restore Utility
$200.00 Utility + 2 boxes 5 1/4" 96 TPI high-quality diskettes

For more information, call CTD at (313) 540-1282.

To order, make check payable to *Computer Trade Development Corporation*. Send to:

BACKUP/RESTORE UTILITY
Computer Trade Development Corporation
30600 Telegraph Rd, Suite 1386
Birmingham, MI 48010

CTD

Company:_____
Contact:_____
Address:_____
City:_____
State:_____ **Zip:**_____

Looking for Business Accounting Software that will conform to your Business?

Look to DSM

Applications
- Can be tailored to your organization's specific needs
- Economy of Standardization
- Follows Generally Accepted Accounting Principles (GAAP)
- Luxury of Adaptibility
- Data Base Incorporation

Software for:
- Not-for-Profits
- Construction
- General Business
- Property Management
- Steel Service Center
- Law Office Management

Hardware
Fortune Systems, NCR, AT & T, IBM

Call or Write
Data Systems Management, Inc.
277 Coon Rapids Blvd.
Minneapolis, MN 55433
(612) 780-5200

DSM

All software was developed using PROGRESS, a powerfull relational data base which allows you to tailor the software to your specific needs.

PROGRESS

Dealer Inquiries Invited
PROGRESS is a trade mark of Data Language Corporation.

NOTICE: If you have recently received a UNIX TOOLS diskette or the UNIX|STAT diskette, you may have experienced problems when trying to install it. If you receive the message "Improper mount or illegal flexible disk," then you have a non-installable diskette. Please contact us and we will send you a replacement copy immediately. Our apologies for this inconvenience.

/u/fortune news
73 Jason Street
Arlington, MA 02174

/u/fortune* news

The Newsletter for Users of the Fortune 32:16 Computer — February 1987/Volume 4 Number 2

Are You A Fortune:Word Master?

Ten Fortune:Word Hints

You've been using **Fortune:Word** for a while and you think you know it inside out, or maybe you've just gotten started and you're still nervous about trying anything too complicated. Well, if you fit in either of those categories, or even somewhere in the middle, this quiz may be useful to you -- either to pat yourself on the back or to teach you some new tricks. Good Luck.

1. *Assuming you are at the beginning of a sentence, what three keys would you hit to delete the entire sentence.*

Answer: **DELETE PERIOD (".") EXECUTE**. When you are copying, moving, or deleting, the block of text that will be selected can be determined by pressing a letter, number, or punctuation mark. When any of these keys are pressed, the cursor will search forward to find the first occurrence of that character. In this case, we entered the Delete mode by pressing the gray **DELETE** key. Next we highlighted the text all the way up to the period simply by pressing the **PERIOD**. Finally we completed the deletion by pressing the **EXECUTE** key.

Note that this will leave you with two spaces after the period. These can be avoided by starting the deletion immediately after the period in the sentence before. Then your complete deletion can be defined by the period.

This same technique can be used to delete *words* by pressing the **SPACE BAR** (assuming there is a space after each word), or *paragraphs* by hitting the **RETURN** key, or any small section by using a letter or character for which the cursor can search.

2. *If you are writing a glossary entry, what is the difference between typing in the command "up" vs. "UP"?*

Answer: Many "keywords" in **Fortune:Word** have two functions, a SHIFTED and an UNSHIFTED function. For instance, typing in "up" will move the cursor one character up the screen, as if you hit the **UP-ARROW** key. "UP" is the same thing as using the **SHIFT UP-ARROW** combination, which activates the cursor mode movement. Normally this will move the cursor backwards one word, but this can be changed. This is determined by the word "word" on the second line of the display. You can change this cursor mode by hitting **MODE PERIOD** to change it to sentence, **MODE RETURN** to change it to paragraph, or **MODE PAGE (F2)** to page. Whatever mode you are in determines how far the cursor will jump when you press **SHIFT** and **UP** or **DOWN**. For more about moving the cursor quickly through your text, see Volume 3 Number 9 page 16 of **/u/fortune *news***.

For a more complete list of what the **SHIFTED** and **UNSHIFTED** versions of keywords will do, see

See **Quiz,** page 19

News From Fortune

Fortune Releases Financial Results

Fortune Systems Corp. has just released its financial results for the last quarter of 1986 and for the entire year. As you will note in the accompanying table that appeared in the New York Times (see page 2), Fortune Systems' financial performance has dramatically *improved* compared with 1985. The revenues decreased for both the quarter and the year, however the bottom line was considerably up.

In the last quarter of 1985, Fortune showed a loss of more than $15.6 million while in the last quarter of 1986 there was a profit of $.03 per share. For the year, Fortune reduced its losses from $23.6

See **Fortune News**, page 2

In This Issue...

Physicians and Fortune–First of a series looks at five packages for doctors...**Page 3.**

The Unix Directory–Part 7 in our series on shell scripts...**Page 9.**

Unix Basics–Unix Filesystems and associated commands...**Page 13.**

Fortune:Word Quiz–Check your Word processing IQ...**Page 1.**

News From Fortune–Fourth Quarter results are up...**Page 1.**

The BASIC Advisor–Ray Wannall talks to Choice Systems...**Page 22.**

*/u/fortune *news* is published by The Cambridge Consortium, Inc. It is not affiliated with Fortune Systems Corporation.

/u/fortune *news*...

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz, Ph. D.

Technical Editor
Scott Bradner

Advertising Manager
Mark Palmerino

/u/fortune *news*... is published 12 times per year by The Cambridge Consortium, Inc. Neither /u/fortune news... nor The Cambridge Consortium are affiliated with Fortune Systems Corporation.

Subscriptions cost $48 for one year within the U.S.A. For foreign subscriptions, add $27 each year for added postage and handling costs.

Back issues are offered at:

Vol 1, Numbers 1 - 3	$1/each
Vol 1, Number 4	$.50/each
Vol 1, Numbers 5 - 8	$3/each
Vol 2, Numbers 1 - 7	$3/each
Vol 3, Numbers 1 - 12	$4/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $10 each. Please add $5 per order for foreign destinations.

Although every attempt is made to insure that our contents are error-free and without risk, The Cambridge Consortium is not responsible for losses that occur due to correct or incorrect interpretation of **/u/fortune** *news*.

For more information, call or write:

/u/fortune *news*
73 Jason Street
Arlington, MA 02174
(617) 663-7700, (617) 646-5603

The text for this newsletter was prepared using a Fortune 32:16 for text input. The information was then transferred to an Apple Macintosh Plus with kermit and formatted for typesetting using the Aldus Pagemaker program.

UNIX™ is a trademark of Bell Laboratories, Inc. Fortune 32:16™, Fortune:Word™ and FOR:PRO™ are trademarks of Fortune Systems Corporation. Multiplan™ and MS-DOS™ are trademarks of Microsoft. IBM-PC™ is a trademark of IBM. Macintosh Plus™ is a trademark of Apple Computer. Pagemaker™ is a trademark of Aldus Corporation.

©1987 by the Cambridge Consortium, Inc. 2/87

Fortune News

continued from page 1

Qtr. to Dec. 31	1986
Revenue $	11,902,000
Net inc	631,000
Share earns	.03
Shares outst	21,603,000
Yr rev	38,340,000
Net loss	b1,704,000
Shares outst	21,068,000
Qtr. to Dec. 31	1985
Revenue $	14,885,000
Net inc	c15,611,000
SWhare earns	
Shares outst	21,188,000
Yr rev	47,510,000
Net loss	23,561,000
Shares outst	21,082,000

b - includes a gain of $471,000 on the sale of a German Subsidiary
c - Net loss.

million to $1.7 million. As the table indicates, this reduction was due in part to the sale ($.47 million) of Fortune's German subsidiary. As an independent company with a contract to be the master distributor for Fortune Systems International in Germany and Austria, the company operates under its original name -- Fortune Systems GmbH.

Last June Fortune Systems created a separate division to market its software for non-Fortune hardware. The wholly owned subsidiary is called **Tigera Corp**. and it will market a line of UNIX office automation software for OEM hardware makers. It has already announced the availability of its first product -- **WORD ERA**, a word processing program for multi-user systems. It is an enhanced version of **Fortune:Word**.

Finally, Fortune Systems has announced that it has received an offer for a buyout of its computer operations. The offer was made by a management group led by **Robert A. Davis**, the recently appointed Executive Vice President for Marketing, and **Brooke P. Taylor**, the long-time President of Fortune Systems International. **James S. Campbell**, President and Chief Executive Officer of Fortune Systems, said that the Board of Directors would give the offer serious consideration while exploring alternative proposals. The terms of the offer were not made public and it is not clear what the implications of this offer are for Fortune end users.

More News...

Things are looking up at Fortune. As has been reported in the past, Fortune Systems is putting great effort into improving their sales and marketing organizations. As part of this drive, their goal was to sign on 50 new master dealers across the country by the end of the year. They already have contracted over 25 dealers. In addition, they have begun an advertising campaign aimed at developing more interest among smaller dealers and VAR's. During the month of February, Fortune has placed 2-page four-color ads in Computer Reseller News and Computer Systems News. These ads highlight Fortune's strengths, including over 100 vertical markets that Fortune supports. Later in the spring, they will begin a campaign aimed at the end user market.

The Fortune **Formula**, which was released late last year has been selling well, and there are many in operation in the field. A **four megabyte** memory board has just been released for the **Formula**, making it possible to have sixteen megabytes of total system memory installed. The Formula is targeted at users who require a greater number of users with faster processing speed than that available with the 32:16 line.

There still is no release date set for the new version of **For:Pro** and **Fortune:Word**, although we've heard that For:Pro will be due out shortly.

Physicians Using The Fortune

by Benzion Chanowitz

What software is available for physicians who use the Fortune computer? In this article, five packages are discussed that perform a wide variety of functions that are necessary for running everything from a small practice to a small hospital.

In our Spotlight column this month, we begin a set of articles on physicians and the software packages that are available for them on the Fortune 32:16. Each of the packages is sketched; and together they present a spectrum of software tailored for different settings -- from the single physician's practice to group practice clinics and to hospital environments; and they vary in the degree to which they support the administrative/financial versus clinical care activities of physicians. (Please note the **TOM Software** has withdrawn its medical product, Medical Group/Clinic Management Information System, along with all its other software products from the Fortune market.) In addition, two end-users are profiled. Each uses a different software package. In practice, they offer contrasting, innovative pictures of how the Fortune machine can be used in the medical setting.

While sorting through the research for this article a number of facts emerged. First, many of the medical software packages that run on the Fortune machines will also run on other computers and under other operating systems including IBM, NCR, and Plexus. However, in more than one case, the software subsequently produced for many machines was *originally* developed on the Fortune 32:16. Several developers and vendors use the Fortune in-house, though they understandably chose to remain anonymous since they offer a variety of hardware products. Often, interviewees volunteered the information that their Fortune users constituted the group that was *most* satisfied and least harassed by hardware problems. The Fortune users' major complaint was that they felt "isolated." One of our profiled end-users specifically asked us to relay a request through **/u/fortune *news*** for contact with any other physicians that use the software that he does, **The Resident**, put out by Wallaby Software Corporation (Dr. Hellerstein's telephone number at Houston Diagnostics Associates is 713-675-3431). Also, as some of you may remember, **The Cambridge Consortium** runs an electronic bulletin board for Fortune users. (Currently, there are no charges but you will have to pay the telephone company for using its facilities.) We have set up a separate notefile on the bulletin board for physicians so that they can share problems and solutions common to their practice. The bulletin board telephone number is 617-648-1263.

We will first briefly review the features of five software packages, highlighting the orientation of these programs. Then we will focus on two pieces of software and the ways in which two end-users employ the packages innovatively in the medical setting. (In a subsequent article, we will profile the practice of physicians that use the five briefly reviewed programs detailing those programs in greater depth.) In some cases, the software is directly available from the software developer. In any case you can request more information from your local dealer.

Sumware, Inc. (Laguna Hills, CA) offers **MEDI:FOR**, a program primarily oriented to support the administrative and financial tasks of single physicians and group practices. It can track the progress of patients but it isn't aimed to support the activities associated with patients' clinical care. Rather, it supports such functions as: Patient register, patient scheduling, electronic claims, and third party billings. It can also generate reports monitoring productivity in the practice. With an appropriate configuration the system can account for an "infinite number of patients." Sumware recommends a minimum capacity of 512K of memory and a 20 Meg hard disk for running the software. It reports that a

continued on next page

Tom Software Not For Fortune

TOM Software (Seattle, WA) has informed us that they are temporarily pulling their software for the Fortune System off the market. Their software is developed using **TOM Basic** and compiled for the Fortune 32:16. A sub-contractor is responsible for this compiler. Several bugs have been discovered in this compiler and until the sub-contractor fixes these bugs, **TOM Software** will not support the Fortune 32:16.

substantial number of physicians are using the program on the Fortune, all satisfied with its performance. **MEDI:FOR** is one of the three software systems that have been endorsed by **American Medical International**, one of the largest hospital chains in the United States. Sumware indicates that the Fortune has run up to 14 users on its system. Although other hardware has run up to 80 users with **MEDI:FOR**, we can anticipate that with the advent of the **Formula** from Fortune Systems the capacity to run many users will be greatly increased. (Note that software crafted for the Fortune 32:16 will not run directly on the Forumula. In principle, there should be no problem transferring this software although a revised version will be necessary. If you anticipate purchasing a **Formula**, make sure that any current software you have or will buy which was tailored for the 32:16 will *in fact* run on the **Formula** before you make the financial commitment). The software license for **MEDI:FOR** is currently available at $6,000. This includes 40 hours of training. Further training and support is available for $50 an hour.

Heizler, Granzon Associates Inc. (Denver, CO) offers **Hospital Incident Reporting System**. It was originally developed for a company that owns a chain of sixteen hospitals, currently its only users. Intended for the hospital environment, it is a tool for risk management and hospital service planning. That is, it tracks hospital incidents in terms of their potential for litigation and incurred insurance costs. It monitors the prevalence of such incidents as wrong dosages of medication, cigarette burns, and falling out of chair. At the same time, the software registers facts about the incident including name, ID, etc. of patients, whether they were in or out patients, area of the hospital in which the incident occurred, description of incident, and type of incident. As the latter item implies, classification schemes can be devised at the initiation of the software user and any number of items can be added to the reporting instrument. The system uses **PREFIX** for the development of screens and runs off the **RUBIX** database system. The developer recommends at least 512K memory and 10 meg hard disk space to operate the software. In questioning the developer, we confirmed that this system could be adapted to epidemiologically oriented data collection and study.

Computer Information Services of New Mexico (Alberquerque, NM) offers three related pieces of software: **CIS-MED**, **CIS-EMC**, and **CIS-SKD**. In order to run any of them on the Fortune machine, you need **Business BASIC** and **IDOL**. All three are oriented to supporting the administrative and/or financial activities of single or group practice physicians. **CIS-MED** can do direct billing and accounts receivable; and it can accumulate a data base from patient chart information in order to facilitate practice management. **CIS-EMC** is an insurance module geared

continued on next page

UNISOFT INC.

Experts in Computer Consulting - Sales - Service

Presents

FORTUNE SYSTEMS HELP

We Provide:

- Expert Trouble-shooting
- Years of programming and accounting experience
- *Free Initial Consultation*
- Training Available for:
  - * BASIC/IDOL/BAS and Multiplan
  - * UNIX and Ext. Fortune:Word
- Hardware Repairs & Upgrades
- FORTUNE trained experts at a cost that will surprise you!
- *Trade-Ins Welcome*
- Specializing in:
  - * Hard Disk Trade-In
  - * Hard Disk Upgrades

Authorized Fortune Master Dealer

Authorized Fortune Systems Repair Depot

CALL: (303) 671-8600 EMERGENCY HELP: (303) 691-HELP

UNISOFT INC. 14100 East Jewell Avenue Suite 27 Aurora, Colorado 80012

to bill and handle third-party reimbursement procedures. **CIS-SKD** is a patient scheduling module. The minimum configuration includes 512K memory and a 20 meg hard disk. CIS also recommends two terminals and two printers -- one for patient data entry and one for patient accounting. The cost for **CIS-MED** is $2498 and include 24 hours of training. Each of the other two programs are available for $998 and includes four hours of training. This price also includes a six-month warranty with maintenance and updates. For $100 a month, subsequent maintenance, support, and updates are available. This system, too, is currently being used by a substantial number of single and group practice physicians.

Spectra Data Systems (Van Nuys, CA) offers **Medical Billing**, a straightforward system designed for direct billing, third party reimbursement, and accounts receivable. Several Fortunes currently use the system including physicians in group practice and independent organizations that provide billing services for physicians. The minimum recommended configuration is 512K memory and 20 meg hard disk. The system is available through dealers for approximately $3,995 and this includes eight hours of installation and training. Hourly support/maintenance is available from Spectra at the rate of $75; and monthly contracts are available at the rate of $150 for phone calls up to 15 minutes long.

ATEK Computer Distributors (Sacramento, CA) offers **Sierra Medical Systems** through its so-named division. (Similarly, ATEK also offers Sierra Dental Systems). the name ATEK may ring a bell. They are the new Fortune Master Dealer for Northern California. In addition, they develop software and distribute it through local dealers. As a new person on the Fortune block, they currently have no Fortune users employing **Sierra Medical Systems**. However, they are actively marketing the system for use on the Formula; and there are several user-groups of physicians in the country organized around the use of this package. Members run the system on a variety of hardware. (For the next article, we hope to interview the head of one user-group). The software is designed for single physicians and group practices. In some cases, it is used within the hospital (e.g., the emergency room or surgery unit); but under these circumstances it operates as a stand-alone system independent of the hospital's larger activity. It is not designed to record, for example, the volume of data required of patients at hospital admission or nurses notes. Rather it is designed to support administrative/financial activities such as claim systems, invoice entries, purchase orders, and patient file entries. However, in addition, the software is designed to create clinical data and case histories for patients even though the data entry clerks may not be aware that they are operating under anything other than a financial/administrative mode.

A common configuration cited by ATEK would be a physician operating a private practice with 2 megabytes of memory, a 70 meg hard disk, two printerrs and four terminals.

A common configuration cited by ATEK would be a single physician operating a private practice with 2 meg memory, 70 meg hard disk, two printers, and four terminals. Prices for the system are graduated on the basis of how many users will be on the system. The price ranges from $7,695 to $15,000 depending on the number of users. These prices also include 40 hours of training provided by dealers (who are in turn trained by ATEK) and a 90-day warranty. Further monthly maintenance is available (from ATEK) but rates depend upon the size of the system.

Logic One (Wheaton, IL) offers the **Clinic Management System**, an integrated billing and record keeping package that is appropriate for single practitioners, group practice clinics, and small hospitals. It was developed in conjunction with a medium-sized clinic, Wheaton Orthopedics, the end-user that we profile below. The software is oriented to support financial and administrative activities and in addition it can track patient progress.

The system provides management and control procedures including: multiple clinic identification, patient identification (automatic assignment of patient ID), system security, system customization, end of day procedures, end of month procedures, telecommunications capability, on-line help screens, and menu driven backup procedures. Patient record-keeping capabilities also include patient look-up by phonetic name search (in case of misspelling), patient transactions (e.g. appointment dates, procedure and diagnostic codes, place and type of service, physician, and financial class), and on-line inquiries by coded options. Various kinds of reports and forms can be generated. These include patient information sheet, daily appointment roster, and mailing labels. CMS can also produce productivity reports on a number of dimensions including procedure, diagnosis, revenue source, and insurance company. The software also supports accounts receivable and billing (direct and third party reimbursement) activities. For example, it produces daily and monthly accounts receivable information as part of the end of time period procedures; and it tracks aged receivables producing a different message for accounts by aged period (e.g., 30, 60, 90, 120 days old).

Logic One reports that their system requires at least 1 meg memory and 20 meg hard disk. The price of the software alone, available through Logic One, has just been reduced from $4495 to $3995; and they provide on-site training, telephone support and a hardware maintenance

Continued on next page

service. Contact **Mary Courtney** at 312-690-8208.

Wheaton Orthopedics Ltd., a clinic that specializes in sports medicine, has used **Clinic Management System** for some time. As we noted above, they were involved in the development of the system. The practice involves three orthopedic surgeons, three physical therapists, and three aides. In addition, the staff includes four data entry personnel and three administrators. Most of their medical activity occurs in the clinic. However, they also follow and bill their patients for emergency room visits and hospital surgery. Their hardware includes a total of 100 megabytes of hard disk space, 1 meg of memory, seven terminals, and four printers (three dot matrix and one letter quality). They also have a modem but it isn't used often. The physicians don't directly use the system much, except for some word-processing. Rather, the system is used to manage the patient and paper flow efficiently. That is, the system is not used to support clinical care activities by reporting such items as prescriptions and allergies. They currently have 15,000 id numbers in use.

The system is used to structure the clinic's activity. At the end of each day, bills generated that day at outside sites (i.e., emergency room and hospital surgery) are printed and the next day's appointment roster is produced. The next day when patients come to the clinic for their appointment, they are greeted by a data-entry person who inputs information elicited by questions. These questions are directed by menu driven screens. Four copies of these forms are then printed out and placed in the chart which patients take with them when they see the physician.

The physician notes the procedures and fees on the form when treating the patient; and, upon the conclusion of treatment, the patient hands both the chart and form to a cashier who calculates the bill. Two copies of the bill are handed to the patient, one remains in the chart, and one is used in case of third party reimbursement. At that point, the cashier purges the patient from the appointment roster. For patients treated off-site, CMS merges with **Fortune:Word** and generates a bill for mailing the next day. End of day and month procedures generate accounts receivables and all aged receivables are forwarded to the collections department. Month-to-date and year-to-date productivity reports are also generated that keeps track of the clinic by activity, income, physician, etc. **Janet Luckey**, system manager at Wheaton Orthopedics, reports that they are extremely satisfied with the performance of **Clinic Management System**.

Wallaby Software Corporation (Ramsey, NJ) offers **The Resident** through local dealers. It is aimed for single physicians and group practices rather than hospital settings; and it is oriented to support the range of financial and

continued on next page

PSG

We are the *largest* Authorized Fortune Dealer in the Pennsylvania, New Jersey and Delaware area. As part of the **P**rofessional **S**olutions **G**roup of companies with over 35 years in business automation, we have the *stability* that our customers demand

Our services include a factory trained field engineering team, a complete parts inventory including system upgrade capability and a software support department.

For all of your Fortune needs contact:

Mark Shields of PSG at
2550 Eisenhower Avenue
PO Box 762
Valley Forge, PA 19482

or you can call Mark at (215) 666-1400

administrative activities. These include billing, account receivable, patient scheduling, charting records, practice analysis, and productivity reports. It, too, can be merged with **Fortune:Word** to produce correspondence. Wallaby recommends 1 to 2 meg memory and a 40 meg hard disk. The software received a highly favorable review in a recent issue of *Physicians and Computer Review*. Wallaby reports that a substantial number of physicians use **The Resident**.

Houston Diagnostics Associates is our profiled end user of this system. **Patsy Fineman**, system manager, reports that the clinic is quite satisfied with the software's performance. But she wonders why the latest updates for **The Resident** have not been made available for the Fortune 32:16. Wallaby reports that there has been no great demand for the Version 2.4 update on the Fortune and, in any case, it does not differ greatly from Version 2.0 that is currently available. Rather than review the ways in which this software package supports the financial and administrative activities of the clinic, we will examine the innovative ways in which the Fortune 32:16 is used at **Houston Diagnostics Associates** to support patient management and clinical care.

Dr. Hellerstein is currently the only physician practicing in the clinic. He specializes in internal medicine, oncology, and hematology. He also practices at a local hospital but the clinic's Fortune system is currently not tied into the hospital computer. The system configuration includes a 70 meg hard disk with floppy back-up, four terminals, and runs such software as **Fortune:Word** and **Multiplan** in addition to **The Resident**. Dr. Hellerstein has been using the Fortune for several years now and employs it in several intriguing ways that are suggestive for other physicians' practice. First, while taking histories from patients, he enters the information in his laptop Tandy 2000 and then (using a communication program and an RS-232 cable) he loads it onto the Fortune. A copy of the report is sent routinely to the referring physician by merging through **Fortune:Word** from a list of physicians. Dr. Hellerstein also uses his modem for accessing data bases made available by the **National Library of Medicine**. (See the associated inset box in this issue on data bases for physicians). We were quite intrigued by his use of **Multiplan** to generate a schedule of chemotherapy appointments for cancer patients. Weight and height of the patient are entered into the spreadsheet program in order to calculate surface area which is a determining variable in fixing chemotherapy dosage. The program then creates a calendar for the patient fixing the dates for chemotherapy treatments. Last but not least, Dr. Hellerstein and his software support person created a statistical program for analyzing the results of blood tests administered in his laboratory, as a form of laboratory quality control. In these analyses, he compares the distribution of commercially available control values with the distributions of those obtained from his own patients on such variables as sugar level, white count and red count.

As we noted at the beginning of this article where we listed his phone number, Dr. Hellerstein is eager to be in touch with any other physicians who use **The Resident**. In addition, we have established a notefile on our bulletin board through which physicians can exchange information. Over the next couple of months we will interview more end-users for a second article on this topic. If you have any feedback to offer or wish to contribute innovative ideas for the use of the Fortune in the medical setting, please be in touch with us. □

Data Bases for Physicians

Physicians! *Break out thy modems.* Do you know that the **National Library of Medicine** maintains more than 20 specialized data bases with useful, current information? There is no cost for setting up an account. Charges vary according to which data base is being used and whether the data base is being accessed during prime time (10 a.m. to 5 p.m. Eastern Standard Time) or non-prime time. There is no flat per hour fee for each data base. Rather, charges are based on a complex algorithm that includes such things as connect time, on-line characters received, cpu time, etc. However, there are distinct differences in rates across data bases depending on the degree of "specialization." For illustration, we will relay average rates for two data bases that were provided by the **National Library of Medicine**. These are just estimates since actual charges depend on the computed algorithm. **Medline** provides a bibliography of worldwide biomedical journal articles from 1966 to the present. Approximately 60% of the citations include abstracts of the research. The average charge for **Medline** is $22 per hour during prime time and $15 per hour during non-prime time. For **Toxline**, a toxicology data base, the average charge is $109 per hour during prime time and $102 per hour during non-prime time. All data bases are accessible 24 hours a day and seven days a week. If you are interested in more information or wish to create an account, call the **National Library of Medicine** at one of the following numbers: 800-638-8480, 301-469-6193.

Introducing:

Simply the most comprehensive, extensive, and reasonably priced service arrangement available for Fortune Systems.

Period.

You've heard the promises, signed on the dotted line and shelled out the cold hard cash. But -- and let's be honest -- has the service arrangement you're currently under really lived up to the promises, to your expectations . . . to your needs?
Well, Micro Resources, Ohio's leading Fortune Systems dealer, backed by our affiliates, offers Ohio Fortune users a service arrangement package that simply cannot be beat. With factory specified hardware upgrades worth up to $2,390 -- At No Cost! -- a free subscription to /u/Fortune News, free monthly software service support, and 20% discounts on all subsequent purchases of Fortune System released software and hardware. And there's more.
If your Micro Resources agreement is negotiated by April 15, 1987, we'll extend our standard one year contract to cover the time remaining on your current contract, for up to one additional year . . . FREE!
Best of all, you get the premier Fortune Systems service experts. No dealer in the nation has sold or serviced more Fortune Systems. That means faster, more complete service at a reasonable cost than any other dealer in the state.

Period.

So call Keith Lamb today at (614) 766-2335, and let's talk about bringing your Fortune System up to speed.

Micro Resources
535 Metro Place South
Dublin, Ohio 43017
(614)766-2335
Authorized Fortune Systems Dealer

The UNIX Directory

Introduction to Shell Scripts - Part 7

by David E. Kloes

David E. Kloes is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the Fortune Computer, and is vice-president of the Houston UNIX User's Group. He contributes independently to ***/u/fortune*** *news.*

IIn Part 6, we completed our discussion of passing variables and their values from one shell script to another. We also introduced and discussed some miscellaneous shell concepts. In **Part 7**, we will discuss **control characters** and some more miscellaneous commands/concepts.

Turning *echo on* and *echo off*

When we write shell scripts, there are times when we will *not* want the user to see what is being entered on the terminal. For example, when our shell script requires the user to enter a password, we would not want the password displayed on the screen. The **stty** command is used to change the characteristics of a terminal, such as the baud rate, parity, etc. We can also use this command to turn "echo on" or to turn "echo off". Let's look at the following shell script:

```
stty -echo
echo -n "Enter the password "
read password
stty echo
echo
```

The **stty -echo** turns "echo off". This means that whatever the user types in will not be displayed on the terminal screen. The **stty echo** turns "echo on" and the terminal is back to its normal state. We include the last echo so that the cursor and prompt will be at the next line after the user has entered the password. This is because with echo off, his **<RETURN>** is not echoed. Without this last echo, the next thing echoed in the shell script would appear after the **password:** portion of this line.

Changing The *interrupt* Key

We can also use the **stty** command to change the "interrupt" key. The normal interrupt key on the Fortune system is the **<CANCEL/DEL>** key. Why would we want to change the interrupt key in our shell scripts?

Sometimes we don't want the user to see what they type. The ***stty -echo*** *command turns off what the user sees and a* ***stty echo*** *command turns it on.*

Remember that we stated previously that one of the reasons we write shell scripts is so that we may accomplish a task using the power of Unix, but without letting the user have direct access to it. Unless we protect against it otherwise (we will talk about some other ways later on), if a user depresses the **<CANCEL/DEL>** key while our shell script is running, the shell script is "interrupted" and *he will be in the shell with the $ prompt.* Obviously we do not want this to happen so some shell programmers change the interrupt key to some odd character combination so that the **<CANCEL/DEL>** does not interrupt the program. Look at the following command:

```
$ stty intr a
```

This changes the interrupt key to the letter **a**. Anytime this key is depressed, it will act just like the **<CANCEL/DEL>** key.

If the **<CANCEL/DEL>** key is depressed after we enter this command, it will only display **^?** on the screen and will no longer be the interrupt key. Obviously, the **a** key is not a good key to use as our interrupt key. To set the interrupt back to the **<CANCEL/DEL>**, enter the same command except instead of entering a, depress the <CANCEL/DEL> key. What we see on our screen is:

```
$ stty intr ^? <RETURN>
```

Now the **a** and **<CANCEL/DEL>** keys will act as they normally do. The trick to reassigning the interrupt key is to pick some character sequence that is not likely to be found by a user. We are also limited by the **stty** command because we may only enter *one* character. Consider the following:

```
$ stty intr abc
```

This command would still set the interrupt to just the letter **a** because **stty** will only use the first character it sees as the interrupt key. Instead of using "normal" characters, we should use a "control key sequence". The control sequence we use should be one that is *not* normally used by a user. The one we use is **CTRL-SHIFT-6**. This would be entered by holding down the **CTRL** key, **SHIFT** key and the **6** key (not keypad) simultaneously. If you remember, we stated earlier that if you use "control keys" in your shell scripts, they should be preceded by the **CTRL V** key.

Continued on next page

(**Editor's Note**: Actually the **CTRL V** is necessary when writing the shell script in either the **ed** or **vi** editor. If you use the **sc** or **screen** editor, then you need to precede the control characters with a **CTRL C**.)

So our entire sequence to enter would be:

```
CTRL V, CTRL SHIFT 6
```

On our screen we would see **^^** after this sequence is entered. Once the interrupt is set to this sequence, the user would have to depress the **CTRL**, **SHIFT**, and **6** key simultaneously in order to interrupt. You can experiment and find your own sequence.

Seems simple. All we have to do at the beginning of our shell script is enter the **stty** command to change the interrupt key and before we exit, use another **stty** command to change it back. Unfortunately, nothing is ever quite so simple. We do have one problem. Changing the keys using the **stty** command at the **UNIX** prompt is not a problem. Entering them into our shell scripts is another matter. The problem is that when we are in the **vi** or **ed** editor, these editors react to certain control sequences. In particular, if we depress the **<CANCEL/DEL>** key in either editor, the editor reacts abnormally. Normally, the sequence that changes the interrupt to some other key is not a problem. It is the **stty** command that changes it back to the **<CANCEL/DEL>** key that gives us headaches because this is where we must depress the **<CANCEL/DEL>** key. How do we get around this problem? The answer is that before we fire up the **vi** or **ed** editor to enter these **stty** commands, we should change the interrupt key to some other key so that we can depress the **<CANCEL/DEL>** key. Since it is no longer the interrupt key, we can now use it in the editor. Let's review the steps:

1. Change the interrupt to some other key. We suggest the ~ key:
stty intr ~

2. Use **vi** or **ed** to enter the **stty** commands into your shell script:
```
stty intr CTRL V, CTRL SHIFT 6

stty intr ^?
```

3. Exit the editor and change your interrupt key back to **<CANCEL/DEL>**):
```
stty intr ^? <RETURN>
```

Using Other "control characters"

While we are on the subject of control characters, this is a good time to talk about how to make our shell scripts more functional and "pretty". Control characters give us a lot of flexibility when writing our shell scripts. For example, we can use them to display menus or information with reverse video, highlighting, blinking, etc. In this portion we will discuss how these control keys can be used to turn on and turn off certain *video attributes*. We will also talk about how we can use the backspace control character and cursor control characters to make our shell scripts more functional.

Here's an example of how we can use the backspace control character (**^H**):

```
echo -n "Enter starting
date: MMDDYY^H^H^H^H^H^H"
read date
```

In this example, we have entered six backspace characters after **MMDDYY** (don't forget your **CTRL V** before each **^H** if you are using either the **ed** or **vi** editor). The echo line will be displayed followed by the six backspaces. The effect of this is that the cursor will be over the first **M** of **MM** after the line is echoed. In this case, we have given the user the format of the data to be entered and put the cursor over the first character of data to be entered. Here's another way it can be used:

```
echo -n "Enter Selection:
__^H^H"
read selection
```

In this case, the cursor would be over the first _ character and implies to the user that the response must be one or two characters.

Sometimes in our shell scripts, we want to re-display the same line in the same place on the screen. We might, for example, ask a user to enter some information and if what they enter is not correct, instead of having to redraw the entire screen, we just want to display the same question at the same place on the screen. Let's see how we can use the other control characters to accomplish this task:

```
up2="^K^K"
ext="n"
while test "$ext" = "n"
do echo -n "Continue
(yes/no):       ^H^H^H^H^H"
   read answer
   case $answer in
     yes) ext="y";;
      no) exit;;
       *) echo "$up2";;
   esac
done
```

In this script, we set the variable **up2** equal to two **CTRL K**'s. The **CTRL K** is the character that moves the cursor up one line. We ask the user to answer "yes" or "no". If they answer "yes", control of the program goes to whatever statements follow the **done** line. If they answer "no", we exit the program. For any other answer, we move up two lines which puts us back on the line the original question was asked. The purpose of the **spaces** and **^H**'s after the "Continue" question is to erase any previous answer (assuming it was 5 characters or less) and re-display the question. To be sure we erased the answer, regardless of the length, we would put as many **spaces** and **^H**'s as necessary to the end of the screen (79 minus the length of the question).

Here are the cursor control characters:

```
^K = up
^J = down
^H = left (backspace)
" " = right (space)
```

Video Attributes

In addition to the cursor motion keys, we also have video attribute characters that can be used to jazz

continued on next page

up our shell scripts. Unlike the cursor motion keys, video attributes require a *sequence* to turn the attribute on and a *sequence* to turn the attribute off. Here's an example:

```
echo "     ^HDMain Menu^ID"
```

This line would be displayed four spaces from the left of the screen. The sequence **^HD** turns reverse video on and the sequence **^ID** turns reverse video off. Remember, the sequences when entered in **vi** would consist of:

```
CTRL V, CTRL , HD
CTRL V, CTRL , ID
```

Whatever characters that are between the control characters (in this case the words **Main Menu**) would be in reverse video. When control characters are entered in **vi**, they are visible; however, when you **cat** the file, they are not. You must use the **-v** option with **cat** in order to see the control characters. Figure 1 shows some of the other video control sequences (refer to the "**Using Fortune Terminals**" manual for a complete list).

These attributes may also be used in almost any combination. For example to set reverse video and blink:

```
echo "^HD^HBMain Menu^ID^IB"
```

A good method to use to keep from have to continually retype these control sequences is to set them up as variables and export them for use in subsequent shell scripts. Here's a sample:

```
# Set up control characters
# for bold on & bold off
bon="^H "
boff="^I "
export bon boff
echo $bon"Main Menu"$boff"
```

One advantage of writing our shell scripts this way is so that if we use them on a terminal other than a Fortune terminal, it is easier to go to one place in our script to change the control sequences rather than to each place that they are used.

Other Predefined Parameters

There are a few more predefined variables (or parameters) that we still need to discuss. Although they are not used frequently, you do need to know that they are available.

The first one is **$*** (or **$@**). This variable can be used when we want to reference all of the positional parameters that are passed to a shell script (**$1** through **$9**). Figure 2 shows how it is used.

Notice that both programs produce the same result. Program 1; however, is the more efficient way of accessing each of the parameters that are passed to a shell script. Basically, **$*** refers to all of the positional parameters that are passed, so that we don't have to refer to each one as we did in Program 2.

A second predefined variable that is available to us is **$#**. This variable returns the number of positional parameters that were passed. For example:

Program: all.sh

```
for pos in $*
do echo -n "$pos "
done
echo
echo $#
```

Use and Output:

```
$ all.sh apples 2 3 grapes
apples 2 3 grapes
4
$
```

continued on next page

Figure 1.

Attribute	On	Off
Reverse Video	^HD	^ID
Underline	^HP	^IP
Highlight	^H (space)	^I (space)
Blink	^HB	^IB
Reset	^_	

Figure 2.

Program 1: all.sh

```
for pos in $*
do echo -n "$pos "
done
echo

$ all.sh 1 2 3 four five
1 2 3 four five
$
```

Program 2: single.sh

```
for pos in $1 $2 $3 $4 $5
do echo -n "$pos "
done
echo

$ single.sh 1 2 3 four five
1 2 3 four five
$
```

Figure 3.

```
ls -R . > /tmp/list&
echo -n "
Do you want all the files in alpha order (y/n)? "
read ans
if test "$ans" = "y"
  then wait $!
       echo "Sorting - please wait ..."
       sort -d list > list1
       rm list
else wait $!
     mv list list1
fi
more list1
```

Tri-Tech Computer Services, Inc.

Tri-Tech Computer Services, Inc., with offices in Newton, Mass. and Kensington CT, provides local on-sight service as well as nationwide depot service by Fortune Trained, Fortune Authorized personnel.

In addition to technical support, **Tri-Tech** offers a variety of customer services.

New Systems	Office Supplies	Word Processing
On-Sight Service	Laser Printers	Spread Sheets
Consulting	Communications	Accounting

Pre-Spring Discounted Specials:

Micropolis 70MB Hard Disk Upgrade (requires WD06 and 1.8 UNIX) *Call for prices.*
CDC 30MB Hard Disk Upgrade
Fortune:Word to Extended Fortune:Word Upgrade
Fortune FIS1000 Workstations (Refurb)
Genicom 3404 Printers (Dot Matrix 180 to 500 cps - Refurb)

(All Equipment sold by Tri-Tech carries a 90 day Warranty)

COUPON #1

10% off the total cost of an Onsight or Depot **Maintenance Agreement.**

(expires 5/31/87)
call for quote.

COUPON #2

10% off office **supplies** on purchase up to $100.00 or 15% off office **supplies** and/or software on purchase over $100.

(expires 5/31/87)

COUPON #3

$30.00 per hour labor rate for On-Sight Service (reg. $80.00 per hour) on labor services for the following: Computer Repair, **Upgrades and add-ons**, Consulting and Cabling

(expires 5/31/87)

* This coupon special is offered by **Tri-Tech Computer Services, Inc**. and does not apply to items on sale or already discounted. Void where prohibited by law.

Call **TRI-TECH COMPUTER SERVICES, INC.** Today!!!
Massachusetts (617) 965-4453
Connecticut (203) 828-9784

This program echoes each of the positional parameters and then the **echo $#** line displays the number of positional parameters that were passed. Frequently, we use this variable to test to insure that the correct number of positional parameters have been passed before a script is allowed to execute.

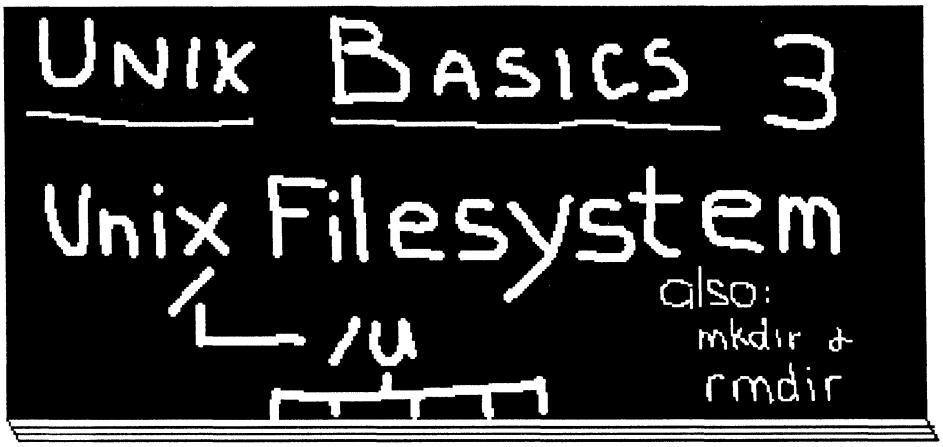
The last predefined variable we will discuss is **$!**. This variable returns the *process id* number of the last background process we ran. Frequently in our shell scripts, we will take a command and run it in the background (**&**) so we can continue on with our shell script. We can find out the process number of the last command we ran in the background by using the **$!** variable. The **wait** command can then be used to stop execution of the shell script until that process is completed. Let's assume we had a menu script and one of the selections on the menu was to list all the files in the current directory. The user has already selected this item and now our shell script has the following logic:

First of all we run the recursive **ls -R .** command; redirect the output to the file **/tmp/list**; and run the command in the background (**&** at the end of the command line). We do this so that the command will be running while we continue to ask the user some more questions (here we are asking only one more question). Whether the user answers "y" or "n", we use the **wait $!** command which says that the shell script is not to continue until the background process (in this case, the **ls** command) is complete.

That brings us to the close of another session. Next time we will discuss the **trap** command, **syntax errors** and one of the most important concepts you will learn about shell programming. See you then! □

UNIX BASICS 3

Unix Filesystem



This month we will explore several more useful UNIX commands. Each of these commands is useful for working with the UNIX *file system*. Now, don't be put off by this thing called a file system because if you understood the UNIX **cd** command (described in Volume 3 Number 12), then you already have a notion of what a file system is. If you were a little shaky about the purpose of the **cd** command, then this article may help you understand UNIX and its file system a little bit more.

So, what is a file system, anyway? I will try to explain what a file system is and I will make liberal use of analogies. First, however, let's start at home - your **home directory**, that is. You have an account on a Fortune computer, otherwise you probably wouldn't be reading this article. Each time you log into the computer you are automatically "put" into your home directory. A directory is a receptacle for containing files. It also refers to the list of contained files. Your home directory can be conceptually understood as an area on the computer's hard disk that is all yours. This is where you keep all your files: Fortune:Word files, Multiplan files, other text files, etc. When you log into the computer and you do an **ls** command, you get a listing of all the files in your home directory.

Now, your home directory is just one directory in a much larger file system. As a matter of fact, each person that has an account on your computer has his or her own home directory. Not only that but all the users' home directories are just one small part, or branch, of the UNIX file system.

OK. Now it's time for an analogy. Consider your desk. It is your work area, much like your home directory. Now imagine that you have several piles of pieces of paper on your desk. These pieces of paper have information on them, e.g., notes from phone calls, letters from colleagues, letters to colleages, etc. Each one of these piles of paper, on your desk, is like a file in your home directory.

If your desk is like mine, it quickly fills with pieces of paper of all sizes and types and pretty soon there is little or no order to the top of my desk; and finding a particular piece of paper is very hard. When that happens, I begin to clean up. Old pieces of paper I throw away - this is analogous to using the **rm** command in your home directory. Another thing I do is gather pieces of paper together that are related and put them in a manila folder. I then put that folder in my desk drawer. This process of specifying a drawer to hold a set of files is analogous to making sub-directories.

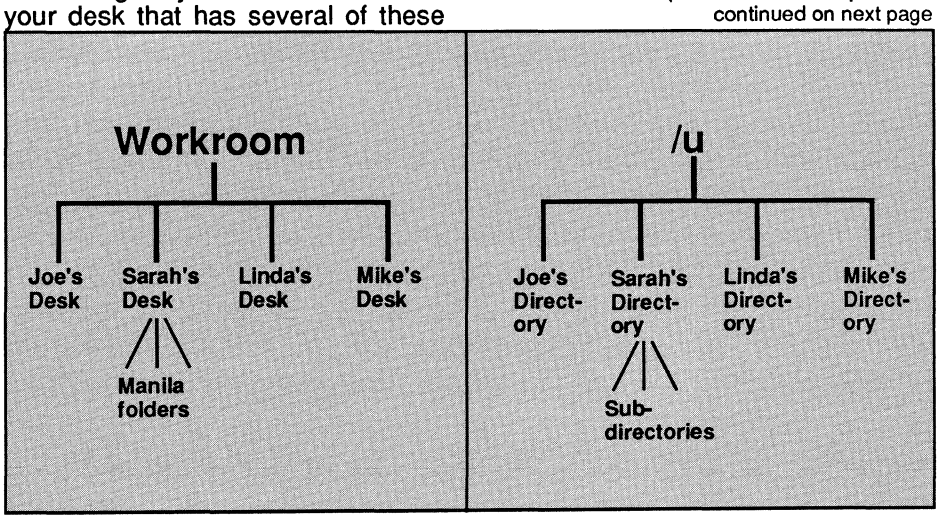
Now imagine you have a drawer in your desk that has several of these manila folders in which there are pieces of paper. The drawer is literally under your desk and it is a part of your desk. In a similar way, sub-directories which you create from your home directory are under your home directory.

So far we have illustrated a small portion of the UNIX file system with this desk analogy. Let's expand the analogy so that you get a grasp of the rest of the file system.

Let's suppose you share your room with several co-workers. For the sake of this article let's say your name is Joe and your co-workers are named Sarah, Linda, and Mike. All of these co-workers, of course, have their own desk with their own drawers and manila folders. The room, which holds these desks, now becomes a part of the structure. Or, in the language of UNIX, we have just moved up one level in the UNIX file structure. This room is like the **/u** directory on your Fortune's hard disk. Figure 1 shows the analogy side-by-side with the UNIX file structure.

I spoke earlier about finding pieces of paper on my desk. When I do this I use my eyes to scan MY desk. I don't look on my colleague's desk. If I do want to look on my colleague's desk (perhaps I left something there) then I must get up from my desk, walk over to his or her desk and then use my eyes to scan that desk. The only desk that I can scan is the desk that I am currently sitting at - that makes sense, doesn't it? Well, this scanning of a desk is similar to using the **ls** command (which was explained

continued on next page



in Volume 3 Number 12) to look at the contents of a directory. Also, when you issue the **ls** command, you will "see" only the files in that directory in the same way that you will "see" only the pieces of paper on the desk at which you are sitting. If you want to "see" the files in another directory, you use the **cd** command to change directories (which is the same as changing desks). Once you **cd** to another directory, then you can look at the files in that directory. Figure 2 shows this analogy.

Thus, if we mix metaphors, we say that **cd** is used for moving from one desk to another and **ls** is used to scan the current desk for pieces of paper.

OK. Suppose I'm back at my desk. I start scanning for a piece of paper on my desk and I don't find it. Then I remember that I took that piece of paper and put it into a manila folder which is in my drawer. If I want to find that piece of paper, I must go to that folder and then scan the pieces of paper in it. Here again, like going from my desk to my colleague's desk, I must go from my desk top to the desk drawer that contains this folder. I am moving from my home directory to one of my sub-directories and to do this I must also use the **cd** command because I am switching from one directory to another. Once I am in my sub-directory, I again use the **ls** command to "look" for a filename just as I use my eyes to scan the folder names in the drawer. Once I get to the folder, I look inside.

Recall that we stopped at your office which contained several desks. This we likened to the /u directory in the UNIX file system. Now suppose that your office is in a building with many other offices. Some offices handle accounting problems, there is a mail room, there are offices for the executives, etc. A picture of this office building is shown in Figure 3. In this picture, you will recognize your workroom with your desk and the desks of your colleagues. This picture is meant to show you that your workroom is just one part of a larger system and this is analogous

continued on next page

Figure 2.

1) Find paper on my desk	**1) Find file in my directory.**
2) Use my eyes to scan my desk.	**2) $ ls**
1) Find paper on Sarah's desk	**1) Find file in Sarah's directory.**
2) Get up and go to Sarah's desk.	**2) $ cd /u/sarah**
3) Scan Sarah's desk.	**3) $ ls**

Our Memory Doesn't Cost You A Fortune

Would you like to:

Speed up your computer?
Add more users?
Use the latest software, like Fortune:Windows™?

Compatibility:	The F1MB is absolutely 100% compatible with Fortune's own 1 Meg memory board.
Affordability:	USA is proud to have equaled and exceeded the quality and performance of our competition, while offering substantial savings for both the dealer and end user.
Dependability:	All boards are burned in for 72 hours and tested for 48 prior to shipment. Our One Year Repari or Replace Warranty underlines our confidency in the product.
Repairability:	The F1MB facilitates repair in the field as it is 100% RAM socketed. Now you can repair a board by simply replacing a defective chip rather than face an expensive board exchange -- reducing maintenance costs and system down time.

If you have a Fortune SX, XP45 or recent model XP30, you can use our board by simply plugging it in. Older systems can be upgraded to use our board with a simple and economical modification.

For Details Call: ***Upgrade Service Associates***

3 Marion Road, Burlington, MA 0180 Phone: (617) 239-8147

Our Guarantee: If you're not completely satisfied within 30 days, USA will gladly refund your money.

to the idea that your home directory on the Fortune is just one directory in a larger UNIX file system.

Pathnames

Perhaps you have heard the word *pathname* before. For some Fortune users the idea of a pathname is complex but it need not be so. Let's illustrate the idea of a pathname with the analogy we have been using. Suppose you want someone to meet you in your office at your desk and suppose this person is coming in from the outside. You would naturally give them directions and these directions might go something like this: "Come to my Office Building, then go to my workroom, then go to my desk and I'll see you there." In essence you have given your friend a *path* to your desk. This may seem pedantic, but lets rewrite the directions to your friend and replace some words with symbols. Let's replace "Come to my" with **cd** and "then go to" with a / (slash). Now we can rewrite the directions like this:

```
cd Office_Building/my_work-
room/my_desk.
```

Well, what do you know? Looks like a UNIX **cd** command, doesn't it? OK, now if you wanted your friend to get a piece of paper in a folder called *letters* that is in your drawer, you would direct this friend to that folder by saying, in our symbolic language: cd Office-Building/my-workroom/my-desk/my-drawer/letter. The sequence:

Office-Building/my-workroom/my-desk/my-drawer/letter

is known as a pathname because it identifies how to get to a particular directory or file.

Let's back out of this analogy and look at pathnames in the UNIX file system. Figure 4 shows an abbreviated map of the UNIX file system on your hard disk. The top of the map is marked simply with a /. This directory, in which every other directory and file resides, is called the **root** directory. It is from this directory that all other directories branch just like all the parts of tree comes from its roots. That is why a file system is often called a tree. You will note that the / in Figure 4 corresponds to the "Office Building" in Figure 3. In some respects this is the top of the world and every part of the system has a pathname that includes the root directory. For example, if you wanted to tell someone to go to Sarah's directory you might say something like "Come to my root directory, then go to the u directory, then go to Sarah's directory." If we make the substitutions as we did in the preceding example we get:

```
cd /u/sarah
```

which is the command that will take you into Sarah's home directory no matter where you are in the UNIX file system.

Creating New Manila Folders

Let's go back to our desk. It's really messy and it's time to clean up. Naturally we pull out of some fresh new manila folders. Then we gather up related pieces of paper from our desk and put them into the folders. We mark the folders and put them into our desk drawer. That's how we create new manila folders. We do an analogous thing when we create sub-directories under our home directory.

Let's say for example that we have been writing some letters to a colleague and we have been using Fortune:Word. We've written this colleague five letters and by now these five letters, composed of three Fortune:Word files per letter, are cluttering up our home directory. So, we get the very good idea of creating a sub-directory where we will put these files - in essence clearing our desk of some related pieces of paper and making a new manila folder. The UNIX command for making a new sub-directory is called **mkdir** and its use is extremely straightforward. If we wanted to name this new directory **col_letters**, we would issue the following command to the UNIX prompt:

```
mkdir col_letters
```

After we issue this command, UNIX will create a new sub-directory called **col_letters** for us. If we do an **ls** of our home directory, we will see that there is file there called **col_letters**. However, this is not

continued on next page

Figure 3.

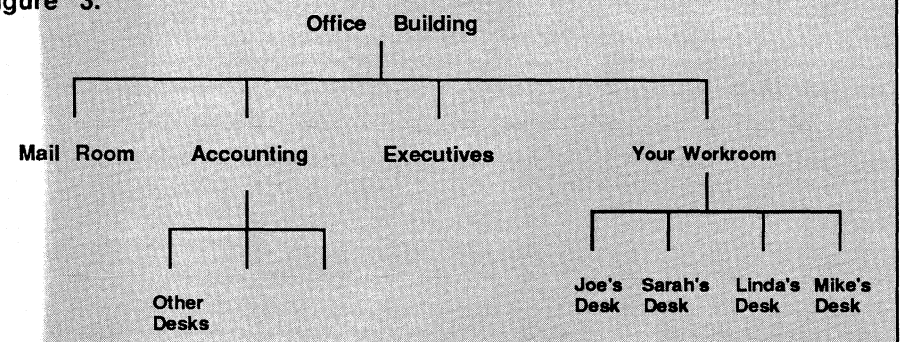
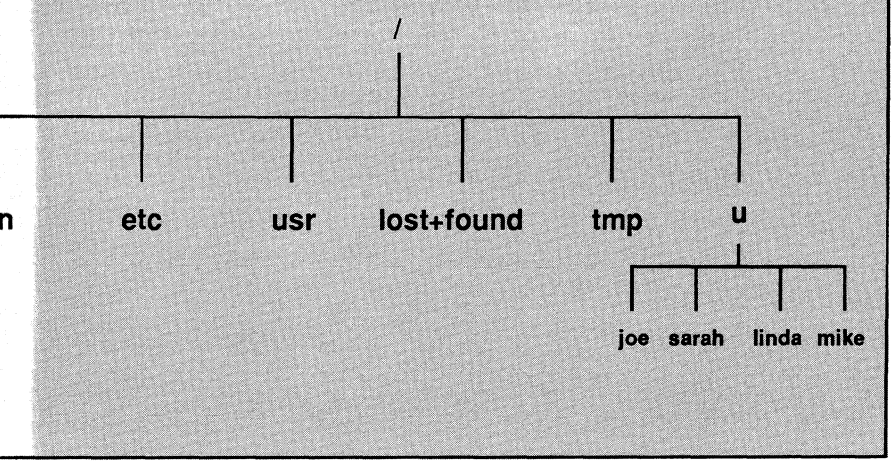


Figure 4.



an ordinary file, rather it is a special file which marks the fact that there is sub-directory under your home directory. The sub-directory is created beneath the directory in which you execute the **mkdir** command. This would be something like having a piece of paper on your desk with the names of all the manila folders you have in your drawer. You could look at that piece of paper to remind you of your manila folders but you could *not* look at that piece of paper to find out what you have in the folders that are in your drawer. In the same way, if you want to look at things in your sub-directory called **col_letters** you will have to go there. To do that, remember, we use the **cd** command as in

```
cd col_letters
```

So, **mkdir** is the UNIX command that creates sub-directories. We will now discuss the flip side of the coin - how to delete sub-directories. The UNIX command that does this is **rmdir** which stands for remove directory. Its use is as straightforward as **mkdir**. If we want to remove the **col_letters** sub-directory, we would simply type:

```
rmdir col_letters
```

However, there is one thing you must know about the use of **rmdir** for it to work properly. It will not remove a sub-directory which has any files in it. If you want to remove a sub-directory you must first **cd** into that sub-directory and then use the UNIX **rm** command to remove, or delete, all the files in that directory. Once the directory is empty, you can use the **rmdir** command without any problems.

The **mkdir** and the **rmdir** are the commands you need to make and remove directories. We have implied that making directories is useful for imposing order on your home directory. If you use your Fortune frequently, then you will no doubt have many files in your home directory. With a little thought you could probably rearrange things so that files of a similar type are brought together in a sub-directory.

For example, we have a directory on our Fortune called **/u/newsletter**. I think you can guess that this is where we keep our files that are associated with the production of **/u/fortune *news***. I also think you can imagine what a mess this directory would become if we kept all the files in it from the newsletter for *every* issue of the newsletter. Besides the fact that there would be hundreds and hundreds of files, it would be almost impossible to find any particular file in a directory listing. What we have done is to create sub-directories for each issue of **/u/fortune *news***. So, for example, while I am writing this article I am actually in the sub-directory called **vol4.2**. The entire pathname for this directory is **/u/newsletter/vol4.2**. In this way we can more easily find things when the need arises.

du and dtree

Finally I want to talk about two other commands that you may find useful in dealing with your home directory and sub-directories. The first is the **du** command which stands for disk usage. This

continued on next page

ATTENTION FORTUNE SYSTEM OWNERS

Paying for a new power supply?
Fixing or upgrading your present one could save you hundreds of $$$$.

Repairing	(Zenith)	$ 75.00
	(Western Electric)	$ 95.00
Upgrading of Zenith to 250 watts		$300.00

Terminal supplies repaired also.

Repaired supplies warranted for 90 days.
Upgraded supplies warranted for 1 year.

Quanitity discounts available. (Call for quote.)

POWER SUPPLIES "R" US
5916 Southwind Dr.
San Jose, CA., 95138
408-365-7678

Looking for Business Accounting Software that will conform to your Business?

Look to DSM

Applications
- Can be tailored to your organization's specific needs
- Economy of Standardization
- Follows Generally Accepted Accounting Principles (GAAP)
- Luxury of Adaptibility
- Data Base Incorporation

Software for:
- Not-for-Profits
- Construction
- General Business
- Property Management
- Steel Service Center
- Law Office Management

Hardware
Fortune Systems, NCR, AT & T, IBM

Call or Write
Data Systems Management, Inc.
277 Coon Rapids Blvd.
Minneapolis, MN 55433
(612) 780-5200

DSM

All software was developed using PROGRESS, a powerfull relational data base which allows you to tailor the software to your specific needs.

PROGRESS

Dealer Inquiries Invited
PROGRESS is a trade mark of Data Language Corporation.

command, if issued without any flags and while you are in your home directory, will tell you how many blocks your home directory is using and how many blocks your sub-directories are using. The following is an example of the **du** command and its output when used in the **/u/newsletter** sub-directory:

```
$ du /u/newsletter

27      /u/newsletter/vol3.1
2       /u/newsletter/vol3.2
1       /u/newsletter/vol3.3
1       /u/newsletter/vol3.4
1       /u/newsletter/vol3.5
1       /u/newsletter/vol3.6
1       /u/newsletter/vol3.7
1       /u/newsletter/vol3.8
1       /u/newsletter/vol3.9
2       /u/newsletter/vol3.10
100     /u/newsletter/vol3.11
95      /u/newsletter/vol3.12
149     /u/newsletter/vol4.1
50      /u/newsletter/vol4.2
...
966     /u/newsletter
```

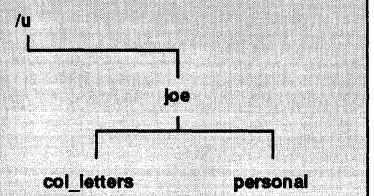
Each sub-directory of **/u/newsletter** is listed preceded by the number of blocks used by that sub-directory. The last line of the output (**966 /u/newsletter**) shows how much space is being used within the **/u/newsletter** sub-directory - this naturally includes all the sub-directories under **/u/newsletter**.

This command is a helpful supplement to the **ls** command. Recall that the **ls** command will list out the files of the directory in which you issue the command. A variant of the **ls** command, **ls -s**, will list the files in your directory along with their size. However, this will not give you the amount of blocks in a sub-directory. This is where **du** comes in because it is often helpful to know how much space is being used by a sub-directory. Also, since **du** lists out the names of the sub-directories, you can get a sense for what the directory "tree" looks like under the directory of interest. So, look at the following output:

```
$ du /u/joe
1       /u/joe/col_letters
1       /u/joe/personal
3       /u/joe
```

We can see that this output

Figure 5.



suggests the directory structure as shown in Figure 5.

But guess what. There is another command, called **dtree**, which actually gives you a pictorial representation of the directory structure under any directory. **Du** is a command that comes with your computer but **dtree** does not. However, we have **dtree** on our software diskette called **Compressor's Delight** which can be purchased from us for $10. An example output from **dtree** is shown below:

```
$ dtree /u/joe
Sun Feb 22 14:06:33 EST 1987

Directory structure of /u/joe
(excluding entries that begin
with '.')

|-joe--------|-col_letters---
             |-personal------
```

This output shows that there are two sub-directories which branch off of the **/u/joe** directory and these sub-directories are called **col_letters** and **personal**. **Dtree**, like many other UNIX commands, has flags which modifies how it works. For example, the above output showed only sub-directories. Supposed we wanted to see the files in these sub-directories. Figure 6 demonstrates how to do this.

This output shows that there are five (count 'em) files in the **/u/joe** directory. Two of those files are the names of sub-directories (**col_letters** and **personal**) which is obvious since there is a dashed line following that name. On the other hand, regular files are just listed without any lines behind them (e.g., **finances.MP**, a Multiplan file). Also we see that there are six files in the **col_letters** sub-directory and two files in the personal sub-directory.

Conclusion

In this article we explained the UNIX file system and several UNIX commands that are useful when dealing with the file system. **Mkdir** and **rmdir** are commands that are used to make and remove directories. **Du** and **dtree** are commands that are used to understand different things about the sub-directory structure under any particular directory. **Dtree** is a useful UNIX command but does not normally come with the Fortune computer. If you would like a copy, you may purchase our software disk called **Compressor's Delight** for $10. Send your request to the address listed on page two of this newsletter. □

Mark Palmerino

Figure 6.

```
$ dtree /u/joe
Sun Feb 22 14:15:29 EST 1987

Directory structure and contents of /u/joe
(excluding entries that begin with '.')

                                |-bill.ltr
                                |-bill.ltr.dc
                |-col_letters---|-bill.ltr.fr
                |               |-mary.ltr
                |               |-mary.ltr.dc
|-joe-----------|               |-mary.ltr.fr
                |-finances.MP
                |-notes
                |
                |-personal------|-shop_list
                |               |-to_do
                |-phonenum
```

** *Print Fortune:Word on Terminal Printer* **

See Terminal/Printer Link

SOFTWARE FOR THE FORTUNE 32:16

MULTIPLAN LINK - can pass data from BAS/IDOL/BASIC files to Multiplan and back. For example, it could be used to send data from any of the BAS files (Chart of Accounts, Customer Master, Inventory Master, etc). Allows selection of fields to be sent and provides logical retrieval and key range selection. **$295.00**

RECORDS PROCESSING LINK - can pass data from BAS/IDOL/BASIC files to Records Processing and back. For example, it could be used to take customer address data from the Customer Master and will automatically make the Records Processing List document for mailouts. Allows selection of fields to be sent and provides logical retrieval and key range selection. **$295.00**

ASCII LINK - can pass data from BAS/IDOL/BASIC files to ASCII and back. For example, it could be used to send data from any of the BAS files (Chart of Accounts, Customer Master, Inventory Master, etc). Allows selection of fields to be sent and provides logical retrieval and key range selection. **$295.00**

TERMINAL/PRINTER LINK - provides capability to print Multiplan, and BAS/IDOL reports on a printer connected to a workstation view any report on the terminal before printing, convert any report to FortuneL:Word format; accumulate multiple BAS/IDOL reports before printing, backup reports onto diskette; copy or rename report files; print multiple copies of the report on either the primary and/or terminal printer. Also has option to print BAS/IDOL reports on HP LASER printer ($50 extra). ** ***Also supports printing off auxiliary workstation with Fortune:Word*** ($50 extra) **. **$195.00**

TELEPHONE LINK - on line telephone message, telephone directory and inter-office mailbox/memo system which can also be used as a simple calendar reminder system. Provides control, printing and viewing of on-line data for each user. **$295.00**

FINANCIAL LINK - stores BAS Income Statement and Balance Sheet data in files. Once stored, it can be sent to Multiplan or instantly printed. **$195.00**

MENU LINK - provides software and documentation for adding items to the Global Menu as well as a sub-menu for your use Includes backup script for user directories and BAS/IDOL/BASIC programs and data. **$95.00**

CALCULATION LINK - provides programs for amortization, depreciation, loan repayment, averaging (with graph), linear correlation, and breakeven analysis. **$95.00**

KOMPACT PERSONNEL ACCOUNTING - provides for data capture of personal, wage/salary, job, education, salary/wage, appraisal training, dental, medical and life insurance data. Includes over 80 pre-defined reports/worksheets. **$495.00**

CRT LINK - provides information for using other terminals on your Fortune (WYSE, TV, IBM 3101). **$95.00**

AUTOMATIC REPORT LINK - provides capability to print numerous IDOL defined reports automatically based on frequency or application without having to select each from a different menu. You identify frequency/application codes. **$95.00**

PRINTER LINK - provides printcaps for Okidata 84, 2410, TI 855, and Anadex 6500. Includes shell scripts for changing pitch and LPI. Allows these printers to be used in word processing. **$95.00**

MODEM LINK - allows Hayes 1200 Smartmodem or compatible to be used on one port for both incoming and outgoing calls. **$95.00**

INSTALL LINK - documentation and software example for installing your product/menu through "S5". **$95.00**

REPORT LINK - modifies IDOL Report Generator Program to allow stacking of up to 8 fields in an IDOL defined report. Also corrects problem with displaying subtotals for all stacked fields. **$95.00**

WESTERN LINK - provides instructions on how to use your Fortune for sending/receiving telex messages using Western Union's Easylink. Can replace your telex! Fortune ITE software is required. **$95.00**

UNI-KOMP 9803 Rocktree Houston, Texas 77040 (713) 937-7782

Quiz

continued from page 1

Volume 2 Number 5 page 8 of **/u/fortune *news***.

3. *In the same vein, suppose you want to underline a single word or sentence. How would you create a glossary entry to do that?*

Answer: Glossary entries execute a series of keystrokes automatically like a player piano. A simple glossary to underline a word might use the "cursor mode" movements included with **Fortune:Word** which were discussed in the last answer. If you hold down the **SHIFT** key and press the **UP** or **DOWN** arrows, the cursor will probably jump one word to the right or left.

This is the glossary entry:

```
entry u
{
        mode "_"
        DOWN
        mode "_"
}
```

If you have switched cursor modes, then this same entry would underline an entire sentence, paragraph or page. Similar functions can be created for bolding, double underlining, etc. To speed things up, you might want to add the line **Call display(false)** before the first mode line, and **Call display(true)** after the second mode line. A more complicated glossary entry which uses other punctuation besides just a space to determine where the end of a word is is included in Volume 2, Number 3 of **/u/fortune *news*** on page 2.

4. *Once you've created some useful glossary entries how would you make them available to all of the users on your system?*

Answer: There are several ways to accomplish this. The simplest way is just to change the permissions on the glossary file in the owners directory. If the permissions are set so that anyone can access the glossary files, anyone will be able to attach it by using the full path name, e.g. **/u/fred/letgloss**. You can change the glossary permissions through the global menu, or by typing **chmod 777 letgloss***. Bear in mind that each glossary has four files, letgloss, letgloss.dc, letgloss.fr, and letgloss.gl. If everyone should be able to alter the glossary, change all four files. If you just want them to be able to access the glossary, only change the permissions on the letgloss.gl file.

Another way to do it is to have root or manager copy the entire glossary into the root account (/), or some other short account, so that the glossary can be accessed by simply typing **/letgloss**. This shortens what users have to type. The permissions for this file will have to be changed the same way you changed them in fred's directory.

Finally, you can *link* a copy of the glossary into each person's individual directory. This way, each user will attach the glossary by simply entering letgloss, but because the glossaries are linked to each other, any change made in one version of the glossary, will be made in everyone's version. Although it takes longer to set this up initially, it is probably the simplest approach to use and maintain.

To link the files, first create the glossary. Then change the permissions to 777 by typing **chmod 777 glossname** (substitute the name of your glossary). Finally, type **cp -l /u/fred/glossname* /u/joe**. /u/fred is the original location of the glossary, and /u/joe is the destination. Repeat this step for each user who needs access to the glossary. For more on this whole process, see Volume 2 Number 1 page 6 of **/u/fortune *news***.

5. *How do you print the copyright or registered symbols?*

Answer: Your keyboard is limited to 47 keys, or 94 characters. This isn't enough room for special characters, foreign language characters, etc. To print them, you have to enter **CONTROL-Y** sequences, and alter the **/usr/lib/wheels** file for your particular printer. Although this is not terribly complicated, a detailed explanation would require too much room here. See Volume 2 Number 7 page 12 of **/u/fortune *news*.**

6. *Is it possible to single space a heading page and double space the text of a document?*

Answer: Yes. Line spacing is controlled by the **COMMAND S** option from within **Fortune:Word**. After typing those two keys you will be prompted to enter the desired line spacing. Entering 1, 2, or 3 produce single, double, and triple spacing respectively. **W** produces 1 1/2 line spacing.

If you want more than one line spacing format in a document, you need to create new format lines. These can either be within the text itself, or in the header or footer pages. Each format line can have its own spacing requirements, which are displayed at the top of the screen.

7. *Other than manually typing in current date into each letter, is there a way that Fortune:Word can automatically insert it?*

Answer: A glossary entry can be created which will produce a date in the format "February 14, 1987", or any other format you desire. A complete listing of the entry is given in Volume 2 Number 4 page 4, and another version is in the following issue.

8. *When you are in the Fortune:Word editor, what would be the fastest way to go into Multiplan?*

Answer: Use the **COMMAND !** sequence to issue a UNIX command. To go directly into the Multiplan spreadsheet you would type **COMMAND ! mp68 EXECUTE**. (The words in all caps are a single key.) To go to the Multiplan Menu, type **COMMAND ! master EXECUTE**. It is important to type **mp68** and **master** in lower case. Any UNIX command you would like to use can be put here. For instance, if you have our Electronic Rolodex, you could quickly type **COMMAND ! rolo EXECUTE** and the rolodex would pop right up.

If you are in even more of a hurry, create a glossary entry with those commands in it. Then by typing **LF/GL m**, you would go directly to Multiplan, or **LF/GL r** would give you the rolodex.

Continued on next page

There are two disadvantages of this method you should be aware of. First, your **Fortune:Word** document will not be saved until you return to it and **CANCEL/DEL** out of it. Therefore if your system had any problems before you came back, you might lose your latest work. Although this is an unlikely problem, you should be aware of it.

Secondly, this method will cause more information to swapped to the hard disk. This also will probably not cause any great problem, but it is something to keep in mind.

As a rule of thumb, using the **COMMAND !** method to go to another program should be used for quick access to the other application where you will be going back to Fortune:Word right away. If you plan to go off into the other application for hours, go back through the Global Menu.

9. *Which of the following are set as default values by the prototype document:*

1. Margin Settings
2. Spelling Exception Dictionary
3. Printer Number
4. Header Pages
5. Glossaries to Attach
6. Headers and Footers
7. Workpage notes
8. Form Letter templates
9. Printer Pitch
10. Tab Settings

Answer: If you answered 1, 3, 4, 6, 7, 9 and 10 then you score 100 on this question. **Prototype** documents set default values for 3 general areas when you create a new document. The first is formatting. A prototype document always contains at least one format line which will set the tabs, line length, line spacing, and page length. It may contain as many format lines as you wish. You will be able to copy these format lines in your new document by using the **COPY FORMAT # EXECUTE** sequence, substituting the number of the format you desire for the **#**.

The second area prototypes affect is the printer values. Any of the values that can be set on the Print Document Menu will be stored in the prototype document. Besides Printer Number and Pitch, you can also define the number of bins in your sheet feeder, the first page for the header, etc.

Finally, any text that is stored in the header, footer, or workpage will be copied to a new document from the prototype document. Any other text in the prototype document will not be copied.

You may create and edit a prototype document in the same way you do any other document. You can even use another document as the prototype when you are prompted for the prototype name. If you print the prototype document, the print values you set will be stored. For more information on using prototypes, see Volume 3 Number 4, page 1 of **/u/fortune *news***.

10. *What is the difference between the Spelling Checker and the Spelling Editor?*

Answer: This question could have been phrased, "What is the difference between Interactive and Background processing with the Spelling Checker?". If you have a short document that you want to check immediately, the Interactive Spell Checker will bring you right into the document and scan each word for spelling errors. When mistakes are found, you will be asked to correct them. The actual scanning of each word and comparison to the dictionary can be time consuming, especially for a long document. In these cases, you can check the document against the dictionary in the *background* while you perform other activities. Once this is done, you go back and use the Spelling Editor to correct the document. The Spelling Editor does not compare your text to the dictionary, it only scans for words that have been marked while the Spelling Checker was doing its work in the background. This method requires a shorter period of operator intervention to correct the document, because it is much faster.

Congratulations. If you made it this far, we hope you learned something. (Who cares what your score was?) If you haven't tried some of the techniques discussed here, we encourage you to do so. As long as you have a computer, you may as well try to make it do more work while you do less. Good luck. □

Josh Lobel

Expand your Fortune® without spending a fortune.

New equipment costs big bucks. The Good Fortune Trader sells fully tested burned-in equipment, refurbished to Fortune® standards at prices that won't make waves with your cash flow. Call or write today for a listing of our specials.

THE GOOD FORTUNE TRADER

60 DAY WARRANTY

a division of Computer Services of Atlanta

(404) 392-9938

1534 Dunwoody Village Parkway/Atlanta, Georgia 30338

ATTENTION LAW FIRMS

You made a smart decision when you bought your Fortune computer for word processing and office automation.

Now take another giant step forward and install Gill & Piette's

TIME & FINANCIAL MANAGEMENT SYSTEM FOR LAWYERS (TFM)

on your existing Fortune system.

The product of over 25,000 hours of research, development and testing, TFM is a total law office accounting and management information system featuring:

- *autoflex billing in multiple, user-definable formats*
- *integrated client accounting and general ledger*
- *flexible management reporting for work in progress, write offs, write ups, time history, billing realization, etc.*
- *integrated modules for Accounts Payable, Docketing, Conflict checking, Trust accounting, Finance Charges*
- *multiple realization measure, including attorney billing realization and fee receipts by working attorney*

Support? We have a national software support and maintenance program that covers everywhere from Bermuda to Washington State. Nobody can support your installations as well as we will, and we've got the track record to prove it.

HOW MANY PROFESSIONAL HOURS CAN YOU AFFORD TO SPEND DESIGNING YOUR OWN SYSTEM OR TINKERING WITH SOMEBODY ELSE'S UNSUPPORTED SOFTWARE?

MAKE THE RIGHT BUSINESS DECISION WITHOUT FURTHER DELAY AND INSTALL TFM, THE FINEST LAW OFFICE MANAGEMENT SYSTEM AVAILABLE TODAY!

FOR MORE INFORMATION CALL

GILL & PIETTE, INC

(202) 546-6170

The BASIC Advisor

*The following is an open letter written by Ray Wannall of The BaSiC Software Corporation located in Baltimore, MD. As most of our readers know, Ray has contributed The Basic Advisor column to **/u/fortune** news independently. This month he breaks from his traditional format to air some grievances against Choice Systems. The views presented here are not necessarily shared by us at **/u/fortune** news. However, we welcome the additional opinions of any of our readers on this issue.*

Dear CS:

I have finally had the opportunity to observe, first hand, parts of your **2.0 BAS Payroll**, and I have a few questions. Normally, when I have questions, I call someone and ask, but, for reasons which will shortly become apparent, this does not seem to be an option. Let me start at the beginning.

It all started just before St.Patrick's Day in 1985, approximately six months before the birth of the Basic Advisor. At the time, I was what Fortune Systems called an ISV, or Independent Software Vendor, attending a Fortune sales conference in San Mateo, California. I was there to learn, observe, and, as I subsequently discovered, spend alot of money. Being an old **BAS** bit-twiddler, I was especially interested in finding out what Fortune was planning to do in the future with the Business Applications which had been such a success on the 32:16's. I was generally happy with the progress made to that point (in the form of "TECHNICAL TIP : COMPANY CONFIDENTIAL" releases); **BAS** on the Fortune Systems was one of the cleanest versions available at any price. Therefore, I sniffed out and gravitated toward anyone who was involved in the **BAS** side of Fortune software.

During the first two days of the conference, I was unable to find much enthusiasm among the **BAS** people at Fortune. I knew something was in the air, and, on the third day, I finally found a gentleman who reluctantly let me in on what was happening. Fortune Systems, I was told, was tired of being in the **BAS** Technical Fix business. They had just written a contract (for two-point-something million dollars as I recall) in which you, CS, agreed to take over support of the Business Software. (I told him I would have done it myself for $1.2 million, but you know how humorless these people can be at times.)

I must admit I had reservations at the time, but you see, CS, I am generally a trusting and positive-thinking type of guy. I assumed that passing such an important project to you would be in the best interest of us all. I decided to let you settle in with **BAS** for a while before I called and introduced myself.

I guess you have forgotten by now, but it was in the early summer of 1985 when I first contacted you. The man I talked to was quite the technical salesman. He bowled me over with talk about the largeness of your organization, your flawless reputation and the exciting new enhancements you had planned for **BAS**. Why, your Payroll itself was already at level 2.0 while **SMC** was still stuck in 1.1.1! I was thinking to myself, what in the world could you have done to improve Payroll? (Later I figured it out, however what I imagined turned out to be nothing at all like what you have done. But I will be happy to discuss my ideas with you. Does $1.2 million sound fair?)

We said our goodbyes and I promptly mailed you information about my doings. About two weeks later I called you back, but no one was available to speak with me. I left a message with the gal. I called again about a week later after there was no response and again left a message. You and I kind of went our separate ways after that.

I guess it was last summer when I first started hearing about your 2.0 packages from other dealers and end users. As you know, someone like me would not hear about any package unless there were problems with it. It was then that I realized that you had done more than just improve the packages. Why you had completely rewritten some of them! Now you and I may not agree on this, CS, but it has long been my philosophy that if something ain't broke, you don't fix it. I had to assume that you were attempting to bring the accounting software up a generation or two, and any new package on the market would need time to get the bugs out. So, again, I sat back and waited.

Well, this month I met your new 2.0 Payroll head-on when I was asked by a dealer to make one of my vertical packages interface with it. I asked the dealer to ask you a couple of simple questions about how the coding had been altered. The dealer reported back to me that nobody in your company seemed to know. Now I am sure this is not the case! Perhaps the person with whom this dealer spoke was having a bad day or something. Nevertheless, we had to take action, so I asked the dealer to print a listing of a few programs so that I could see what you had done to them. Now this is where I'm confused.

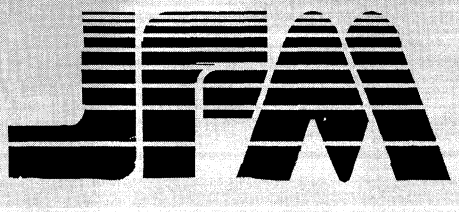
When I received the listings the first thing I noticed was that the majority of the coding was identical to the original package. Upon closer examination, however, I saw that you had changed the structure of the key to the time sheet file called **"CPWKF"**. This constitutes a major change to Payroll! You must have had your reasons, but I cannot, for the life of me, figure out what they were. For the benefit of that person in your company who was having a bad day last week, let me explain my concerns in detail.

You see, CS, in the original **BAS** Payroll time sheets were filed by Company Code (2 characters) plus Employee Number (5 characters) plus a sequence counter (1 character). The sequence counter was included so that operators would be able to enter up to ten time sheets per employee per pay period. This proved to be very helpful for companies who were accumulating time sheets from the field on a daily basis. Operators could enter them into the computer as they arrived, and the computer would total them together at the end of the week and print one payroll check. We software schleppers have been very comfortable with this system for many years. But you changed all this.

For some reason you decided to file time sheets by Company Code (2 characters) plus Employee Number (5 characters) plus Check Number (6 characters) plus Department Code (2 characters) plus Workman's Comp Classification (4 characters). Now maybe there was a good reason to expand Workman's Comp Code from 2 to 4 characters, and there may even be a reason to include Department Code in the time sheet file key. But Check Number is totally beyond my comprehension. I thought check numbers were assigned long after the time sheet entry process was completed. Is it that you had a new payroll report in mind that you did not wish to sort before printing? If so, is it really necessary to eliminate sorting from the payroll reporting process? I mean, unless a company is carrying thousands of employees, sorting for a payroll report is no problem. Come on, CS, we're talking Fortune Systems here!

I guess what I really need is two answers. Why have you changed my Payroll? And what have you done to my other business applications? I would have called you, but frankly I don't know who to ask for. So would you please call me or send me a written response? I would love to publish your reply.

Sincerely,
Ray Wannall



Dear Fortune Owner:

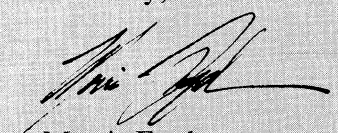
As you have probably read in */u/fortune news,* Fortune Systems SX models are significantly more powerful and faster than the PS and XP models. If you have a PS or XP model system, you should take advantage of our special promotion on SX upgrades. For a limited time, and while our supplies last, we will upgrade your system, including the new SX Cabinet at no additional charge for the cabinet. With this special offer, you may choose any combination of the following upgrades and/or expansions:

- *SX Processor with Current Release of Multi-User For:Pro*
- *60 MB Internal Streaming Tape*
- *30 MB, 45 MB, 70MB High Speed Hard Disk*
- *6-Port Intelligent Asynchronous Controller*
- *Upgrade to Current Release of all Fortune Software*
- *Fortune:Windows (Included at NO Charge)*

Additionally, we can perform these upgrades within 24 hours of receiving your system. In fact, if you ship your system overnight on a Friday for Saturday delivery, we will have it back to you on Monday morning, all ready to plug in and run; not losing the use of your system for a single work day. And, we will issue you a full 90-day warranty on the entire system.

JFM provides services to Fortune Systems owners either directly or through our authorized Affiliated Dealers. If you haven't used our services, I invite you to discover JFM; you will be glad you did.

Sincerely,



Morris Frydman
President

Talk to us, then talk to our clients



JFM BUSINESS SYSTEMS
The Merchandise Mart
Lobby Suite 144
Chicago, Illinois 60654
Call (312) 644-0870

Authorized Fortune Systems Master Dealer

SUPPLEMENT TO ALL VERSIONS OF BAS AT A REASONABLE PRICE, TOO!

JOB COSTING

The Right Tool for All Trades

BaSiC SOFTWARE CORPORATION

For More Information Call or Write...
BaSiC Software Corp.
5201 Powhatan Street
Baltimore, MD 21207
301-448-9460

Available on:
IBM AT & compatibles
UNISYS/SPERRY
Altos, All Levels

DO YOU KNOW...

there is an organization that has a thorough knowledge of

FORTUNE BAS
MULTIPLAN IDOL

We have been providing widespread support to our Fortune clients for more than 4 years.

If you need:

- training
- support
- maximum computer performance
- peace of mind relating to processing
- customization
- application results
- minimum errors and confusion
- better ideas on how to increase the power of your Fortune

Call Dan Borrowman at (314) 837-7111

SUPERIOR COMPUTER SYSTEMS, INC.
917 N Hwy 67, Suite 105
P.O. Drawer T
Florissant, MO 63031

Making computers work for you

Software Disk Owners: We need to explain that some of our disks can be installed through the Global Menu. However, our install script does **not** install any programs on the Global Menu. If you want to install a program on the Global Menu, you should use the **addglobal** program. This program can be found on the **D.C. Grabbag** diskette in the shells directory. We will publish more information on this in a coming issue of **/u/fortune** ***news***.

/u/fortune news
73 Jason Street
Arlington, MA 02174

/u/fortune* news

The Newsletter for Users of the Fortune 32:16 Computer March 1987/Volume 4 Number 3

The Fortune:Word Glossary

Printing Labels From Records Processing In Fortune:Word

by Richard Alan Dow

Richard Alan Dow is owner of Dow Information Services which is a technical publications firm located in Rhode Island and is contributing to this publication independently.

We've talked about a number of ways to use **Fortune:Word Records Processing**, but perhaps one of the most common is also one of the trickiest. If you maintain a list of names, whether members, customers, prospects, or whatever else, sooner or later you may want to print labels to do a large mailing.

Different Types of Labels

Making labels can be approached in many ways. There are automated processes which allow you to print your list on plain paper and then let specialized equipment cut, glue, and affix the label to the mailing piece. This is usually done by a mailing house, and they will tell you how to set up to print labels for their particular brand of equipment.

Smaller lists can be accommodated with hand-applied pressure-sensitive labels. The label comes lightly attached to a backing. After printing, it can be peeled off the backing and applied to the mailing piece. Pressure sensitive labels come in a number of sizes and formats, and your first step is to choose the right one for your purpose.

The most popular format is a single strip of labels on continuous form backing which will run through your printer. You can also get them with two labels side by side, or three, or four strips across the width of the backing. These are referred to as *one-up*, *two-up*, etc. Labels also come in a variety of widths and depths, and your choice here is determined by the amount of information you need to put on the label and the size (pitch) of the type you will be using to print. The most common label size, 3-1/2" wide by 15/16" deep, permits a four line address. It allows you about 32 characters per line at 10 pitch or 40 at 12 pitch (which is what I usually use for address labels). The labels

Making mailing labels is a tricky but common use of Fortune:Word

come packed in quantities which vary depending on their size and format. Obviously the price also varies; the 1-up, 3-1/2"x15/16" are the cheapest (cheaper even than smaller sizes) because of their wide use.

A word of caution about handling labels. Different brands adhere differently to their backings, and the weight and stiffness of the backing also varies. A label that sticks to the backing too tightly may be difficult to peel off when you are doing the addressing, while one that is too loose can come off and jam your printer. When you find a brand that works well, stick with it (pun intended). I also suggest that you use care when unloading labels from the printer. Rolling them

See **Glossary,** page 22

News From Fortune

Fortune has released some new products for high-end users. A 4-megabyte board has been released for the **Fortune Formula**, which boosts its current memory capacity to 16 megabytes. Using the **SCSI** ("Scuzzi") controller, a 145 megabyte drive can be attached to the Fortune 32:16 or Fortune Formula. In order to attach this drive to a Fortune 32:16, it is necessary to replace the current WD-06 hard disk controller with the SCSI controller. The SCSI controller also acts as an interface for the Tape Streamer, eliminating the need for a PIO, and freeing 1 expansion slot in the 32:16.

FOR:PRO 2.0 (See **/u/fortune** ***news*** Oct 1986) has been released. The new version supports up to 3.5

See **News,** page 2

In This Issue...

The BASIC Advisor–Error 103 and more...**Page 19.**

Global Menu Secrets–Did you you know these things?...**Page 17.**

The UNIX Directory–Part 8 in our series on shell scripts...**Page 9.**

UNIX Basics–Options: What they are and how to use them..**Page 13.**

Fortune:Word Glossary–Printing labels...**Page 1.**

News from Fortune...Page 1.

Are you up-to-date with your Fortune Software?...**Page 3.**

*/u/fortune *news* is published by The Cambridge Consortium, Inc. It is not affiliated with Fortune Systems Corporation.

/u/fortune *news*...

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz

Technical Editor
Scott Bradner

Advertising Manager
Mark Palmerino

/u/fortune *news*... is published 12 times per year by The Cambridge Consortium, Inc. Neither /u/fortune news... nor The Cambridge Consortium are affiliated with Fortune Systems Corporation.

Subscriptions cost $48 for one year within the U.S.A. For foreign subscriptions, add $27 each year for added postage and handling costs.

Back issues are offered at:

Vol 1, Numbers 1 - 3	$1/each
Vol 1, Number 4	$.50/each
Vol 1, Numbers 5 - 8	$3/each
Vol 2, Numbers 1 - 7	$3/each
Vol 3, Numbers 1 - 12	$4/each
Vol 4, Numbers 1 -	$4/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $10 each. Please add $5 per order for foreign destinations.

Although every attempt is made to insure that our contents are error-free and without risk, The Cambridge Consortium is not responsible for losses that occur due to correct or incorrect interpretation of **/u/fortune** *news*.

For more information, call or write:

/u/fortune *news*
73 Jason Street
Arlington, MA 02174
(617) 663-7700, (617) 646-5603

The text for this newsletter was prepared using a Fortune 32:16 for text input. The information was then transferred to an Apple Macintosh Plus with kermit and formatted for typesetting using the Aldus Pagemaker program.

UNIX™ is a trademark of Bell Laboratories, Inc. Fortune 32:16™, Fortune:Word™ and FOR:PRO™ are trademarks of Fortune Systems Corporation. Multiplan™ and MS-DOS™ are trademarks of Microsoft. IBM-PC™ is a trademark of IBM. Macintosh Plus™ is a trademark of Apple Computer. Pagemaker™ is a trademark of Aldus Corporation.

©1987 by the Cambridge Consortium, Inc. 3/87

News

continued from page 1

megabytes of memory and four CommA-6 controllers. It also supports the SCSI controller described above, has expanded hard disk bad block sparing, and an enhanced print spooler. We will be writing a review of this product in the coming months.

Fortune:Word 3.0 (See ***/u/fortune news*** Oct 1986) has been released as of the end of March. The new release has many new features including: Floating Footnotes, Improved Spelling correction, Print Documents from an Index, Sort Within a Document, Autosave, and Wildcard Searches. This product will also be reviewed shortly.

In order to obtain either product, contact your dealer. □

Field Service Notes from Fortune

Product: Multiplier Card

Description: Changes necessary for Multiplier to work in PC's with clock speeds over 6Mhz.

Corrective Action: The following upgrade will allow the Multiplier to run in machines with clock speeds up to 8Mhz.

Materials Required: one 74S05 I.C. (Fortune part #1003853-01) This chip is available from most electronics supply stores.

1. Separate the two boards from each other, being careful not to bend the pins of the connector which holds the two together. See Appendix A of the Multiplier user's guide for more information.

2. With an I.C. puller, pull the 74Ls05 I.C. from socket U9, and replace it with the 74S05 I.C.

3. Re-assemble the two boards and install the Multiplier into the PC, following instructions in the Multiplier user's guide.

Product: Graphics Coprocessor

Description: Firmware version 2.0 for the Graphics Coprocessor is not compatible with FOR:PRO 2.0

Corrective Action: Contact your dealer to get the new EPROMS necessary to use FOR:PRO 2.0 with the Graphics Coprocessor. □

Monthly Bug Reports

Periodically we receive reports about known bugs with Fortune software and hardware. We feel that the following may help some of our readers.

Problem: Multi-volume back-up failure (**cp** with **-B** option) Problem Reported: 3/86O/S Level: 1.8.1 and below. Work around available: Yes, Software Bulletin: #2/#16.
Status: Problem not resolved.

Problem: Canceling out of multi-volume tape back-up results in "Preamble incorrect" message when attempting to restore or display tape. Date Problem Reported: 8/26/86 O/S Level: 1.8-1.2.1 Streamer Tape software. Work around available: Yes. Target Fix Date: ASAP.
Status: Support can't reproduce problem. Don't cancel out.

Problem: In Basic, if one escapes to the shell, re-enters Basic and then uses an input statement, Basic will time out if the input statement is left indefinitely on the screen. Date problem reported: 9/3/86 O/S Level: 1.8.1.1.
Status: Problem reported in error. Problem could not be reproduced - this issue is considered closed.

Problem: All new 32:16's and spares with the 300 watt **Digi-power** power supply. When the cover of the system is removed for service you will have 30 to 45 seconds before the air flow sensor shuts off. When the system is running for testing, this can cause excess heating and severe damage to the system. Date of Problem: 12/20/86 Work around available: Y Target Fix Date: ASAP.
Status: Engineering is working on a way to bypass this problem during service. Temporary fix is to place a fan so the air will circulate over the air flow sensor. This sensor is located on the open side of the heat sink.

Problem: PIO Diagnostic will fail on test "e" when run on an SX mother board. Date : 2/18/87 O/S Level: 1.8.1.1 Work Around Available: N Target Fix Date: ASAP.
Status: To be fixed with the next release of Diagnostics.□

Release Levels for Fortune Products

Periodically we like to list the current release levels of Fortune Products. This helps you to stay current. Here is the table that we received from Fortune dated 2/15/87:

Fortune 32:16 line

Product	Release Level
SYSTEM SOFTWARE	
Single User FOR:PRO	1.8
FOR:PRO 1.8.1.1 Update	1.8.1.1
Multi-User FOR:PRO	1.8
FOR:PRO Development Utilities	1.1
Language Development Tools	1.8.2
Tape Streamer Software	1.2.1
Expansion Disk	1.0
Fortune:Link	1.0
Fortune:Windows	1.0.1
FIXIT	1.1
HP Laser Interface	1.1
FOR:FROG	1.0
LANGUAGES AND DATABASES	
C	1.7
Pascal	1.3
FORTRAN 77	1.3
Thoroughbred BASIC	6.5.12
CBasic/16	1.1
Basic Four Interface	1.0
MF LEVEL II COBOL	2.5
MF LEVEL II COBOL/ET	1.1.2
SIBOL Compiler	1.0
RUBIX	2.0
RUBIX-C Interface	2.0
R/M Cobol	2.1
VUE Basic	5.07
VUE Graphics Plotter	4.51
VUE MultiProjects	4.52
VUE Report Writer	4.52
IDOL	2.0
MIMIX	2.1
BAS/IDOL Update	1.0
Informix 4GL	1.00.05c
Informix SQL	2.00.05c
Informix ESQL/C	2.00.05c
Informix File It	3.00.05c
Informix C-ISAM	3.00.05c
FOR:PRO/PC Software	
FOR:PRO/PC Cold Boot Set	1.0
FOR:PRO/PC Coprocessor Setup	1.0
Fortune:Word/PC Coprocessor	2.1.2
Development Utilities (PC)	1.2
Fortune:Windows/PC	1.0.1
Interactive Terminal Emulator/PC	1.1.1
Fortune to Fortune Copy/PC	1.1.1
Fortune Keyboard Handlers	1.1
COMMUNICATIONS	
Fortune:Works (MS-DOS)	2.1
Fortune:Works (FOR:PRO)	2.1
Interactive Terminal Emulator	1.1.1
Fortune to Fortune Copy	1.1.1
RJE Emulator 3780	1.0
3270 BSC Emulator	1.2
3270 SNA Emulator	1.0
X.25	1.0
Tektronix 401X Emulator	1.1
VT/E Terminal Emulator	1.0
OFFICE AUTOMATION	
Fortune:Word	2.1.2
HP Laser Interface	1.1
Fortune to Wang Conversion	1.0
Multiplan	1.4.2
GRAPHICS	
Graphwriter System Disk	2.2
Graphwriter Basic Formats	2.1
Graphwriter Extended Formats	2.1
GSS-DRIVERS (FOR:PRO)	1.0.1
GSS-Metafile Interpreter	1.20
GSS-Plottalk (FOR:PRO)	1.02
GSS-DRIVERS (MS-DOS)	1.03
GSS-TERMINAL (MS-DOS)	1.02
GSS-C Binding (FOR:PRO)	1.0.1
CDI Sound Presentations (FOR:PRO)	1.04
Fortune:Works Graphics Upgrade	2.0
Graphics Spooler	1.0

continued on next page

Expand your Fortune® without spending a fortune.

New equipment costs big bucks. The Good Fortune Trader sells fully tested burned-in equipment, refurbished to Fortune® standards at prices that won't make waves with your cash flow. Call or write today for a listing of our specials.

THE GOOD FORTUNE TRADER

60 DAY WARRANTY

a division of Computer Services of Atlanta

(404) 392-9938

1534 Dunwoody Village Parkway/Atlanta, Georgia 30338

FORTUNE FORMULA PRODUCTS

SYSTEM SOFTWARE

FOR:PRO Cold Boot Set	3.0
FOR:PRO 32 User Upgrade	3.0
Tape Streamer Software	1.3
FOR:PRO Development Utilities	1.2
Fortune:Windows	1.0.1
Diagnostics	3.0

LANGUAGES 7 DATABASES

THOROUGHBRED BASIC	6.5.12
INFORMIX 4GL	1.00.05C
INFORMIX SQL	2.00.05C
INFORMIX ESQL/C	2.00.05C
INFORMIX FILE IT	3.00.05C
INFORMIX C-ISAM	3.00.05C

COMMUNICATIONS

Interactive Terminal Emulator	1.1.1
Fortune to Fortune Copy	1.1.1

OFFICE AUTOMATION

Fortune:Word	2.1.2
Multiplan	1.4.2

To simplify the process of determining what release level you have on your machine, the following table may be helpful. To use it, log in as root, find the product, and type the what command associated with it.

Command	Product
what /etc/init	Operating System
what /bin/od	Development Utilities
what /m/wp2/wp	Fortune:Word
what /m/mp/mp	Multiplan
what /usr/bin/dolphin	Fortune:Windows
what /usr/bin/gw	Graphwriter
what /usr/bin/makedb	Rubix
what /b/basic.psd	Basic
what /usr/bin/cc	C compiler
what /usr/bin/cbas	Cbasic
what /usr/lib/fcom1	Fortran77
what /usr/lib/cobol/rts5	Micro Focus Cobol
what /usr/bin/pc	Pascal
what /usr/bin/rmcobol	RM Cobol
what /usr/bin/ld	Language Dev. Tools
what /bin/cu	ITE
what /usr/lib/uucp/uucico	Fortune to Fortune Copy
what /usr/bin/vte	VTE
what /m/WPS/WPS.rbte	Doc. Conversion
what /b/BAS/CAP/CAPDA0	Accounts Payable
what /b/BAS/CAR/CARDA0	Accounts Receivable
what /v/BAS/CGL/CGLDA0	General Ledger
what /b/BAS/CGL/CGLUD0	Fixed Assets
what /b/BAS/CPO/CPODA0	Purchase Orders
what /b/BAS/CBI/CBIDA0	Order processing
what /b/BAS/CPR/CRPDB0	Payroll
what /b/IDOL/CUT/CUTUJB	IDOL □

UNISOFT INC.

Experts in Computer Consulting - Sales - Service

Presents

FORTUNE SYSTEMS HELP

We Provide:

- Expert Trouble-shooting
- Years of programming and accounting experience
- *Free Initial Consultation*
- Training Available for:
  - * BASIC/IDOL/BAS and Multiplan
  - * UNIX and Ext. Fortune:Word
- Hardware Repairs & Upgrades
- FORTUNE trained experts at a cost that will surprise you!
- *Trade-Ins Welcome*
- Specializing in:
  - * Hard Disk Trade-In
  - * Hard Disk Upgrades

Authorized Fortune Systems Repair Depot
Fortune Systems National Value Added Reseller (VAR)
CALL: (303) 671-8600 EMERGENCY HELP: (303) 649-8505

UNISOFT INC. 14100 East Jewell Avenue Suite 27 Aurora, Colorado 80012

System Errors - And What to Do About Them

When your system crashes it often gives a little message with a number which indicates what caused the crash. But what do those numbers mean and what should you do about them. In the following section we list the **System Error Messages** as found in **Fortune's Field Service Manual**. Also listed will be the recommended action. In most cases, the recommended action will be to contact your local dealer. However, we thought this list was useful in and of itself. We will also follow-up with more articles on what these error message mean.

1. Invalid block device

Contact your dealer.

2. Invalid device table entry

Contact your dealer.

3. Error in swap I/O

Contact your dealer.

4. IOCCOM

Contact your dealer.

5. No file system

Contact your dealer.

6. Timeout table overflow

Power on reset*. If problem does not clear, there is the possibility that there are too many work stations for the operating system. Contact your dealer.

7. No mount table

Contact your dealer.

8. Error reading in root inode

Contact your dealer.

9. Running dead process

Contact your dealer.

10. No more process table entries

Contact your dealer.

11. No more swap space

Power on reset

12. No more swap space

Power on reset

13. Unexpected trap

Power on reset. If the problem does not clear, contact your dealer

14. Kernal bus/address error

Record message number and all events immediately preceding the message, i.e., number of users, applications in use, etc. Power on reset. If problem does not clear, Contact your dealer.

15. Invalid ROM format

Contact your dealer.

16. Double parity error

System memory has failed. Contact your dealer.

17. Refresh timeout

Probably a hardware error. Contact your dealer.

18. The init file is dead

Contact your dealer.

19. Mfree overrun. Memory allocation error

Power on reset. Contact your dealer if problem does not clear.

20. zero wchan

Software error. Power on reset. Contact your dealer if problem does not clear.

100. Hardware failure

Contact your dealer.

120. Memory parity error

Contact your dealer..

121. Unable to log error

Contact your dealer. The system is not able to tell what is wrong.

122. Spare table overflow. Space unavailable. Hard disk and flexible disk overflow.

Contact your dealer.

123. Spurious interrupt. Unexpected hardware problem.

Contact your dealer.

124. Can not read configuration block.

Power on reset. If problem does not clear, contact your dealer.

125. Invalid magic number in configuration block.

Power on reset. If problem does not clear, contact your dealer.

128. Physio to an odd address Program error, Configuration is inaccurate.

Power on reset. If problem does not clear, Contact your dealer.

129. No more file table entries

Power on reset. If problem does not clear, increase approximate number of users on configuration menu by one.

130. No more inode table entries

Power on reset. If problem does not clear, increase approximate

continued on next page

number of users on configuration menu by one.

132. Address in inode 2 24
File system error

Power on reset. If problem does not clear, Contact your dealer.

133. Stuff Operating System error.

Contact your dealer

134. Invalid user access
Operating System error

Contact your dealer.

135. Error doing suiword
Operating System error

Contact your dealer.

136. Error in process queue

Contact your dealer.

137. No text table entries

Increase approximate number of users by one on configuration menu. If problem does not clear, contact your dealer.

138. Random Interrupts

Hardware problem. Power on reset. If trouble does not clear, contact your dealer.

139. Bad free count

File system error. Power on reset. If trouble does not clear, contact your dealer.

140. No more space
The disk is full

Power on reset. If trouble does not clear, remove some files from disk. If trouble does not clear, Contact your dealer.

141. Bad block

Power on reset. The File system is not properly configured.

142. No inodes.
The disk is full

Power on reset. If trouble does not clear, remove some files from disk. If trouble does not clear, contact your dealer.

143. Bad Count
The file system is not consistent.

Power on reset. If trouble does not clear, contact your dealer.

144. Hardware error.

Contact your dealer.

200. Hardware error.

Contact your dealer.

400. Software failure.

Contact your dealer.

800. Software failure.

Contact your dealer.

*Power on reset - Turn off system power. Wait approximately 10 seconds. Power on the system. □

We are the *largest* Authorized Fortune Dealer in the Pennsylvania, New Jersey and Delaware area. As part of the **P**rofessional **S**olutions **G**roup of companies with over 35 years in business automation, we have the *stability* that our customers demand

Our services include a factory trained field engineering team, a complete parts inventory including system upgrade capability and a software support department.

For all of your Fortune needs contact:

Mark Shields of PSG at
2550 Eisenhower Avenue
PO Box 762
Valley Forge, PA 19482

or you can call Mark at (215) 666-1400

Multiplan - How To Handle Zeros

by David Hinze

The following hints for handling zeros in Multiplan were contributed by David Hinze of ICT, Inc. in Shelby, MI. We very much appreciate his helpful ideas.

As a CPA, I stare at countless worksheets daily, I hate to see columns of formulas print out zeros (usually 0.00) when the value generated by a formula is 0. Especially when a worksheet is printed at 15 CPI, the numbers themselves are hard enough to read and one doesn't really need to strain to see if the number is 0.00 or 0.80.

As an example, I have a weekly payroll for over 50 employees. Most of these are hourly and are paid in tenths of an hour. For control purposes, I enter the hours worked from the time cards onto a Multiplan sheet. This worksheet cumputes the number of regular and overtime hours that each employee is paid for. As errors are sometimes made, I print out a portion of the worksheet showing name, regluar hours and o.t. hours and display it by the timeclock for employees to review. In this way, changes can be made before the payroll checks are made and eliminates 'adjusting the next check' for any misunderstandings. The rest of the worksheet that the employees do not see, computes gross pay which I can check against the computer figures from the earnings report or more easily by a simple report on the screen from file 157 (if you have and use IDOL). Checking figures insures that the data has been entered correctly and that the gross pay will be correct.

Three versions of the report are shown in Figure 1. In all cases, once the sheet is setup, all that is entered into the worksheet is the total hours worked in column 2. The program will generate the data in column 3 and 4 for the hours and the gross pay in dollars in columns 6 through 8.

In example 1, the formula in column 4 merely subtracts regular hours from total hours and in most cases displays 0.00.

Example 2 changes the column 4 formula to an IF statement that shows overtime hours if there are any and if not prints a blank cell. The result, unfortunately, is that the formula for overtime pay in column 7 produces a **#VALUE!** message as it is trying to multiply a value times a alpha character.

Example 3 cleans up the act! The formula for column 4 is unchanged but the formula in column 7 is changed so that it contains a **SUM** command. The sum command ignores alpha characters and what we do in effect is sum column 4 to itself! The formula for column 7 now reads: **IF(SUM(RC[-3]:RC[-3])<>0,ROUND(RC[-3]*RC[-2]*1.5,2)," ")** Note the " " equals quote/space/quote. □

EXAMPLE 1 - REGULAR SIMPLE FORMULAS

NAME	TOTAL HOURS	REGULAR HOURS	OT HOURS	RATE	REGULAR PAY	OT PAY	TOTAL PAY
ALICE	40.00	40.00	0.00	8.00	320.00	0.00	320.00
BILL	40.00	40.00	0.00	10.75	430.00	0.00	430.00
CRAIG	39.00	39.00	0.00	15.85	618.15	0.00	618.15
DAVE	40.80	40.00	0.80	15.85	634.00	19.02	653.02
JOHN	20.00	20.00	0.00	5.25	105.00	0.00	105.00
JOSH	40.25	40.00	0.25	10.75	430.00	4.03	434.03
	220.05	219.00	1.05	66.45	2537.15	23.05	2560.20

EXAMPLE 2 - LETS GET RID OF THE ZEROS AND REPLACE WITH BLANKS
OOPS!!!!!!

NAME	TOTAL HOURS	REGULAR HOURS	OT HOURS	RATE	REGULAR PAY	OT PAY	TOTAL PAY
ALICE	40.00	40.00		8.00	320.00	#VALUE!	#VALUE!
BILL	40.00	40.00		10.75	430.00	#VALUE!	#VALUE!
CRAIG	39.00	39.00		15.85	618.15	#VALUE!	#VALUE!
DAVE	40.80	40.00	0.80	15.85	634.00	19.02	653.02
JOHN	20.00	20.00		5.25	105.00	#VALUE!	#VALUE!
JOSH	40.25	40.00	0.25	10.75	430.00	4.03	434.03
	220.05	219.00	1.05	66.45	2537.15	#VALUE!	#VALUE!

EXAMPLE 3 - NO ZEROS WITH NO #VALUE! MESSAGES IN COLUMN 7
THE ONLY WAY !

NAME	TOTAL HOURS	REGULAR HOURS	OT HOURS	RATE	REGULAR PAY	OT PAY	TOTAL PAY
ALICE	40.00	40.00		8.00	320.00		320.00
BILL	40.00	40.00		10.75	430.00		430.00
CRAIG	39.00	39.00		15.85	618.15		618.15
DAVE	40.80	40.00	0.80	15.85	634.00	19.02	653.02
JOHN	20.00	20.00		5.25	105.00		105.00
JOSH	40.25	40.00	0.25	10.75	430.00	4.03	434.03
	220.05	219.00	1.05	66.45	2537.15	23.05	2560.20

** *Print Fortune:Word on Terminal Printer* **

See Terminal/Printer Link

SOFTWARE FOR THE FORTUNE 32:16

MULTIPLAN LINK - can pass data from BAS/IDOL/BASIC files to Multiplan and back. For example, it could be used to send data from any of the BAS files (Chart of Accounts, Customer Master, Inventory Master, etc). Allows selection of fields to be sent and provides logical retrieval and key range selection. **$295.00**

RECORDS PROCESSING LINK - can pass data from BAS/IDOL/BASIC files to Records Processing and back. For example, it could be used to take customer address data from the Customer Master and will automatically make the Records Processing List document for mailouts. Allows selection of fields to be sent and provides logical retrieval and key range selection. **$295.00**

ASCII LINK - can pass data from BAS/IDOL/BASIC files to ASCII and back. For example, it could be used to send data from any of the BAS files (Chart of Accounts, Customer Master, Inventory Master, etc). Allows selection of fields to be sent and provides logical retrieval and key range selection. **$295.00**

TERMINAL/PRINTER LINK - provides capability to print Multiplan, and BAS/IDOL reports on a printer connected to a workstation view any report on the terminal before printing, convert any report to FortuneL:Word format; accumulate multiple BAS/IDOL reports before printing, backup reports onto diskette; copy or rename report files; print multiple copies of the report on either the primary and/or terminal printer. Also has option to print BAS/IDOL reports on HP LASER printer ($50 extra). ** ***Also supports printing off auxiliary workstation with Fortune:Word*** ($50 extra) **. **$195.00**

TELEPHONE LINK - on line telephone message, telephone directory and inter-office mailbox/memo system which can also be used as a simple calendar reminder system. Provides control, printing and viewing of on-line data for each user. **$295.00**

FINANCIAL LINK - stores BAS Income Statement and Balance Sheet data in files. Once stored, it can be sent to Multiplan or instantly printed. **$195.00**

MENU LINK - provides software and documentation for adding items to the Global Menu as well as a sub-menu for your use Includes backup script for user directories and BAS/IDOL/BASIC programs and data. **$95.00**

CALCULATION LINK - provides programs for amortization, depreciation, loan repayment, averaging (with graph), linear correlation, and breakeven analysis. **$95.00**

KOMPACT PERSONNEL ACCOUNTING - provides for data capture of personal, wage/salary, job, education, salary/wage, appraisal training, dental, medical and life insurance data. Includes over 80 pre-defined reports/worksheets. **$495.00**

CRT LINK - provides information for using other terminals on your Fortune (WYSE, TV, IBM 3101). **$95.00**

AUTOMATIC REPORT LINK - provides capability to print numerous IDOL defined reports automatically based on frequency or application without having to select each from a different menu. You identify frequency/application codes. **$95.00**

PRINTER LINK - provides printcaps for Okidata 84, 2410, TI 855, and Anadex 6500. Includes shell scripts for changing pitch and LPI. Allows these printers to be used in word processing. **$95.00**

MODEM LINK - allows Hayes 1200 Smartmodem or compatible to be used on one port for both incoming and outgoing calls. **$95.00**

INSTALL LINK - documentation and software example for installing your product/menu through "S5". **$95.00**

REPORT LINK - modifies IDOL Report Generator Program to allow stacking of up to 8 fields in an IDOL defined report. Also corrects problem with displaying subtotals for all stacked fields. **$95.00**

WESTERN LINK - provides instructions on how to use your Fortune for sending/receiving telex messages using Western Union's Easylink. Can replace your telex! Fortune ITE or Handshake™ software is required. **$95.00**

UNI-KOMP 9803 Rocktree Houston, Texas 77040 (713) 937-7782

The UNIX Directory

Introduction to Shell Scripts - Part 8

by David E. Kloes

*David E. Kloes is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the Fortune Computer and is President of the Houston UNIX User's Group. He contributes independently to **/u/fortune news**.*

We start off this session with a clean slate. In Part 8, we will discuss a few remaining commands and concepts, including syntax errors and the **trap** command. We will complete the **Introduction to Shell Script** series next time by presenting an actual shell script which we will go through line by line as a summary of the commands and concepts we have presented.

SYNTAX ERRORS

If you have been entering the shell script examples we have given you throughout this series or if you have been trying to write your own shell scripts, you are probably very familiar with shell *syntax* errors. In this section, we will show you the most common errors made when writing shell scripts and also the associated shell error message. You have probably already noticed that the line the shell *thinks* has the problem is not necessarily the line that caused the problem. Here is a list of the more common shell programming errors:

1. Forgetting the word "test" when testing a condition in the "if", "while", or "until" command sequences.

2. Forgetting the appropriate ending sequence command in a command sequence (i.e. fi, done, esac).

3. Leaving out a matching ", ', or `.

4. Leaving out the ;; at the end of each "case" in the case command sequence.

Here's an example of the error in #1 above and the resulting syntax error:

Program: error.sh

```
ext="n"
while "$ext" = "n"
do echo "No errors!"
   exit
done

$ error.sh
error.sh: n: not found
```

In this example, the error message indicates that the shell is trying to run a command "**n**". Since we left out the word "test" after the "while", the shell tries to run the value of "**$ext**" (which is "n") as a command and cannot find a command called "n". Normally we can locate the line that has this type of problem by looking for a line that the shell could interpret as the command "**n**".

In this article, we will show you some common shell script errors and what to do about them.

In this next example, we will correctly insert the word "test" after the "while" and leave out the "done" at the end to show you the type of error message you will get in situation #2 above:

```
$ error.sh
error.sh: syntax error at line
5: `end of file' unexpected
```

This is the shell's way of telling you that it has reached the end of the shell script when it is still expecting something. In this case, it knows that the "while" command sequence should end with a "done" and doesn't find it. If we had the "while" and the "done" without the "do", the shell would tell us that the "done" was unexpected because it knows there should be a "do" and doesn't find it.

The following is the same shell script with the error indicated in #3 above:

Program: error.sh

```
ext="n"
while test "$ext" = "n"
do echo No errors!"
   exit
done

$ error.sh
error.sh: syntax error at line
6: `end of file' unexpected
```

In this case the shell finds an unmatched quote (on the echo line). Since it is still looking for the matching quote when the shell script is at the end, we get the "end of file unexpected" message again. We would get the same message for situation #4 above.

Depending on how the shell script is written, we may or may not get the same error messages above. Most of the time; however, the error can be traced to one of the above listed problems. As a rule, you should glance through your shell script and look for these obvious situations to find the source of the syntax error. It is also a good idea to write your shell scripts in small segments and test each segment out as it is completed. This narrows down the amount of code we have to look through to locate the source of a problem.

In many cases, it is not a syntax error that causes us problems, but it is a **logic** error. While the shell script may be syntactically correct, it simply does not do what we asked it to do. Finding the source of this type of problem can sometimes be aided by using the "**sh -v**" (verbose) command to execute our shell scripts. The verbose option will show us each line of the shell script as it is executed.

Continued on next page

Figure 1 is an example of a logic type error.

Our intention when we wrote this little script was to insure that an error message was given if the user entered a value less than 1 or greater than 99. The script is syntactically correct; however, we would never get an error message because we used the **-a** operator instead of **-o**. We could never get a response that less than 1 **AND** greater than 99. We wouldn't get a syntax error, our shell script just would not work right. You should always check out your shell script logic so that you can find these types of errors. If we are trying to test out whether a particular command or command sequence works or not, it is sometimes easier to enter the commands at the shell prompt before we put the logic into our shell script (remember the **"PS2"** prompt we talked about earlier on?). Here's an example:

```
$ for count in 1 2 3 4 5
> do echo -n "$count "
> done
1 2 3 4 5 $
```

Figure 1.

Program: logic.sh

```
echo -n "Enter a number from 1 to 99: "
read num
if test $num -lt 1 -a $num -gt 99
  then echo "Invalid number entered"
fi
```

Figure 2.

```
echo -n "Enter today's date: YYMMDD^H^H^H^H^H^H"
read tdate
if test `expr $tdate : '.*'` -ne 6
   then echo "Enter six digits for the date."
fi
```

Figure 3.

Program: main.sh

```
trap 'main.sh' 2
echo -n "^L          Main Menu

     1.  List Current Directory.
     2.  Go to Home Directory..
     3.  Exit Menu.

     Enter selection: "
read ans
case $ans in
  1) ls | more;;
  2) cd;;
  3) ext="y";;
  *) echo "Invalid Selection.
Depress <RETURN> to continue: "
     read bunk
     main.sh;;
esac
```

LOGGING A USER OUT IN A SHELL SCRIPT

There are times when we are writing shell programs when we will want to log a user out of the system. An obvious place to do this would be when we asked for a password and the user does not type one in correctly after a given number of tries. If we execute subsequent shell scripts from our main shell script, we might want to give the user the option to log off without having to step his way back through the menu system. To do this, execute the command **"kill -9 0"** and all of the user's processes will be killed and he will be logged out of the system.

TESTING FOR THE LENGTH OF A VARIABLE

Occasionally, we will want to test an operator's input to insure what they enter is a certain length. We can use the **expr** command to do this kind of test. Basically, the following would return the length of **$entry** (or whatever other variable we want to test):

```
expr $entry : '.*'
```

A typical use is shown in Figure 2.

THE *trap* COMMAND

The **trap** command is used to execute a command or sequence of commands if certain signals are detected. All of the different kinds of signals and their associated number can be found under the **signal** command. The one we are particular interested in for our shell scripts is the *interrupt* (signal number 2). This signal is sent when the **<CANCEL/DEL>** key is depressed.

Remember in our last article, we said that one way to handle this key in our shell scripts was to change the interrupt using the **stty** command to some other control character sequence. We can also "trap" this signal in our shell scripts so that some command is run whenever it is depressed. The format of the command is:

```
trap 'exit' 2
```

This says that if the interrupt key

Continued on next page

(CANCEL/DEL) is depressed, "exit" the shell script. The number 2 indicates the signal that we are trapping. We could have used any command (or multiple commands separated by ;) in place of the "exit" and we could have put in multiple signal numbers.

Let's say we want to write a shell script menu where the user is returned to the "main menu" whenever the **<CANCEL/DEL>** key is depressed. This would allow us to have sub-shells and we can use the interrupt key to have the user return to the "main menu" script regardless of how deep into our sub-shells he goes. We do this to keep total control of our shell scripts and as a result normally provide an "exit" selection on any menu or prompt we display. We also want an error routine so that if the user enters an invalid selection, he will get an error message and the menu screen will be redrawn. Take a look at Figure 3.

There is a major problem with this shell script. We give you this problem in order that we may state what we will call the *golden rule* in shell programming:

NEVER, NEVER, *NEVER* RUN A SHELL SCRIPT FROM WITHIN ITSELF!

Since a shell script is a process, if we run a shell script from within itself (as we have here in the **trap** and in the ***)** lines of this shell script), we are actually building multiple processes on top of each other. As an example, let's say a user enters two incorrect selections in the above shell script. He will get the error message and then "main.sh" is executed again each time he enters an incorrect response. In this case, not only is the original menu process running but two other menu processes are running also. Now if he correctly selects the "Exit Menu" selection, what will happen? The answer is that he will exit from the current menu process but previous menu processes are still running. In effect, what happens is that the menu screen will be redrawn as the shell takes him back to the previous menu process instead of exiting him entirely. In fact, he would have to select "Exit Menu" several times to completely back out of all the processes. Let's see if we can look at this graphically by looking at Figure 4.

The *trap* command is used to execute a command or sequence of commands if certain signals are detected.

After the second invalid selection, the active script that is running is the 3rd process. If at this point the user selects the "Exit Menu" (3) selection, the active script that will be running will be the 2nd process. What we have is a mess and a thoroughly confused user. Since each user has a limit on the number of processes that can be running and since the system has a process limit for all users, it may not be long before the system is "hung-up". Also the more processes that are running, the slower the system will run. Therefore, even though you may not understand the concept of what is happening here, if you follow the golden rule, you will avoid this situation.

There is another problem with this shell script. By using the **trap** command in the main shell script, we will get some interesting and confusing results if the interrupt (**CANCEL/DEL**) key is depressed. First of all, the user will have to depress the **<RETURN>** key in order for the interrupt and the **trap** to be executed. Second, error messages will appear when we don't expect

Continued on next page

Figure 4.

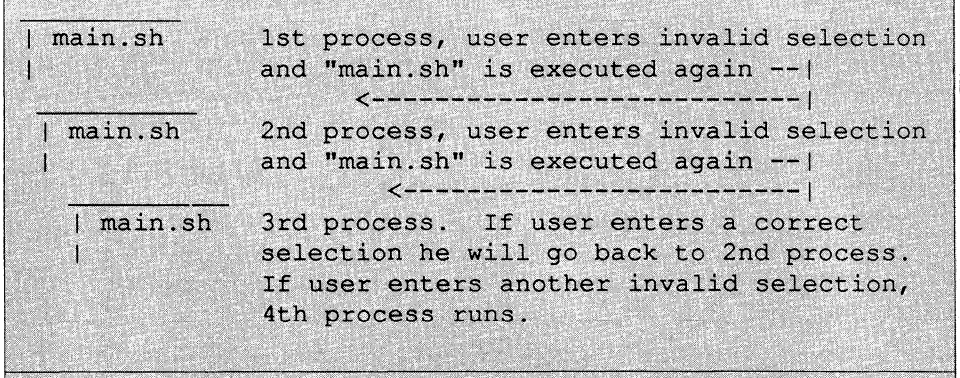


Figure 5.

Program 1: main.sh

```
ext="n"
while test "$ext" = "n"
do echo -n "^L      Main Menu

     1.List Current Directory.
     2.Go to Home Directory..
     3.Exit Menu.

     Enter selection: "
   read ans
   case $ans in
     1) ls | more;;
     2) cd;;
     3) ext="y";;
     *) echo "Invalid Selection.
Depress <RETURN> to continue: "
        read bunk;;
   esac
```

Program 2: trap.sh

```
trap 'main.sh' 2
main.sh
exit
```

them because the previous shell is reacting to the **<RETURN>** key. Try entering this shell script and running it to see the confusing results. We give you this as an example of how *not* to use the **trap** command.

Let's rewrite this shell script to get rid of these problems by correctly using the **trap** command and putting our "main.sh" script in a "while do done" loop instead of running it from within itself. Figure 5 lists these two shell scripts.

By using these two scripts, we have solved both of the problems.
First of all, we start the shell scripts by executing "trap.sh" first. It sets the "trap" and runs "main.sh". Now if the user depresses the interrupt **<CANCEL/DEL>** key, the "main.sh" process is killed and control is returned to "trap.sh" which re-runs "main.sh" as a new process. This in effect will redraw the menu screen and the **<CANCEL/DEL>** key will work without having to depress the **<RETURN>** key. Regardless of how deep we are into sub-shells, depressing the interrupt key will kill all script processes and return control to "trap.sh". This allows us to use the interrupt key as the "kill" key throughout the main script and all subsequent scripts. Whenever the key is depressed, the user will return to the main menu.

Secondly, instead of re-running the "main.sh" script from within itself, we now have the whole script put into a "while do done" loop. This means that the only way the user can actually exit the script is by selecting option 3 (Exit Menu). When this selection is made, **ext** is set to "y" and control returns to "trap.sh" which "exits" back to wherever "trap.sh" was called from.

This concludes the material we will be presenting for this article. Next time we will summarize the material by giving you a fairly comprehensive shell script that will serve as a review of the commands and concepts we have discussed. □

Tri-Tech Computer Services, Inc.

Tri-Tech Computer Services, Inc., with offices in Newton, MA and Kensington CT, provides local on-sight service as well as nationwide depot service by Fortune Trained, Fortune Authorized personnel.

In addition to technical support, **Tri-Tech** offers a variety of customer services.

New Systems
On-Sight Service
Consulting

Office Supplies
Laser Printers
Communications

Word Processing
Spread Sheets
Accounting

Pre-Spring Discounted Specials:

Micropolis 70MB Hard Disk Upgrade (requires WD06 and 1.8 UNIX) *Call for prices.*
CDC 30MB Hard Disk Upgrade
Fortune:Word to Extended Fortune:Word Upgrade
Fortune FIS1000 Workstations (Refurb)
Genicom 3404 Printers (Dot Matrix 180 to 500 cps - Refurb)

(All Equipment sold by Tri-Tech carries a 90 day Warranty)

COUPON #1

10% off the total cost of an Onsight or Depot **Maintenance Agreement.**

(expires 5/31/87)
call for quote.

COUPON #2

10% off office **supplies** on purchase up to $100.00 or 15% off office **supplies** and/or software on purchase over $100.

(expires 5/31/87)

COUPON #3

$30.00 per hour labor rate for On-Sight Service
(reg. $80.00 per hour) on labor services for the following: Computer Repair, **Upgrades and add-ons**, Consulting and Cabling

(expires 5/31/87)

* This coupon special is offered by **Tri-Tech Computer Services, Inc.** and does not apply to items on sale or already discounted. Void where prohibited by law.

Call **TRI-TECH COMPUTER SERVICES, INC.** Today!!!
Massachusetts (617) 965-4453
Connecticut (203) 828-9784

UNIX Basics 4

Options & Arguments

including Common Options for ls and cp

In the first three parts of our series on UNIX Basics, we discussed, in a basic manner, several useful UNIX commands. In our last installment, we spent some time explaining an important concept about UNIX - its file system. In this article, we want to explain several more concepts.

Options - Modifying the Effect of A UNIX Command

First, we want to talk about *options*. Sometimes options are called flags, switches or arguments. Regardless of what they are called (and we will call them, in general, options), they modify how a particular UNIX command works.

Let's back up a little and talk about the structure of a UNIX *command line*. The following is an example of a command line:

```
$ ls -sail /u/joe /u/sarah
```

This line has a certain structure. First, this line has four elements (if we ignore the $, which represents the UNIX prompt). Each of these elements, in UNIX parlance, is called an *argument*. An argument is a series of characters separated by spaces. Thus, the four arguments on the above command line are as follows:

```
1) ls
2) -sail
3) /u/joe
4) /u/sarah
```

I think you will recognize the first argument as the name of the particular UNIX command that we are invoking, the **ls** command. Although this is obviously the first argument, it is actually numbered argument **0** (zero). This first argument, the 0th argument, is very important to the UNIX operating system because it is the name of a command that the operating system will attempt to find in its list of valid commands. You see, UNIX will "look" for the command **ls** and if it finds this command, it will start it running. If this is successful, then it will be the **ls** command that will process the rest of the arguments you have typed (in this case, -sail, /u/joe, and /u/sarah).

Thus, the 0th argument should always be the name of a valid command, that is, a command that the UNIX operating system can find. Now, computers are very precise but we humans often are not. Sometimes I will be trying to help someone out over the phone with their computer. I might say something like, "Ok, issue an ls dash sail command." Now, I meant, of course, that the person should type "ls -sail". However, what they heard caused them to type **ls-sail** - which, by the way, is a reasonable interpretation of my verbal description). Nevertheless, UNIX will interpret **ls-sail** as the name of a command (with no spaces, it sees ls-sail as the 0th argument) and will try to find this command. Since there is no command by this name, the operating system will, in all likelihood, respond with

```
ls-sail: command not found
```

which is UNIX's way of saying it couldn't find the command called **ls-sail.**

-sail: Some Options for the ls Command

Recall our original example: **ls -sail /u/joe /u/sarah**. We've discussed the **ls** part. The second argument, the *dash sail* part, is what we are trying to get at in this discussion. This is a special type of an argument which we call options. Options are used to modify how a UNIX command works, and often, although this is not always the case, options are preceded by a dash (-). The dash is often used as a signal to the command (in this case, the **ls** command) that the following letters are options. Thus, the dash distinguishes this argument from other arguments on the command line. In our example, then, the **-sail** is a different type of argument than /u/joe and /u/sarah.

To understand this, let's review what the **ls** command does. The **ls** command lists out the names of the files in the directory that we are in. For example, the output from an **ls** command on joe's directory might look something like that shown in the top portion of Figure 1 (shown on page 14). As you can see, this simply lists out the names of the files - it tells us nothing else.

On the other hand, look at the bottom portion of Figure 1. This is also a directory listing of joe's directory. However, much more information is displayed. What caused the difference between how **ls** worked in the top portion as compared to the bottom portion of Figure 1 was the **-sail** argument.

To delve a little deeper into options, I want to pose a question. Would the following commands work the same way?

```
1) ls -lisa /u/joe
2) ls -sail /u/joe
3) ls -sa -il /u/joe
4) ls -s -a -i -l /u/joe
```

Continued on next page

The answer is, yes they would all work the same. If you are puzzled by this answer, good! However, don't despair.

There is nothing magical about the **-sail** part of the **ls** command. It doesn't stand for something like "sail through this directory listing." In this case, sail is composed of four separate characters, each of which is a *different* option for the **ls** command. Because this is the case, it doesn't matter at all if we type sail, or lisa, or ails, or any other permutation of these four letters.

The **Introduction to FOR:PRO** book lists 19 separate options for the **ls** command. The four that we have chosen to use are described as follows:

s - Give size in blocks, including indirect blocks, for each entry.
a - List all entries; usually all files beginning with a '.' are suppressed.
i - Print i-number in first column of the report for each file listed.
l - List in long format, giving mode, number of links, owner, size in bytes, and time of last modification for each file. If the file is a special file, the size field will instead contain the major and minor device numbers.

Ok, so now we know why **ls -sail** works the same as **ls -lisa**, but what about **ls -sa -il** or **ls -s -a -i -l**? The answer to this question lies in the use of the dash. Recall that I said that the dash is generally used to tell the command that what follows is an option. Since this is the case, it does not matter if we type **-sail** or we type **-s -a -i -l**. It is really all the same to the **ls** command, that is, it recognizes these four letters as options and then modifies its behavior accordingly.

To generalize and summarize, we have made the following points. First, arguments that begin with a dash are almost always options. They modify how the command performs. The exceptions to this rule are so infrequent as to not be important. You will learn about these peculiarities as you need them (e.g. in the command **rm - -junk**, the **-junk** is not a set of options. It is the name of a file).

Second, options must begin with a dash. This may sound like a restatement of the first point but it is different. The point here is that **ls -sail** will work very differently from **ls sail**. However, be aware that this is a generalization and that for some UNIX commands, options do not have to be preceded by a dash (e.g. **ps lax**).

Third, many UNIX commands have options but not all of them do. Some examples of commonly used commands that do not have options are **cd**, **pwd,** and **mkdir**.

Common Options for ls and cp

In this next section, we will speak of some common and, hopefully, useful options for the **ls** and **cp** command. We have already discussed the difference between the **ls** and the **ls -sail** commands. Figure 1 shows how they work and how the outputs are different. The **ls** command, without

continued on next page

Strategic Networks
Authorized Fortune Master Dealer

Dedicated to providing Fortune Users with:

- Systems Consulting
- Programming Services
- Equipment Sales
- Hardware Maintenance

A *full* staff of experienced Sales, Software and Service Personnel is ready to serve users in Connecticut and New York.

Please give us a call today.

Strategic Networks
360 Bloomfield Ave. Windsor, CT 06095
203 683-1070
Jack Runquist, President
Chris Liston, Sales Manager

any options, is useful if you want a quick listing of the names of the files in your directory. However, if you want a "longer," more complete information, then you must use some options. One of the most useful options is the **-l** (this is the "dash ell" option). As explained above, the **-l** option gives more information including the *mode* of the file, the number of links to other files, the owner of the file, the size of the file in bytes (for text files, this is equivalent to the number of characters in the file), the time of last modification of the file, and, finally, the name of the file. So, take a look at Figure 2.

Each line of this output contains the information for a file. The first ten characters of the line is the **mode** of the file. For example, for the file called bunix4, the mode is **-rw-rw-r--**

This basically codes the protections on the file and controls who can look at and write to the file. We won't get into an extended discussion of these protections in this article but if you are interested in learning more about this subject, we refer you to Volume 1 Number 5 of **/u/fortune *news***. Also, Chapter 3 in the **Introduction to FOR:PRO** book talks about file permissions.

Notice, however, the mode for the second file called **fred**. It starts with a **d**. This indicates that this "file" is really a directory.

The number following the mode represents how many *links* there are to this file. We will talk more about links below. The next entry on the line is the *owner* of the file. In our listing, the user **root** owns these files. The next entry is the number of bytes in the file. For the file called bunix4, we see that it contains 7,520 bytes. The next section is the time of last modification. The file bunix4 was last modified on March 23 at 9:43 A.M. Since a year is not listed, we know it was the current year, or in this case, 1987. Finally, as is obvious, we have the name of the file.

The very first line of the output for this command, in this case **total 11**, is the total number of blocks being used by this directory. So, the **ls -l** form of the ls command gives you some very useful information. In fact, it is such a useful set of information that there is a special UNIX command that does just this. It is called the **ll** command.

Figure 1.

```
$ ls /u/joe
col_letters
finances.MP
notes
personal
phonenum

$ ls -sail /u/joe
total 4
 4531  1 drwxrwxr-x 4 root    128 Mar 22 15:58 .
  109  1 drwxrwxr-x63 bin    1024 Mar 22 14:18 ..
 4527  1 drwxrwxr-x 2 root    128 Feb 22 14:12 col_letters
 4525  0 -rw-rw-r-- 1 root      0 Feb 22 13:57 finances.MP
 4529  0 -rw-rw-r-- 1 root      0 Feb 22 13:57 notes
 4337  1 drwxrwxr-x 2 root     64 Feb 22 14:14 personal
 4522  0 -rw-rw-r-- 1 root      0 Feb 22 13:57 phonenum
```

Another useful flag is the **-F** flag. The output from the **ls -F** command is just like the list you get from the ls command except that directories are marked with a trailing / and executable files are marked with a trailing *. This makes it easy to locate directories and programs in your directory. The following output shows what the **ls -F** command gives:

```
bunix4
bunix4.fig1
bunix4.fig2
fred/
tfile
tfile2
```

Notice the file called **fred**. It is listed as **fred/** which indicates that it is a directory. Also, none of the other files end with an asterisk (*), therefore there are no executable files in this directory. Just like the ls **-l** command, the **ls -F** command is considered so useful that there is special command for it as well. It is called the **lf** command.

The last version of the **ls** command that we wish to discuss is the **ls -tlr** command. Normally, when your files in your directory are listed, they are listed alphabetically by name. That is, they are sorted by name. The **-t** option instructs the **ls** command to sort the files not by name but by time modified, with the most recently modified file first. We already know what the **-l** option does. Finally, the **-r** option instructs the **ls** command to reverse the order of the sort. Thus, in this context, the listing will give the most recently modified files last.

So, what is the **ls -tlr** command good for, anyway? We often use it when we have forgotten the name of a file that we know we have created relatively recently. When we look at the output from this command, we see at the bottom of the list the three, or four, or five most recently modified files and this is usually enough to refresh our memory.

continued on next page

Figure 2.

```
total 11
-rw-rw-r-- 1 root       7520 Mar 23 09:43 bunix4
drwxrwxr-x 2 root         48 Mar 22 14:29 fred
-rw-rw-r-- 3 root         10 Mar 22 14:13 tfile
-rw-rw-r-- 3 root         10 Mar 22 14:13 tfile2
```

Some Useful Options for the cp Command

We described the **cp** command in Volume 4 Number 1 of **/u/fortune *news***. It is the UNIX command that is used to copy files. In that article, we did talk about some options. For example, the **-i** option is used when you want to protect yourself from accidentally overwriting a file. If you use **cp -i** and a file is about to be overwritten, **cp** will inquire whether or not you really want to overwrite that specific file.

There are three other useful flags that may help you. The first one is the **-t** option. What the -t option does is give the copy of the file that you are creating, assuming you own the file or are a superuser, the exact same modification and access times as the original file. Now, you may ask, "What is this good for?" When you are making backups of your files on a floppy, it is a good idea to make the backup copies have the same dates as the originals. This is because the modification and access times are often critical information when you later try to pinpoint an exact version of a file.

Normally, the **cp** command is very silent about its operation. When it runs, it hardly tells you anything. Sometimes, however, you want to have some indication of how successful or unsuccessful your copy has been. This is especially true when you are copying many files from one place to another. In this case, it is quite possible that some files will be successfully copied and others won't. The **-V** option instructs **cp** to be *verbose*. It prints a line on your terminal noting each copy operation and the status of that operation.

The last option we will talk about in this article is the **-I** option. The **cp -I** command doesn't really make a copy of a file but rather creates what is called a *link*. To understand what a link is we need to explain what an *i-number* is. Every file in your computer has a special number associated with it and this number is called the **i-number**. The i-number, simplistically speaking, points to the actual contents of the file. When you name a file, you are actually creating a link between an i-number, an actual physical number the operating system needs, and the name of the file which we humans like. Would you like to see the i-number for the files in your directory? Well, try the **ls -i** command, which we've done on our directory and which output we list:

```
4717 bunix4
4720 fred
4670 tfile
4670 tfile2
```

The i-number for **bunix4** is 4717. However, notice the i-number for **tfile** and **tfile2**. The i-numbers are identical. This indicates that these files are linked together. In essence, the **cp -I** command creates a link between another filename and an already existing i-number. You can have as many filenames associated with an i-number as you like.

Sometimes it is more convenient to link files together rather than make separate copies of them. Generally, you will want to link files together when you want several people to have access to the same file so that everyone will be able to see changes made by any one individual. For example, in our article about an electronic rolodex, we suggested that if you wanted to set up a central rolodex file that was accessible to several people, then you would link the rolodex file to files in each one of the other people's home directories.

Conclusion

In this article, we discussed the general notion of UNIX commands and their options. Options are arguments that modify how a particular UNIX command operates. We gave examples of some options for the **ls** and **cp** command and spent a little time talking about what we consider to be helpful and useful options that you might want to know about. If you feel we skipped a particularly useful option for the **ls** or the **cp** command, drop us a note and we'll share it with our readers.

Mark Palmerino

Vantine Business Systems

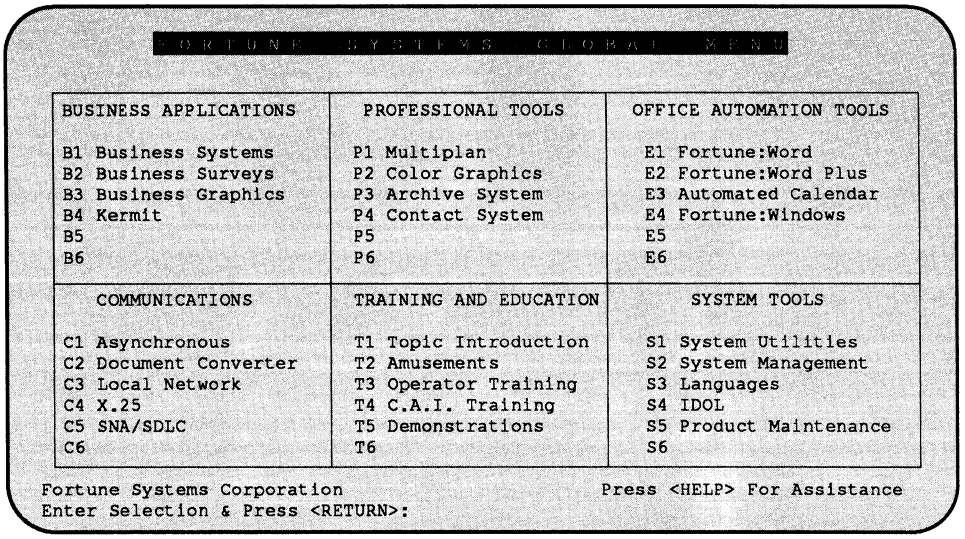
- **System Sales**
- **Consulting**
- **Support**
- **Onsite and National Depot Service**

Over Three and half years supporting Fortune Users

Vantine Business Systems
70 Walnut Street, Wellesley, MA 02181
(617) 239-8084

Authorized Fortune Systems Dealer

Behind The Scenes At The Global Menu



Any program worth its salt has features that are hidden deep within its soul. If the programmer was really clever, he makes sure that some of these capabilities are undocumented -- left to the truly inquisitive to discover. It turns out that the Fortune Global Menu, as straightforward as it appears to be, falls into this class.

Dr. Mark Citron, of the Children's Hospital in Los Angeles, recently called to reveal some of these secrets to us. Most of these hidden features are part of what we call *dot* commands (because they begin with a period, or "dot"). Many people know that if you type **.unix** from the Menu, you will get a UNIX shell. However, who knew that **.0** would return you to the Global Menu, no matter how many layers into it you had gone. For instance, if you choose System Utilities (**S1**), and then **15** to list a file's contents on the screen, and then you decide that you made a mistake and want to go directly back to the Global Menu, just type **.0**. This avoids backing out one menu at a time the same way you came in.

Help for modem users

Perhaps you sometimes call your Fortune over a 1200 Baud modem line and wish that you didn't have to wait forever while the menu is drawn on the screen. If you know the selections you want to make, you can use the .nomenu command. This erases the menu, and just leaves the "Enter Selection & Press <RETURN>" prompt. This works for the other menus in the Menu system as well. (It does not work for Fortune:Word or Multiplan Menus, just the sub-menus off of the Global Menu). By typing **.menu**, you will restore the menu.

Combining Selections on one line

You may already know that if you know all of the choices you need to make, you can combine them on a single line. For instance, to format a floppy from the Global Menu, you can type **S1 32 yes** all on one line. As soon as you press **RETURN**, the computer will begin to format the floppy. Be sure to put the blank floppy in the drive and close the door before hitting **RETURN**.

Using the Global Menu with Non-Fortune Terminals

We've said many times in the past that the Global Menu and **Fortune:Word** will only work with Fortune Terminals. It turns out we were wrong. The Menu will work with other terminals. You can either define the port for another terminal through the **Define Device Connections** menu, or you can even do it on the fly with the **.term** command. Suppose many people use a modem to log into your machine. One person has a Fortune Terminal, another a Mac, another an IBM-PC and the last person has a Model 100. If they log in and get the Global Menu it will look like a mess. If they issue the **.term** command, the computer will say "Please enter terminal type". Once this is entered and **RETURN** is pressed, the Global Menu will be redrawn for that terminal. Some terminals work better than others, but most are at least workable. In order for this to work, you'll need to know the terminal codename for your terminal, e.g. vt100, h19, FT, etc. If you don't know, give us a call and we may be able to help (617 663-7700).

As far as we know **Fortune:Word** is still restricted to Fortune Terminals, but this may change in the near future. **Multiplan** will work with other terminals.

HELP, you say

You can obtain a list of the dot commands by entering **.?** or **.HELP** (HELP is the HELP key). The mail and memo commands are not implemented.

You can also obtain quite a lot of help about the Global Menu selections by pressing the **HELP** key after you have selected any of the options.

Entering UNIX commands from the Menu System

While you are in the Global Menu System, you may enter a UNIX command by entering !command, where you substitute the command name for the word command. For example, you can get a shell by typing **!sh**, or use the rolodex by typing **!rolo** (if you have the Electronic Rolodex on our **UNIX Tools** diskette).

Doubtless there are other shortcuts that we don't know about for the Global Menu as well as other programs. If you would like to share some secrets, let us know.□

Introducing:

Simply the most comprehensive, extensive, and reasonably priced service arrangement available for Fortune Systems.

Period.

You've heard the promises, signed on the dotted line and shelled out the cold hard cash. But -- and let's be honest -- has the service arrangement you're currently under really lived up to the promises, to your expectations . . . to your needs?
Well, Micro Resources, Ohio's leading Fortune Systems dealer, backed by our affiliates, offers Ohio Fortune users a service arrangement package that simply cannot be beat. With factory specified hardware upgrades worth up to $2,390 -- At No Cost! -- a free subscription to /u/Fortune News, free monthly software service support, and 20% discounts on all subsequent purchases of Fortune System released software and hardware. And there's more.
If your Micro Resources agreement is negotiated by April 15, 1987, we'll extend our standard one year contract to cover the time remaining on your current contract, for up to one additional year . . . FREE!
Best of all, you get the premier Fortune Systems service experts. No dealer in the nation has sold or serviced more Fortune Systems. That means faster, more complete service at a reasonable cost than any other dealer in the state.

Period.

So call Keith Lamb today at (614) 766-2335, and let's talk about bringing your Fortune System up to speed.

Micro Resources
535 Metro Place South
Dublin, Ohio 43017
(614)766-2335

Authorized Fortune Systems Dealer

The BASIC Advisor

Ray Wannall is President of BaSiC Software Corporation located in Baltimore, MD, and is contributing his advice and helpful comments to this publication independently.

Question: *I have a client who is getting an error 103 when he tries to use Fortune:Windows with his Business BASIC programs. You know as well as I do that an error 103 is almost impossible to trace. Since I do not have Windows on an in-house system I need to ask you how they work with BAS and IDOL. For example, does each window open a new terminal in BASIC?*

Answer: Don't you just love error 103? The system is saying, "Hey, there's something wrong and I'm not going to tell you what it is!" The documentation isn't much better: "Contact your dealer".

As I am writing this (on **Fortune:Word**) I have a total of seven windows open. My base window is on this document. I first opened three windows for Business Applications (selection b1 off the global menu). I then opened two more windows and entered **IDOL** (s4) in each one. Finally, I opened number seven and entered Languages, Business BASIC (s3 1). In each window running a **BASIC** function, I went into the **BASIC** console mode and typed **PRINT FID(0)**. (For those who are unfamiliar with **BASIC** commands, this returns the identification of the terminal from which the command was entered.) Here is what I got:

```
Business Applications #1:  T0
Business Applications #2:  T1
Business Applications #3:  T2
                IDOL #1:   T3
                IDOL #2:   T4
           Languages #1:   T5
```

As you can see, terminal numbers were assigned in the order of my entry into **BASIC** functions, and the computer kept track of the last number used even though I went from **BASIC** to **IBASIC** to **DBASIC**. Quite frankly, I was surprised it did this considering that each global selection carries its own interpreter and ipl file set.

I have not yet experienced an error 103 in any window, but this may be because of my system configuration. I have a 20 MB 32:16 with one megabyte of memory and multi-user installed, and there are no other users logged onto the system at this time. It is probable that your client is trying to push a smaller system too hard, (i.e., too many users with too little memory). If this is not the case, I guess you'll have to "Contact your dealer".

While we are on the subject of **Fortune:Windows**, I want to inject a few comments and observations. I have been using this product for a little over a year now and have found it very helpful, especially when composing technical documentation. But it has not been working exactly the way I was told it would. I first saw a demonstration of Fortune:Windows back in August of 1985. I was told that one user could conduct a sort in one window and a long print job in another while working on an entirely different project in a third, all from one terminal. In fact as many as eight windows could be opened and actively working at the same time. However, I have found that when I leave a program running in a window, the work in that window is suspended until such time as I re-enter it. (Maybe this only happens with the BASIC programs. I have not tested For:Pro commands). (**Editor's Note**: *We have noticed this as well. However, it seems that a process is suspended **only** when the process tries to write to the terminal. If the process doesn't need to write anything to the screen, it should not be suspended by Fortune:Windows.*) Nevertheless, Fortune:Windows is extremely valuable for it's ability to transfer data from one window to another. I recommend it to anyone who uses the Business Applications and/or **IDOL** in conjunction with **Fortune:Word** or **Multiplan**. Just don't get carried away with it. Too many users coupled with too many windows can make working with a pencil and paper a very attractive alternative to electronic data processing.

Question*: I have a complaint about the Accounts Payable Vendor Invoice Entry. Once you enter an invoice into the computer you have the option of either paying or not paying the entire invoice. On many occasions I need to be able to make partial payments on invoices. Many vendors send invoices for entire orders even when half of the items were backordered and not shipped.*

Answer: The problem occurs because the open accounts payable file is accessed by Vendor Number plus Vendor Invoice Number plus Invoice Date. Technically this means that the system can hold only one of the same invoice number per vender per day. But when the invoice entry program checks for open invoices for the current vendor being entered, it looks to the vendor number and invoice number, not the date. Therefore, you cannot have the same open invoice number more than once for a vendor.

There are two ways to get around this restriction. The obvious solution is to not enter the invoice until such time as all items have been received (causing the vendor to send you a past due notice with horrendous interest charges for the items shipped). A better way to do it is to assign your own supplementary character to the vendor's invoice number and enter the invoice as two invoices. This should work in most

Continued on next page

cases, because the Vendor Invoice Number in **BAS** is ten characters long, and most vendors do not use that many numbers. Here's how this would work: Let's say vendor number V100 sends you his invoice, numbered 12345678, for $1000.00. You have received merchandise valued at $300.00. The additional $700.00 worth has not yet been shipped and isn't expected until sometime next week or next month. You tell Accounts Payable, via Vendor Invoice Entry, that you have received Invoice Number 12345678A from Vendor V100 for $300.00. The invoice becomes an open payable which will not interfere with Invoice Number 12345678B or 12345678C if they need to be added in the next few days.

If you find that a vendor is using all ten characters of the Invoice Number field, I suggest you shorten his number for him so that you can add your supplementary character. This vendor is entirely too tied up in big numbers, and you will be doing him a favor.

Question: *We recently had a hard disk go bad on us and had to have it replaced. Thank goodness we had a current tape backup of our business data at the time! We ran into a problem with the Business Applications after we restored our software onto the new disk. It seems that only the superuser (manager or root) is able get into the programs. Everyone else gets an error 12 before getting to ENTER OPERATOR CODE. How do we fix this?*

Answer: This is a common problem with tape streamer backups. When software is restored, permissions are altered for the **BAS** files locking out all but the superuser. Since the superuser "owns" the files, the superuser must free them up.

Log onto your computer as "root" and type in the following command at the # prompt:

```
chmod 777 /b/BDATA/*
```

This opens permissions for all users to read (r), write (w), and execute (x) all files contained in the **/b/BDATA** directory. (There need be no restrictions on anything in the **/b** directory.) You may exit the UNIX command mode by holding down the grey CTRL key and typing D. It may be necessary to conduct a similar chmod command (using 777, 666, 660, etc., depending on the type of file/directory being changed) on word processing and **Multiplan** files and directories as well. Trouble with these UNIX programs is indicated by a "permission denied" message. If you do not know which permissions should be authorized for the UNIX files, "Contact your dealer" or someone else of knowledge.

Question: *I sent you a question several months ago and have yet to hear from you. Are you angry with me or something?*

Answer: Now how could I be angry with a reader of **/u/fortune *news***? I mean, after all, you have such good taste!

Unfortunately, last December during the Christmas rush a packet of questions was lost by Uncle Sam (can you imagine that?). More than likely your question was a part of the packet. Please submit it again and I will try to get to it as soon as possible.

Question: *We are an IDOL/BAS support firm who, through our own fault, missed the deadline for your 1987 list. Could you please include us in your next available issue?*

Answer: I believe that I gave ample warning this year to all who wanted to be included in the listing. It would be unfair of me at this time to report that **John K. Harris and Associates** of Houston, TX, supports IDOL/BAS at the rate of $65.00 per hour, billed in 15 minute blocks, if one should call (713) 667-1781 and ask for either Kathy Bexley or John Harris. So I guess you will just have to wait until next year. □

News Concerning Our Software Diskettes

We need to explain something about some of our Software diskettes. Several of them can be installed through Fortune's Global Menu. This technique is the most familiar to our subscribers. However, there has been some confusion because you may have expected the programs that you installed to be highlighted on the Global Menu. We purposely did not build this into the install procedure because you would quickly run out of slots on the Main Menu.

If you want to add a program to the Fortune Global Menu, you can use a program called **addglobal** which is available on the D.C. Grabbag diskette. (As of 2/20/87 we are also including **addglobal** on any diskette that we sell.) To use this program, type its name while in the UNIX shell (remember to use its entire pathname if it is still on the D.C. Grabbag disk, **/f/shells/addglobal**).

For example, let's say you purchased our UNIX Tools diskette and decided to install the rolodex program onto your hard disk. You would first use the Global Menu to install the rolodex program onto the hard disk.

At this point the program has *only* been copied onto your hard disk. To run the program you should go to the UNIX shell and type **rolo**. If you want to add **rolo** to the Global Menu, you will have to run the **addglobal** program. To do this put your copy of the D.C. Grabbag diskette into the floppy drive, mount the floppy drive, and type **/f/shells/addglobal**. This program will prompt you for certain pieces of information. When the program is finished, **rolo** will be installed on the Fortune Global Menu.

Hack on the UNIX Tools Diskette

In order for **hack** to run properly on the Fortune terminal, you need to make a minor change to the **/etc/termcap** file. Use your favorite editor and add the following characteristic to the file. Find the line that has the following: `nl=5^J:\`. Make it look like: `nl=5^J:nd=^I:\`. This should help. □

ATTENTION LAW FIRMS

You made a smart decision when you bought your Fortune computer for word processing and office automation.

Now take another giant step forward and install Gill & Piette's

TIME & FINANCIAL MANAGEMENT SYSTEM FOR LAWYERS (TFM)

on your existing Fortune system.

The product of over 25,000 hours of research, development and testing, TFM is a total law office accounting and management information system featuring:

- *autoflex billing in multiple, user-definable formats*
- *integrated client accounting and general ledger*
- *flexible management reporting for work in progress, write offs, write ups, time history, billing realization, etc.*
- *integrated modules for Accounts Payable, Docketing, Conflict checking, Trust accounting, Finance Charges*
- *multiple realization measure, including attorney billing realization and fee receipts by working attorney*

Support? We have a national software support and maintenance program that covers everywhere from Bermuda to Washington State. Nobody can support your installations as well as we will, and we've got the track record to prove it.

HOW MANY PROFESSIONAL HOURS CAN YOU AFFORD TO SPEND DESIGNING YOUR OWN SYSTEM OR TINKERING WITH SOMEBODY ELSE'S UNSUPPORTED SOFTWARE?

MAKE THE RIGHT BUSINESS DECISION WITHOUT FURTHER DELAY AND INSTALL TFM, THE FINEST LAW OFFICE MANAGEMENT SYSTEM AVAILABLE TODAY!

FOR MORE INFORMATION CALL

GILL & PIETTE, INC

(202) 546-6170

Glossary

continued from page 1

backwards under the platen of a letter quality or NLQ type printer, particularly if they have been in the printer long enough to begin to curl, can result in a label peeling off. It's no fun to get at and the residue of glue left behind must be carefully removed.

Labels and Records Processing

Now let's look at Records Processing. Here's the way I would set up a simple name and address list document record:

```
<lastname>
<firstname>
<address1>
<address2>
<city>
<state>
<zip>
<>
```

(If you are unfamiliar with list documents, consult your Records Processing manual or our previous discussion in **/u/fortune *news***.) Note that this list, when formatted in the format document, will result in a four-line address which will fit on my 3-1/2x15/16 labels. If you also remember from the previous article, I recommended that you place "xx" in the address2 field if there is only a one line address. We'll see why in a minute.

When you prepare your label format document, there is one trick that will save you a lot of grief. Disregard the otherwise excellent Records Processing manual instructions and, instead of using carriage returns to separate each name and address on the label, make each one (each record, that is) a separate page. What does this do? It allows you print either three or four line addresses with no change in the label's vertical alignment. Here's how it works.

Set up your format document as shown in Figure 1.

You should also set the format line in the format document to the length of the label, according to what type pitch you plan to use.

Now run Records Processing, making an output document only. Edit your document and use global search and replace to find all the "xx<RETURN>" in your document, that is all the double x's followed by the return symbol. Replace them with nothing (just hit execute when asked what to replace them with and execute again to start the search and replace). When the process is finished, you will have either three or four line addresses, with no blank lines, resulting in good looking labels. You can also check at this time to make sure none of your address lines is too long for the label, since this will throw the vertical alignment off. You might want to abbreviate to keep the information on one line.

Incidentally, while you are editing your output document, you can add information identifying the list and the date of the run. Just insert the information at the beginning of the list and remember to add a page break. The first label printed will contain this information.

The next step is to send your new document of formatted labels to the printer. On the print menu, be sure to set the pitch properly and then set the page length to 6 (six lines per page). Each label is 15/16" deep and there is another 1/16" between labels. Your printer is normally set for six lines to the inch. Thus each new label will start printing at precisely the same position, no matter if the preceding one had a three or four (or even two) line address.

It will probably take some experimenting to determine just how to line up the labels in your printer so that the information will be centered properly on each label. Send only the first three or four pages to the printer and then check the alignment. This way you won't waste a lot of labels. Once you get the alignment right, you can be sure that the rest of the labels will print correctly.

One-up labels print the fastest, are the cheapest, and are the easiest to handle for hand-applying. But what if you already have some in another format, say two-up? Records Processing can accommodate these as well.

The first thing to do is to determine the horizontal spacing. Mark the spot on the left label where you want to begin printing and then measure the distance to the same position on the second label across. This is where you will place a tab in the format line of your format document. The right margin is set to the end of the printed area of the second label. The format document example we used above is now changed to look like Figure 2.

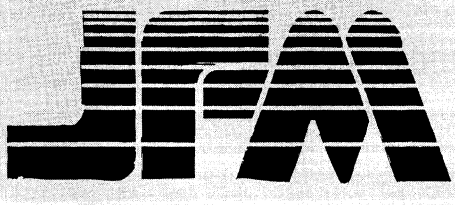
If you have a three or four up format, the procedure is the same. Just add tabs and repeat the field labels, using ",3" and ",4" where appropriate. Note that removing empty address lines is more difficult when using a 2- or more up format. You will have to do some manual editing to keep all the fields in the proper record, and if the list is very long you may have to be content with simply removing the "xx" and leaving a blank line. It won't look as nice, but it will work. □

Figure 1.

```
<firstname> <lastname>
<address1>
<address2>
<city>, <state> <zip>
- - - - - - - - - - - - - - - (page break)
```

Figure 2.

```
<firstname,1> <lastname,1>     <firstname,2> <lastname,2>
<address1,1>                   <address1,2>
<address2,1>                   <address2,2>
<city,1>, <state,1> <zip,1>    <city,2>, <state,2> <zip,2>
- - - - - - - - - - - - - - - - - - - - - - - - (page break)
```



Dear Fortune Owner:

As you have probably read in */u/fortune news,* Fortune Systems SX models are significantly more powerful and faster than the PS and XP models. If you have a PS or XP model system, you should take advantage of our special promotion on SX upgrades. For a limited time, and while our supplies last, we will upgrade your system, including the new SX Cabinet at no additional charge for the cabinet. With this special offer, you may choose any combination of the following upgrades and/or expansions:

- *SX Processor with Current Release of Multi-User For:Pro*
- *60 MB Internal Streaming Tape*
- *30 MB, 45 MB, 70MB High Speed Hard Disk*
- *6-Port Intelligent Asynchronous Controller*
- *Upgrade to Current Release of all Fortune Software*
- *Fortune:Windows (Included at NO Charge)*

Additionally, we can perform these upgrades within 24 hours of receiving your system. In fact, if you ship your system overnight on a Friday for Saturday delivery, we will have it back to you on Monday morning, all ready to plug in and run; not losing the use of your system for a single work day. And, we will issue you a full 90-day warranty on the entire system.

JFM provides services to Fortune Systems owners either directly or through our authorized Affiliated Dealers. If you haven't used our services, I invite you to discover JFM; you will be glad you did.

Sincerely,

Morris Frydman
President

Talk to us, then talk to our clients



JFM BUSINESS SYSTEMS
The Merchandise Mart
Lobby Suite 144
Chicago, Illinois 60654
Call (312) 644-0870

Authorized Fortune Systems Master Dealer

SUPPLEMENT TO ALL VERSIONS OF BAS AT A REASONABLE PRICE, TOO!

JOB COSTING

The Right Tool for All Trades

BaSiC SOFTWARE CORPORATION

For More Information Call or Write...
BaSiC Software Corp.
5201 Powhatan Street
Baltimore, MD 21207
301-448-9460

Available on:
IBM AT & compatibles
UNISYS/SPERRY
Altos, All Levels

DO YOU KNOW...

there is an organization that has a thorough knowledge of

FORTUNE BAS
MULTIPLAN IDOL

We have been providing widespread support to our Fortune clients for more than 4 years.

If you need:

- training
- support
- maximum computer performance
- peace of mind relating to processing
- customization
- application results
- minimum errors and confusion
- better ideas on how to increase the power of your Fortune

Call Steve Rosenfeld at (314) 837-7111

SUPERIOR COMPUTER SYSTEMS, INC.
Making computers work for you

917 N Hwy 67, Suite 105
P.O. Drawer T
Florissant, MO 63031

/u/fortune news
73 Jason Street
Arlington, MA 02174

/u/fortune* news

The Newsletter for Users of the Fortune 32:16 Computer — April 1987/Volume 4 Number 4

Buyout Offer Under Consideration

News About Fortune Systems SCI Systems Makes Offer

There has been some interesting news about Fortune Systems in the last several weeks that you may have heard about in financial circles. In the past, we have reported that an offer to buy the hardware division of Fortune had been made by a management group headed by **Robert A. Davis**, Vice President for Marketing, and **Brooke P. Taylor**, the President of Fortune Systems International. The Board of Directors was said to be weighing this proposal while also exploring alternative proposals. Apparently, at least one of the other proposals was given serious consideration.

On April 3 it was announced that Fortune Systems Corporation and **SCI Systems, Inc.**, of Huntsville, Alabama (NASDAQ-SCIS) had jointly signed a letter of intent pursuant to which SCI would acquire certain assets from Fortune Systems, including their research and development, manufacturing, sales, marketing, and support operations. At the time, their press release stated that the acquisition was pursuant to certain regulatory approvals, the execution of a definitive agreement, and the approval of Fortune's shareholders.

SCI Systems, Inc., is a "Fortune 500," vertically integrated electronics manufacturer whose products and services are supplied to a variety of military, aerospace, and commercial customers. They currently manufacture their own line of multi-user computers based on Intel microprocessors and UNIX System V. They have expertise in surface mount technology and other state-of-the-art manufacturing techniques. Included among the products they produce are several boards for the new IBM series of microcomputers.

During the month of April, Fortune and SCI began working out an agreement for the acquisition. Following an extension of the due diligence period, **Jim Campbell**, the CEO of Fortune, and **Robert Davis** sent out the following memorandum to their dealer network:

SCI's proposed buyout is seen as positive.

Dear Reseller/Var/Distributor:

You undoubtedly continue to be interested in our April 3, 1987 announced letter of intent for the sale of Fortune's computer business assets to SCI Systems, Inc. Many of you have called us for information. This letter is both a progress report on those negotiations and a statement on why we believe the completion of this transaction could be good for your business. In our Robert Davis/Pete Taylor letter dated April 7, 1987 to you, a reference was made that completion of the SCI due

See **Glossary,** page 22

The BASIC Advisor

Ray Wannall is President of BaSiC Software Corporation in Balimore, MD, and is contributing to this publication independently.

My open letter to Choice Systems (**/u/fortune *news***, Volume 4, Number 2) generated many responses, mostly from dealers who were very supportive of my efforts to uncover the 2.0 BAS Application mysteries. My supporters will probably be as surprised as I was to learn the true story behind the development of the 2.0 Payroll and the role played by Choice Systems. Here is a recap of how the story unfolded for me this past month.

See **BASIC Advisor**, page 21

In This Issue...

The Basic Advisor: Ray hears from readers re: Choice...**Page 1**

And Choice Responds - See Letters...**Page 20**

On-line Databases for Physicians...Page 4

The UNIX Directory: The "Finale" Shell Script...**Page 8**

UNIX Basics - Redirecting output...**Page 12**

/u/help - Multiplan printing, Fortune:Word 3.0, etc..**Page 17**

How to Become a Famous Screenwriter...Page 18

*/u/fortune *news* is published by The Cambridge Consortium, Inc. It is not affiliated with Fortune Systems Corporation.

/u/fortune *news...*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz, Ph. D.

Technical Editor
Scott Bradner

Advertising Manager
Mark Palmerino

/u/fortune *news...* is published 12 times per year by The Cambridge Consortium, Inc. Neither /u/fortune news... nor The Cambridge Consortium are affiliated with Fortune Systems Corporation.

Subscriptions cost $48 for one year within the U.S.A. For foreign subscriptions, add $27 each year for added postage and handling costs.

Back issues are offered at:

Vol 1, Numbers 1 - 3	$1/each
Vol 1, Number 4	$.50/each
Vol 1, Numbers 5 - 8	$3/each
Vol 2, Numbers 1 - 7	$3/each
Vol 3, Numbers 1 - 12	$4/each
Vol 4, Numbers 1 -	$4/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $10 each. Please add $5 per order for foreign destinations.

Although every attempt is made to insure that our contents are error-free and without risk, The Cambridge Consortium is not responsible for losses that occur due to correct or incorrect interpretation of **/u/fortune** *news*.

For more information, call or write:

/u/fortune *news*
73 Jason Street
Arlington, MA 02174
(617) 663-7700, (617) 646-5603

The text for this newsletter was prepared using a Fortune 32:16 for text input. The information was then transferred to an Apple Macintosh Plus with kermit and formatted for typesetting using the Aldus Pagemaker program.

UNIX™ is a trademark of Bell Laboratories, Inc. Fortune 32:16™, Fortune:Word™ and FOR:PRO™ are trademarks of Fortune Systems Corporation. Multiplan™ and MS-DOS™ are trademarks of Microsoft. IBM-PC™ is a trademark of IBM. Macintosh Plus™ is a trademark of Apple Computer. Pagemaker™ is a trademark of Aldus Corporation.

©1987 by the Cambridge Consortium, Inc. 4/87

Fortune News

continued from page 1

diligence activities was contemplated by April 15, 1987. Perhaps we were a bit optimistic since a great deal of work needs to be accomplished by both businessmen and lawyers on both sides. The Letter of Intent has not been terminated and SCI and Fortune are continuing the preparation of definitive purchase documents.

We would like to put forth our views on how your business may benefit under SCI ownership of Fortune's computer business if the transaction is consummated:

-Size and Financial Strength of SCI a Strong Plus: As a profitable $500M revenue company with cash assets exceeding $150M, SCI should eliminate your end-user buyers concern over their vendors long term survivability.

-Worldwide Manufacturing and Purchasing: SCI's manufacturing strengths should provide you with state of the art products at competitive pricing.

-Technology: From its founding in 1961, SCI has been an engineering oriented firm. As such, it served NASA and other high technology defense firms. These business relationships help hone SCI's engineering and technical skills which translates to innovation and improved products.

-Expanded Product Line: SCI produces INTEL 286 and 386 based micro and mini computer products primarily for OEMs. This equipment (operating in UNIX System V) as well as other computer devices which they currently manufacture, should complement the existing Fortune product line.

-Distribution Channels: SCI has great interest in diversifying their revenue base into more commercial areas. Fortune's domestic and international marketing channel and you, specifically, give them this opportunity.

We will attempt to keep you abreast of developments as our negotiations continue. I know you understand that Fortune and SCI, as public companies, are prevented from releasing information of a material nature on a selective basis. Any further press releases issued will be immediately sent to you.

In trade press articles, you may have received an impression that we were "dropping out" of the business. Nothing could be further from our intention. SCI's goal is to build a larger business on the base of 65,000 installed Fortune users.

In the meantime it's "business as usual" at Fortune and we ask that you continue your business relationships with us just as you have in the past. Our sales and service departments are prepared to serve your needs and we look forward to your continued business.

Sincerely, Jim Campbell/Robert Davis

As of the end of April, a public announcement of the final agreement was not made. However, it seems very likely that the agreement will come to pass. The approval by Fortune Systems' Board of Directors appears imminent.

What does this mean to end users?

The letter from Campbell and Davis summarizes Fortune's position regarding the sale of their hardware division to SCI. Fortune hopes to gain credibility, improved manufacturing, and an improved product line. SCI is interested in Fortune's distribution channel, including you, the installed base of Fortune users. **James Daniel**, SCI's chief financial officer explained in a recent interview in Computer Reseller News, that "We do have some VARs now, but nowhere near the number Fortune has. We hope that eventually we can integrate SCI's existing VARSs and products with Fortune VARs." As we understand it, SCI is planning to support the current Fortune hardware product line, and maintain the Fortune name on hardware. On the Fortune side, it is hoped that SCI's expertise will allow Fortune hardware to be built

Continued on next page

more efficiently and on a larger scale, allowing Fortune a further penetration in the marketplace.

If the deal comes to pass, the unsold portion of Fortune Systems Corporation will become a holding company whose initial subsidiary is **Tigera**, the company formed to expand and port Fortune software onto other manufacturers' hardware. They hope to buy other profitable companies with the $20 million in cash left over from the original stock offering, along with $60 million in tax-loss carryforward credits that have accumulated from a number of years of losses.

The direct implications for current Fortune users remains to be seen, although the mood amongst Fortune dealers and at Fortune Systems is upbeat. They feel that the acquisition by SCI is a positive move that will strengthen the position of the current Fortune products. The expectation is that the current management and marketing personnel will remain to lead the new company.

As new information becomes available about this purchase, we will keep you informed. **/u/fortune *news*** intends to continue publishing our newsletter for an even larger customer base.

* * * * *

For:Pro and Fortune:Word Upgrades

We have just received our copy of **For:Pro 2.0** and **Fortune:Word 3.0**, and we expect to review them in our next issue. In the meantime, we thought it would be helpful to pass on some installation information about these products.

The notes are grouped by heading, please be sure to read any notes that apply to products you currently have installed on your system:

BAS Applications

The **1.4** version of **BASIC** will not run under **For:Pro 2.0**. If you are planning to upgrade to the new operating system, you will have to purchase the **6.5.12** release of **BASIC**. Once you install the **BASIC** upgrade, all of your applications should run normally. If you have done any customizing of the shell scripts that begin **BASIC** (**DBASIC**, **TBASIC**), you may have to modify them because of the way that **For:Pro 2.0** deals with devices.

When **6.5.12** is installed, it should be installed over **BAS 1.4**, e.g. -- **DO NOT DELETE 1.4 BEFORE INSTALLING 6.5.12**. The upgrade offer for **BAS 6.5.12** is available for a limited time only. If you plan to upgrade to **For:Pro 2.0** at any time, you should contact your dealer about the **BAS** upgrade immediately.

Fortune:Link

The **1.0** version of **Fortune:Link** will ***not*** work with **For:Pro 2.0**. There is a new revision called **1.1** of **Fortune:Link** which will be released shortly. Until it is out, you should ***not*** use **For:Pro 2.0**

Laser Interface

Certain files that were only included with the Laser Interface in the past are now included as standard files with **For:Pro 2.0**, such as **lpd** and a **printcap** which includes the **HP** entry. Because of this change, upgrading **Fortune:Word** or **For:Pro** requires changes in the Laser Interface.

The old version of the Laser Interface will not work with **For:Pro 2.0** or **Fortune:Word 3.0**. If you are planning to upgrade to either of these, you will need to have your Laser Interface disk upgraded to level **2.2**. This can be done by sending your old master disk to the Custom Products Division of Fortune Systems, 300 Harbor Boulevard, Belmont, CA 94002. Include a letter which describes which upgrades you are planning, e.g. **Fortune:Word 3.0** or **For:Pro 2.0**. Also include a check for $50 payable to Fortune Systems for the upgrade cost. (We suggest that you make a backup of your old Laser Interface before sending off the original disk.)

Fortune:Word 3.0

Fortune:Word 3.0 is available as an upgrade to **Fortune:Word 2.0**, or as a complete product for earlier versions of **Fortune:Word**. Fortune says that the upgrade option will only be offered *until* June 30, 1987. The upgrade costs significantly less than the full product.

Included with the new version of **Fortune:Word** is a complete new set of documentation which includes the following manuals: *How to Use Fortune:Word*, a self-paced learning guide; *The Fortune:Word Reference Guide*, a learning and reference tool for all **Fortune:Word** features; *The Fortune:Word Glossary User's Guide*, a manual describing how to create and use special glossary documents in which you can save text and keywords for later recall; and *The Fortune:Word Records Processing User's Guide*, a manual describing how to use **Fortune:Word's** Records Processing feature to quickly and economically produce customized form letters, labels, or lists.

Using Standard Console as the Console

If you are using the standard console (the terminal the plugs directly into the front and back of the CPU), you can continue to use that terminal as the console when converting to **For:Pro 2.0**. The documentation is a little vague about this, but it will work. If you are using a terminal as the console, it must be connected to the **SIO** port on the back of the CPU (the one all by itself in the middle-back of the machine -- also known as **tty01**).

The significance of the console is that it is the terminal that is used to turn on and shut down the computer, and it is also the terminal which will receive system error messages. Therefore if you are using a terminal as the console, you should be sure to leave it on while the system is on, or you may never receive information about system problems.□

Josh Lobel

MEDLARS: Data Bases for Physicians

This month we continue our focus on Physicians and Computers by expanding on the information provided in the last article regarding data bases made available by the National Library of Medicine. In addition, we discuss briefly a low-priced, menu-driven communications package called Grateful Med. Offered by the National Library of Medicine (NLM), it efficiently guides you through your search of two of their data bases -- MEDLINE and CATLINE.

The overall data base system of the NLM is called MEDLARS. As we noted last time, you can contact the NLM for information and application forms at one of the following telephone numbers: 800-638-8480 or 301-469-6193. You cannot access the system without an account and a MEDLARS User ID code with a password. These will be provided to you after your application has been processed. Once processed, a number of services are available to you, many of them free. You receive $40.00 of practice credit on the MEDLARS system and monthly issues of the NLM Technical Bulletin which publishes news about the system, data bases and price changes. In addition, a variety of training services are provided by the NLM.

The Initial Online Services Training Class, which is free of charge, is a four and a half day course consisting of three modules. Module I lasts three days and includes information on: an overview of the system, access procedures, basic commands, use of the controlled Medical Subject Headings (MeSH), an explanation of the MEDLINE unit record, and formulation of search strategies in the MEDLINE data base. Module II lasts one day and includes in-depth discussions of the content, use, and searching methods associated with chemical and toxicological data bases available on the MEDLARS system. Module III is a half-day session covering the scope, content, and mechanics necessary to search the cancer literature found in the MEDLARS system. These include the data bases CANCERLIT, CANCERPOJ, CLINPROT, and PDQ. (All data bases maintained on the MEDLARS system are named and described below). You must take Module I before you can take Module II or III. These classes are offered several times a year across the United States. For any one MEDLARS account, only one person can be trained at a perticular administration of the course. However, other individuals using the same account and ID code can be trained at subsequent administrations of the course.

The Advanced Online Services Training Class is a five-day, five-module course which details the specialized data bases on the MEDLARS system and it provides trainees with additional opportunities to improve their searching skills. The aim of this class is to present the system techniques required to formulate complicated or multi-faceted searches of the various NLM data bases. Each module is devoted to one of the available data bases. You have to take the Initial Training Class in order to take the Advanced Training Class. Again, this course is provided free of charge. Other training services do have some fee associated with them. These include a six-to eight hour course, "The Basics of Searching MEDLINE", that has been designed for health professionals who wish to learn only the rudimentary search techniques for MEDLINE. There is a self-instructional workbook that was designed to supplement this course. This workbook, entitled The Basics of Searching MEDLINE: A guide for the Health Professional, can be purchased separately for $12.00 plus $3.00 for shipping and handling from the National Technical Information Service. (The order number is PB 85-232650/GBB).

We now list alphabetically all of the data bases available on the MEDLARS system along with a brief description of each data base. Unless otherwise noted most of the citations in bibliographic data bases refer to English-language items.

AVLINE (Audio Visuals onLINE): Updated weekly, it contains bibliographic citations to all audiovisual material in clinical medicine catalogued by the NLM since 1975. These materials include motion pictures, videocasettes, slide/casette, and filmstrip/casette programs for subjects ranging from Anatomy to Zoonoses.

BIOETHICSLINE or BIOETHICS: Updated monthly, it contains bibliographic citations covering ethics and public policy in the fields of health care and biomedical science since 1973. Topics range from in vitro fertilization to euthanasia. Source material includes journal articles, monographs, newspaper articles, court decisions, bills, audiovisual materials, and unpublished documents.

CANCERLIT: Updated monthly, it contains bibliographic citationscovering major cancer topics. Source material includes journal articles, government reports, meeting abstracts, monographs,

these, etc. (Approximately 200 foriegn journals are also screened). Many of these citations include abstracts.

CANCERPROJ: Updated monthly it contains summaries of all ongoing research projects and those terminated less than a year previously. Each project is updated annually. These summaries include project descriptions, objectives, experimental design, and preliminary results. Also given are the addresses and telephone numbers of principal investigators.

CATLINE (CATalog onLINE): Updated weekly, it contains more than 600,000 bibliographic records covering the biomedical sciences for virtually all of the catalogued titles in the NLM's collection ranging from fifteenth-century imprints to the present. Source material is primarily monographic including records for serials, monographic series, and audiovisual series. Slightly more than half of the citations refer to non-English items.

CHEMLINE (CHEMical dictionary onLINE): Updated bimonthly, this data base is an online, interactive dictionary of chemicals. It provides a mechanism whereby information on over 650,000 chemical substances can searched and retrieved online.

CLINPROT (CLINical PROTocols): Updated monthly, it contains summaries of clinical investigations of new anti-cancer agents and treatment modalities, including current as well as some closed protocols dating back to 1960. Sponsored by the National Cancer Institute, it provides desciptions of the clinical trials, patient entry criteria, the therapy regimen, and special study parameters.

DIRLINE (DIRectory of Information Resources onLINE): Updated quarterly, it contains a directory of organizations that provide information in specific subject areas. Sources include the Library of Congress' National Referral Center (which lists over 14,000 public and private resource centers) and the National Health Information Clearinghouse (which contains over 1,500 records for organizationsthat provide information for the general public in various areas of health and disease).

HEALTH (HEALTH planning and administration) : Updated monthly, it contains bibliographic references (mostly journal articles) from 1975 to the present covering non-clinical aspects of health care delivery. These include planning and administration of health facilities, services and manpower, health insurance, health policy, financial management, regulation, quality assurance, etc.

HISTLINE (HISTory of medicine onLINE): Bibliographic citations on the history of medicine and related sciences come from CATLINE, MEDLINE, journals in the history of medicine and other sciences, and recent publications in fields such as general history and classics. Source material also includes symposia and congresses.

We are the *largest* Authorized Fortune Master Dealer in the Pennsylvania and Delaware area. As part of the **P**rofessional **S**olutions **G**roup of companies with over 35 years in business automation, we have the *stability* that our customers demand

Our services include a factory trained field engineering team, a complete parts inventory including system upgrade capability and a software support department.

For all of your Fortune needs contact:

Mark Shields of PSG at
2550 Eisenhower Avenue
PO Box 762
Valley Forge, PA 19482

or you can call Mark at (215) 666-1400

MEDLINE: Updated monthly, with more than 20,000 items added at each update, it contains bliographic citations of journal articles in biomedicine from 1996 to the present. 75% of the citations refer to English-language items and many of the citations include an abstract of the reported research.

MESH VOCABULARY FILE: Updated annually, this file contains a thesaraus of biomedical-related terms and an online dictionary of Medical Subject Headings (MeSH), Subheadings, and Chemical Terms that are used in indexing and searching several of the online data bases including MEDLINE, CATLINE, AVLINE, POPLINE, BIOETHICS, CANCERLIT, CANCERPROJ, and HEALTH.

NAME AUTHORITY FILE, NAF This contains the authority records for over 185,000 names, series, uniform titles used as headings in bibliographic records resident in CATLINE and AVLINE. These records may be used in the cataloging process for personal, conference, or corporate names, uniform titles or series to be used as a heading in a bibliographic record.

POPLINE (POPulation information onLINE): Updated monthly, it contains bibliographic citations covering population and family planning primarily from 1973 to the present, but with selected citations dating back to 1886. Source material includes journal articles, monographs, technical reports, and unpublished works. It refers to range of areas including human fertility contraceptive methods, community-based services, program evaluation, demography, census, vital statistics, and related health, law, and policy issues.

PDQ (Physician Data Query): Updated monthly, it contains a databank of information on advances in cancer treatment and clinical trials. This includes detailed summaries of all major tumor types, contact information for 10,000 physicians and 2,000 organizations that devote a major portion of their resources to cancer treatment, and detailed information on more than 1,000 active treatment protocols that have been approved by the National Cancer Institute.

RTECS (Registry of Toxic Effects of Chemical Substances): Updated quarterly, it contains a data bank on more than 70,000 substances with toxic activity. This is the NLM's online version of the National Institute of Occupational Safety and Health's annual compilation of substances with toxic activity, a single source for basic toxicity information.

SERLINE (SERials onLINE): Updated monthly, it contains bibliographic records of all biomedical and other serials and numbered congresses which are currently in the NLM collection (and a limited number of serials that are not in the collection).

TOXLINE (TOXicology information onLINE): Updated monthly, it contains bibliographic citations from 1965 to the present covering the toxicological, pharmacological, biochemical, and physiological effects of drugs and other chemicals. Source materials include journal articles, meeting papers, monographs, technical reports, etc. Approximately two thirds of the citations refer to English-language items.

TOXNET (TOXicology data NETwork): This is a computerized system of toxicologicallyoriented data banks with a variety of modules used to build and review records. For outside users, it offers a sophisticated search and retrieval package which permits efficient access to valuable data drawn from numerous sources on toxic and otherwise hazardous chemicals. It includes acces to two other files: CCRIS and HSDB.

CCRIS (Chemical Carcinogenesis Research Information System): Maintained by the National Cancer Institute, this contains carcinogenicity, tumor promotion, and mutagenicity tests results drawn from a set of core sources plus major journals and special reports.

HSDB (Hazardous Substances Data Bank): Drawn from a variety of sources including goverment documents and special reports, it contains toxicological information enhanced with additional data related to the environment, emergency situations, and regulatory issues.

As we said in the last article, charges for the use of these data bases are determined by a complex algorithm that includes such factors as use during prime (10 A.M. to 5 P.M. EST) versus non-prime time, the particular data base accessed, connect time, cpu time, etc.

As we noted earlier in this article, The National Library of Medicine also offers **Grateful Med**, a communications software package that provides easy and efficient access to parts of the MEDLARS system. It also reduces your cost since it minimizes the time that you spend online searching through the system. It is a menu-driven system that helps you formulate your search procedures through MEDLINE and CATLINE. It then automatically calls the NLM computer, runs the search, and disconnects from the NLM computer after transferring the search results to your computer for offline review. The good news is that this software is available from the NLM for only $29.95. The bad news is that it seems to be currently available for IBM-PC and IBM-compatible machines. Thus, this software would be appropriate for individuals who own Fortune DOS workstations. □

Benzion Chanowitz

*** See the NEW product DOS LINK ***

SOFTWARE FOR THE FORTUNE 32:16

NEW PRODUCT!!! DOS LINK ($495.00) - provides easy-to-use menus and programs that transfer data between any PC (including Laptops) running DOS and the Fortune system. Here are just a few examples of how this product can be used:

- Transfer Multiplan or Fortune:Word Documents from or to your DOS system.
- Transfer other ASCII or binary files.
- Use the PC or Laptop as a remote terminal on the Fortune system.
- Transfer THOROUGHBRED (SMC) BASIC data that has been extracted using one of our other products

Documentation and software for both the DOS and Fortune system are provided. Data may be transferred via a direct connection (9600 baud) between the two computers or via modem (at modem speed). You may also order additional hardware for your PC that may be required to accomodate the transfer.

If you are interested in purchasing a Laptop computer that can be used with this product for people on the road or at home, give us a call!! We also provide consulting services for interfacing your DOS system to the Fortune ($50/hr).

MULTIPLAN LINK - can pass data from BAS/IDOL/BASIC files to Multiplan and back. For example, it could be used to send data from any of the BAS files (Chart of Accounts, Customer Master, Inventory Master, etc). Allows selection of fields to be sent and provides logical retrieval and key range selection. **$295.00**

RECORDS PROCESSING LINK - can pass data from BAS/IDOL/BASIC files to Records Processing and back. For example, it could be used to take customer address data from the Customer Master and will automatically make the Records Processing List document for mail/merge. Allows selection of fields to be sent and provides logical retrieval and key range selection. $295.00

ASCII LINK - can pass data from BAS/IDOL/BASIC files to ASCII and back. Use to send data between machines or databases. For example, it could be used to send data from any of the BAS files (Customer Master, Inventory Master, etc). Allows selection of fields to be send and provides logical retrieval and key range selection. **$295.00**

TERMINAL/PRINTER LINK - provides capability to print data on a printer connected to a workstation. Can print BAS/IDOL/BASIC reports, Multiplan and Fortune:Word documents. BAS/IDOL reports can also be viewed before printing, converted to Fortune:Word format; accumulated for mass printing; backed up to diskette; printed on any system printer. An option to print BAS/IDOL reports on a LASER PRINTER is $50 extra. Many other options are included. **$195.00** (Fortune:Word printing on a printer connected to a workstation is $50 extra)

TELEPHONE LINK - on line telephone message, telephone directory and inter-office mailbox/menu system. Telephone message entry resembles typical pink slip. Telephone directory can search by name or company. Mailbox/message system can send and receive Fortune:Word documents or Multiplan documents. Does much more! Give us a call. Written in Business Basic. **$295.00**

CALCULATION LINK - provides programs for amortization, depreciation, loan repayment, averaging (with graph), linear correlation and breakeven analysis. Written in Business Basic. **$95.00**

KOMPACT PERSONNEL ACCOUNTING - provides for data capture of personal, wage/salary, job, education, salary/wage, appraisal training, dental, medical and life insurance data. Includes over 80 pre-defined reports/worksheets. Written in Business Basic/IDOL. **$995.00**

PRINTER LINK - provides printcaps for Okidata 84, 2410, TI 855, Anadex 6500 and Printronix. Includes shell script for changing pitch and LPI. Allows these printers to be used in Fortune:Word. **$95.00**

MODEM LINK - allows Hayes 1200/2400 modem to be used on one port for both incoming and outgoing calls. **$95.00**

WESTERN LINK - provides instruction on how to use your Fortune for sending/receiving telex messages using Western Union's Easylink. Can replace your telex! Fortune ITE or Handshake software required. **$95.00**

UNI-KOMP 9803 Rocktree Houston, Texas 77040 (713) 937-7782

The UNIX Directory "The Finale" Part 9

by David E. Kloes

Here we are again for the last time in our "INTRODUCTION TO SHELL SCRIPTS" series. As promised, we will summarize most of the commands and concepts we have talked about over the last nine months by presenting a shell script which we will go through line by line. We want to explain in advance that some of what we are about to present is not represented as the most efficient way of writing a shell script - but rather, we have chosen certain sequences in order to demonstrate most of the material we have discussed. To this end, we also apologize for those portions that would never be included in an actual shell script. We will actually present three shell scripts called "trap.sh", "return.sh", and "main.sh". Please see figures 1 and 2 for a listing of these shell scripts. To keep things simple, we have located all three programs in the **/tmp** directory; therefore, you will see the "/tmp" prefix included in the pathnames.

The program that starts up our "main.sh" shell script is "trap.sh". The way we run this shell script is:

$ trap.sh y

As you can see from the command line we have entered, we are passing a variable into "trap.sh" and it is simply the letter "y" (which stands for "yes"). In this particular set of shell scripts, the reason we have done this is because we are going to take some actions in "main.sh" based on this parameter. We will see how it is used when we get to that point in explaining "main.sh". Remember that this parameter is passed to "trap.sh" as **$1**. In any case, the first line of "trap.sh" sets up the trap of the interrupt (**CANCEL/DEL**) key (signal 2) and instructs the shell to execute "/tmp/main.sh" whenever this key is depressed.

The second line of "trap.sh" sets up a variable called "chg" and sets it equal to the value of **$1** (which in this case is "y"). It also exports it. Remember we stated that by "exporting" a variable, we make the value of it available to subsequent shell scripts. The last line of "trap.sh" executes our "main.sh" shell script.

Each of the lines in "main.sh" is numbered. We will reference these line numbers as we go through the shell script. Note that these line numbers are not included in the shell script itself. Remember that we mentioned that appropriate sequences in a shell script should be aligned in order to make them more readable. We try to follow this convention but must occasionally drift away from our normal alignment.

Line 1 of "main.sh" is a comment line. Comment lines in the Bourne Shell must be preceded by the **#**. Line 2 sets a variable called "ext" to an initial value of "n". The "while" test in line 3 says that if the value of "chg" is "n" (which it is at least the first time this line is encountered) then "do" the list of things between "do" (line 4) and "done". Notice that the related "done" statement is line 112 so the "do" list basically consists of the entire shell script.

The first thing to "do" is line 4 which clears the screen (**^L**) and puts the name of the menu at the top of the screen. The name will be displayed in reverse video since we have included the control sequences for start reverse video (^\HD) and end reverse video (^\ID) on either side of "Sample Menu". The quote on line 5 will leave a blank line between this line and the next line that will be echoed.

The "if" test in line 6 checks to see if the "chg" variable is set to "y". Remember we set up this variable in "trap.sh" and exported it so its value can be determined here. If this variable is set to "y" (which it is), then we change directory to the user's home directory. We will see further on that this variable is also used for another reason. **HOME** is one of the "predefined" variables we talked about ages ago. Its value always tells us the pathname of the user's home directory.

In line 9, we display the user's current working directory. The **`pwd`** says to display the result of the **pwd** command. The result of any UNIX command may be assigned to a variable or used in our shell scripts by enclosing the command in back-quotes (`) as we did with **pwd**. The control characters insure that the current working directory is highlighted. The menu is then displayed (lines 11 through 17) along with the "Enter selection" prompt at line 19. The ^H is the control character for the backspace and it insures that the cursor is placed over the "_". The "read" statement at line 20 takes whatever the user enters and puts it into the variable "sel". The value of any particular variable can be found by referencing the variable name with a **$** in front of it. For example, the value of the variable "sel" can be referenced by using **$sel**.

We use the "case" statement in line 22 to determine what action to take based on the menu item selected. Each "case" has a list of tasks to be accomplished and the ;; indicates the end of each case. In this shell script, we have 7 menu selections and 7 cases addressed in lines 22, 32, 35, 46, 58, 71, and 107, respectively. For example, if the user enters "6", the list of things associated with 6) in line 71 are executed. If the user enters anything else other than the number 1-7, the ***)** case in line 109 is executed. Let's go through the list of actions associated with each of the menu selections.

If the user selects "1" (CHANGE current working library), lines 22-31 will be executed. The screen is cleared and the user is asked to enter the library they wish to change to. Whatever they enter is put into the "lib" variable. The "if" test in line 25 checks to insure that they entered a valid directory. If they did, then the "cd" command is executed and the "chg" variable is changed to "n". Since the user elected to change to a directory other than their home directory, they will not be put back to their home directory when lines 6-8 are re-executed. If they

Continued on next page

Figure 1

```
Program 1:  trap.sh                Program 2:
return.sh

trap '/tmp/main.sh' 2              echo -n "^G
chg=$1; export chg                 Depress
<RETURN> to continue: "
/tmp/main.sh                       read bunk
```

Figure 2

Program 3: main.sh

```
1   # Shell script to summarize commands and concepts
2   ext=n
3   while test "$ext" = "n"
4   do echo "^L            ^\HDSample Menu^\ID
5   "
6   if test "$chg" = "y"
7     then cd $HOME
8   fi
9   echo -n "Your current working library is: ^\H `pwd`^\I
10
11   1.  CHANGE current working library.
12   2.  LIST contents of current working library.
13   3.  DISPLAY data in file.
14   4.  COPY data file.
15   5.  RENAME data file.
16   6.  REMOVE data file or library.
17   7.  EXIT menu.
18
19   ENTER YOUR SELECTION: _^H"
20  read sel
21  case $sel in
22    1) echo -n "^L
23  Enter library you wish to CHANGE to: "
24       read lib
25       if test -d $lib
26                then cd $lib
27                     chg="n"
28            else echo "
29  Library $lib does not exist."
30                /tmp/return.sh
31            fi;;
32    2) echo ^L
33             ls|more -d
34             /tmp/return.sh;;
35       3) echo -n "^L
36  Enter name of file you wish to DISPLAY: "
37             read display
38             if test -r $display
39                then echo ^L
40                     more -d $display
41                     /tmp/return.sh
42             else echo "
43  No permission or $display does not exist."
44                   /tmp/return.sh
45             fi;;
46    4) echo -n "^L
47  Enter name of file to be COPIED: "
48           read file1
49           if test -r $file1 -a -w $file1
50             then echo -n "
51  Enter name/pathname of NEW file: "
52                  read file2
53                  cp $file1 $file2
54           else echo -n "
55  No permission or '$file1' does not exist."
56              /tmp/return.sh
57           fi;;
58    5) name1=""
59           echo -n "^L
60  Enter name of file to be RENAMED: "
61           read name1
62           if test -w $name1 -a "$name1" != ""
63             then echo -n "
64  Enter new name: "
65                  read name2
66                  mv $name1 $name2
67          else echo -n "
68  No permission or '$name1' does not exist."
69              /tmp/return.sh
70          fi;;
71       6) echo -n "^L
72  Enter name of file(s) or library(ies) to be removed: "
73           read file
74           x=0
75           for name in $file
76           do
77              if test -r $name
78                 then stop=n
79                      while test "$stop" = "n"
80                      do echo -n "Remove $name? (y/n): "
81                         read del
82                         case $del in
83                            Y|y|YES|yes) rm -r $name
84                                         x=`expr $x + 1`
85                                         stop=y;;
86                              N|n|NO|no) stop=y;;
87                                     *) echo "
88  You must answer 'y' or 'n'."
89                                         stop=n;;
90                         esac
91                      done
92              else echo -n "
93  No permission or '$name' does not exist."
94                   /tmp/return.sh
95              fi
96           done
97           if test $x -lt 10
98              then echo "
99  Less than 10 files were removed"
100          elif test $x -gt 9
101             then echo "
102 10 or more files were removed"
103          fi
104          echo "
105 $x files were removed."
106          /tmp/return.sh;;
107       7) ext=y
108          exit;;
109       *) echo -n "^G^\H Not a valid response^\I "
110          sleep 1
111 esac
```


Expand your Fortune® without spending a fortune.

New equipment costs big bucks. The Good Fortune Trader sells fully tested burned-in equipment, refurbished to Fortune® standards at prices that won't make waves with your cash flow. Call or write today for a listing of our specials.

THE GOOD FORTUNE TRADER

60 DAY WARRANTY

a division of Computer Services of Atlanta

(404) 392-9938

1534 Dunwoody Village Parkway/Atlanta, Georgia 30338

Continued on next page

enter a directory that fails the "-d" option, they will get the error message and our "return.sh" shell script will execute. This is one of the "call" routines we discussed previously. Rather than having to type the lines that asks them to depress the **RETURN** key in many places throughout our shell script, we simply make it a call routine. This is done to insure the user had a chance to see the error message. After the **RETURN** key is depressed, control returns back to the next line in "main.sh". The ;; tells the shell we are finished with this case so control passes to lines 111 and 112 and the menu screen is again displayed and the user is asked to enter a selection. This occurs because the "while" loop says to continue to display the menu as long as the value of the "ext" variable is equal to "n". For all of the menu selections (cases), "ext" will retain its original value of "n" except for selection 7 (line 107) where we set the variable to "y". Therefore, unless the user selects item 7 from the menu, the menu screen will be re-displayed after each "case".

If the user selects menu item 2 (LIST contents of current working library), lines 32-34 of the shell script are executed. The screen is cleared and a list of files is displayed a screen at a time. Again the "return.sh" script is executed to insure the user has a chance to see the last screen displayed. The menu is then again displayed.

If the user selects menu item 3 (DISPLAY data in file), lines 35-45 are executed. The screen is cleared and the user is asked to enter a file name. Line 38 checks to see if the user has read (**-r**) permission for the file name entered. If the file name is not found or if the user does not have read permission, the test will fail and the error message in line 43 is displayed. Otherwise the screen is cleared; the file is displayed a screen at a time; "return.sh" is executed; and the menu is once again displayed.

If the user selects option 4 (COPY data file) from the menu, lines 46-57 will be executed. The screen will clear and the user is asked to enter the name of the file to be copied. His answer is put into the variable "file1". The test in line 49 checks to see if the file has read (**-r**) AND (**-a**) write (**-w**) permission. If he has permission, he is then asked to enter the name for the new file which is assigned to the variable "file2". The copy is then done in line 53. Note that we have not included any logic to test whether "file2" is an existing file name. For the sake of brevity, we have left this out but you would want to give the user the opportunity to exit if he is about to overwrite another file of the same name. If the test in line 49 fails (he does not have permission) then the error message in line 55 is displayed and "return.sh" is executed. In both cases, the menu screen is then re-drawn. There is one test that we forgot to do. Based on the material we have presented over these months, can you figure it out?

When menu option 5 (RENAME data file) is selected, lines 58-70 are executed. This selection initializes the variable "name1" to null. The screen is then cleared and the user is prompted for the name of the file (or directory) to be renamed. The test in line 62 checks to see if the user has write permission AND (**-a**) that the "name1" variable is not equal to null. This is our way of checking to make sure the user entered a valid answer and did not just depress the return key. This also answers the question above - we did not include this test in the previous case. We then ask the user to enter the new name and execute the "mv" command. If the test in line 62 fails, the error message in line 68 is displayed; "return.sh" is executed and the menu screen is re-drawn.

Things in life never stay simple. The commands that are executed when the user selects menu option 6 (REMOVE data file or library) are more complex than our previous examples. There are two reasons for this. One is that we needed some samples of other commands and concepts for the summary. The other is that we always get nervous when we see the word "remove" which necessitates some extra logic to give the user a chance to back out. The user is asked to enter the name of the file/files or library/libraries to be removed and his answer is put into the variable "file". A variable "x" is initialized to 0 in line 74. The "for do done" loop in lines 75 through 96 says that for each of the names he entered, "do" the list of things in lines 77-95.

Here is the logic that is followed for each of the names he entered:

- Line 77 checks to see if the file/directory has read permission. If the user does not have permission or if the file/directory does not exist, the error message in line 93 is displayed and the loop goes on to the next name. If the user does have permission, then execution continues with line 78.

- Line 78 sets the value of the variable "stop" equal to "n". This variable is used in the "while do done" loop in lines 79-91. This logic basically says that while the variable is equal to "n", continue to ask the question in line 80. Since we initially set the variable to "n", the question is asked the first time through and the response from the user is assigned to the variable "del". If the user answers with "Y" or "y" or "YES" or "yes", then we remove the file/directory in line 83. In line 84, we set up a counter using the variable "x" which counts the number of files/directories that have been removed. We then set the "stop" variable to "y" which means that the user entered a valid answer and the "while" loop will not continue to ask the question. If the user answers with "N" or "n" or "NO" or "no", we take no action except to set the variable "stop" equal to "y". In this case the user also entered a valid answer and the loop will terminate. In either of the above cases where the user enters a valid response, the "for" loop continues with the next file or directory in the list. Line 87 executes if the user enters an invalid response (a response other than those indicated in lines 83 and 86).

Continued on next page

The user will get the error message in line 88 and the value of the variable "stop" remains "n" which means control will go back to line 79 and the "remove" question will be asked again.

- After all of the file and directory names the user entered have been processed, control goes to line 97 (the next line following the "for do done" loop). Here we have an example of the "if then elif else fi" command sequence. Line 97 says that if the value of "x" is less than () 10, then tell the user that less than 10 files were removed. Line 100 says that if the value of x is greater than (**-gt**) 9, then tell the user that more than 10 files were removed. While this may not be the most realistic example in the world, it does allow us to demonstrate the use of the "if" command sequence and also gives you an example of how numeric variables are tested.

- Lines 104 and 105 tell the user exactly how many files were removed. The "return.sh" script is then run and the menu screen will be re-drawn. Notice that for this case, we have also left out the test to insure the user does not just hit the **RETURN** key.

If the user selects menu option 7 (EXIT menu), lines 107 and 108 are executed. Actually only one of these commands is needed to exit this shell script. If "ext" is set to "y", then the test in line 3 fails and control goes to the line after the "done" in line 112. Since there are no more commands to execute, control goes back to "trap.sh". There are no more lines in "trap.sh" to execute, so control goes back to wherever "trap.sh" was executed (the program terminates). Using "exit" as we have done in line 108 does the same thing as setting "ext" to "y".

For any other answer the user enters besides a valid option 1 through 7, lines 109 and 110 are executed. The "*)" is executed when none of the previous cases are applicable. In this happens, the bell will ring (**^G**) and the highlighted message will be displayed (**^\H** starts the highlight and **^\I** ends the highlight - space is needed after the H and I). The "sleep 1" command stops the program for 1 second so the user can see the error message and then the menu is re-drawn. We normally would have used the "return.sh" here; however, we wanted to give you an example of the "sleep" command.

That concludes our "INTRODUCTION TO SHELL SCRIPTS" series. We hope you have had as much fun learning about shell script programming as we have had presenting the material. One of our goals was to provide you with another tool to use your system more effectively. Another goal was to present the material in such a way that a non-programmer (with some practice) could begin writing shell scripts. Again, our intent was not to teach every command and concept available in shell programming. We do hope, however, that we accomplished the two goals listed above. □

UNISOFT INC.

Experts in Computer Consulting - Sales - Service

Presents

FORTUNE SYSTEMS HELP

We Provide:

- Expert Trouble-shooting
- Years of programming and accounting experience
- *Free Initial Consultation*
- Training Available for:
  - * BASIC/IDOL/BAS and Multiplan
  - * UNIX and Ext. Fortune:Word
- Hardware Repairs & Upgrades
- FORTUNE trained experts at a cost that will surprise you!
- *Trade-Ins Welcome*
- Specializing in:
  - * Hard Disk Trade-In
  - * Hard Disk Upgrades

Authorized Fortune Systems Repair Depot
Fortune Systems National Value Added Reseller (VAR)
CALL: (303) 671-8600 EMERGENCY HELP: (303) 649-8505

UNISOFT INC. 14100 East Jewell Avenue Suite 27 Aurora, Colorado 80012

UNIX BASICS 5: Redirection & Pipes

In the previous four parts of our series on UNIX Basics, we discussed several useful UNIX commands and some concepts (e.g., file systems and command options). However, every one of the commands and concepts discussed are not really unique to the UNIX operating system. For example, if you have worked on IBM computers with the DOS operating system, then you probably know that the equivalent DOS command for **ls** is dir. Similarly, in the DOS world, copy does what **cp** does in UNIX, rename does what **mv** does, and this list could go on. (Note: This is not entirely true. In general, UNIX commands are *richer* than their counterparts on other systems. For example, **cp** in UNIX is far more flexible than copy in DOS. However, their basic function is the same - to make a copy of a file.)

Also, the newer versions of DOS support a hierarchical file system, like that described in Volume 4 Number 2 of **/u/fortune *news***. The notion of *options* (described last month) also appears on other computers. For example, the command dir/size/protection/owner, issued in a VMS environment (one of the operating systems found on Digital Equipment Corporation computers), would do basically the same thing as **ls -l**.

This month we will discuss some concepts that were once uniquely UNIX: *pipes* and *redirection*. Actually, many other operating systems have adopted these notions because they were recognized as being extremely useful. However, in most cases, pipes and redirection still work best in UNIX. For example, I use a VAX computer occasionally which runs the VMS operating system and the notion of a pipe is, at this time, non-existent; redirection of output to or from a VMS command requires the use of another command. For example, in UNIX, if I want to get a list of the users currently logged into the computer and I want to "redirect" that list to a file, I would type:

```
$ who > who.list
```

But with VMS I would have to type:

```
$  assign/user  sys$output
who.list
$ show users
```

You may not know what's going on here but I think you will agree that the UNIX command is easier to use than the two VMS commands. But we are getting a bit ahead of ourselves.

Standard Input and Standard Output

In order to understand pipes and redirection we need to discuss two related concepts: the *standard input* and the *standard output*. The standard input refers to where a UNIX program will "look" for its input while the standard output refers to where a program will "write" its output. Normally, both the standard input and the standard output are "attached" to your terminal. That is, a program will "look" at whatever you type on the keyboard and will "write" its output to your terminal screen.

Let's get a bit more concrete. The UNIX command **sort** is often used to sort a list of words alphabetically. (Note: This is not all **sort** can be used for. If you want more information on **sort**, see Volume 3 Number 2). Now, the following may be a bit contrived, but it should serve to illustrate what the standard input is and what the standard output is. If I type the word **sort** to the UNIX prompt and then hit the **RETURN** key, I would have started the **sort** command. **Sort** wants to see a list of lines which it will attempt to sort. But where does sort "look" to get this list? Does it look at some file on the hard disk or some file on the floppy disk - yes, where does it look? Well, if you don't tell **sort** anything special, it will expect to find that list on the standard input. Now remember that I said the standard input is originally attached to your keyboard. So, **sort** will be waiting for you to type a list of words in from the keyboard. When you are done typing in a list of words (**sort** knows you are done when it sees a **CTRL-D**, which, by the way, is the signal for an *end of file* in UNIX), **sort** will rearrange that list of words and then send this rearranged list to the standard output (and, of course, you remember that the standard output is originally attached to your terminal). So, if I type the following:

```
$ sort <CR>
Sam <CR>
John <CR>
Harry <CR>
Sylvia <CR>
Mary <CR>
Susan <CR>
Alex <CR>
Wendy <CR>
Zelda <CR>
Fred <CR>
CTRL-D
```

Sort will respond with the following on your terminal screen:

Continued on next page

```
Alex
Fred
Harry
John
Mary
Sam
Susan
Sylvia
Wendy
Zelda
$
```

This is, as you can see, the original list that is now sorted alphabetically. The important point to learn is that almost all UNIX commands read their input from something called the standard input, which is originally connected to your keyboard. Also, UNIX commands send most of their output to something called the standard output, which is originally connected to your screen. Technically, there is one other place that UNIX commands can send output and this place is called the standard error. Generally, all error messages from a program are sent to standard error. The standard error is originally attached, like the standard output, to your terminal screen. In any case, UNIX commands by default, read from your terminal and write to your screen.

Redirection of the Standard Output

However, there are times when you don't want the output to go to your screen. Rather, it would be nice if we could *redirect* what normally goes to the screen to some other place, for example, a file. In our example with **sort**, it was nice that it sorted our list of names for us. However, if we wanted to see this list tomorrow, or at some other time in the future, we would have to re-type all the names and let the **sort** command rearrange them for us. This would be very tedious and not at all in the spirit of allowing the computer to reduce, rather than increase, the amount of drudgery in the world. So, the question is: Wouldn't it be nice if there was some way of intercepting what normally would come to the terminal screen and redirect it to a file which we could look at whenever we wanted? The answer is, of course, that there is a way and an easy way at that.

Redirection of the standard output is accomplished by using the greater-than symbol (**>**). The following example shows how to redirect our sorted names to a file called **sorted.names**:

```
$ sort > sorted.names <CR>
Sam <CR>
John <CR>
Harry <CR>
Sylvia <CR>
Mary <CR>
Susan <CR>
Alex <CR>
Wendy <CR>
Zelda <CR>
Fred <CR>
CTRL-D
$
```

Note that if you did the above just as listed, then **sort** would not write anything to your screen because you asked UNIX to redirect, to a file called

Continued on page 15

"SAVE 85% ON PHONE BILLS . . . ALSO AN EXTRAORDINARY AMOUNT OF OPERATIONS TIME"

says a satisfied user of HANDSHAKE BLAZER. Using HANDSHAKE BLAZER with high speed modems to transmit files between corporate headquarters in Detroit and 13 locations with Fortune Systems throughout the United States and Canada provides an efficient data communications solution making this MIS "very, very happy."

For additional information on HANDSHAKE BLAZER, the file passing software that works with a variety of modems (standard and high speed) over standard phone lines or in direct connect configuration to transmit ASCII and binary files, contact:

SST Data Inc.

(414) 242-3999

250 S. Main St., Suite 201
Thiensville, WI 53092

SAVE A FORTUNE
(QUICK!)

Fortune Systems
Price List

1. 256 KB Memory Boards .....
2. 1 MB Memory Boards (Compatible to 2 MB) .....
3. SX 2MB Motherboards .....
4. Upgrade to 2.0 Operating System .....
5. Upgrade to Fortune:Word 3.0 .....



Don't wait. For a very limited time, Ohio's Fortune Systems Master Dealer is offering amazing discounts on selected Fortune System hardware, memory and software upgrades.

If you're looking for a 1 MB memory board, a Fortune:Word upgrade, or even an SX motherboard with the 2.0 operating system, now is the time to call. We'll send you our price list immediately (we'll even supply quotes over the phone). Out of state inquiries are welcome, too . . . the long-distance call is worth it.

But call NOW. Because supplies on all items are limited.

Micro Resources. We are Fortune Systems in Ohio.

Call (614) 766-2335.

Micro Resources

255 Bradenton Avenue, Dublin, Ohio 43017

FORTUNE SYSTEMS
Authorized Fortune Systems Master Dealer

sorted.names, what would normally be sent to the standard output. At this point, you could use the **cat** or **more** command to view the contents of the file called **sorted.names** (For more on **cat** and **more**, see Volume 3 Number 7).

So, to review, the greater-than sign (**>**) instructs UNIX to redirect any results of a program to a file. The **>** redirects the standard output.

A couple of notes about this type of redirection are in order. First, please be careful when you redirect output from the standard output to a file because if that file already exists, its contents will be replaced by the new output! If the file doesn't already exist, UNIX will create it. But if it does exist, *watch out* - the original contents will be irrevocably erased.

This problem leads to a special variant of redirecting the standard output. Remember that the symbol **>** instructs UNIX to send any output to a file. The special symbol **>>** (two greater-than signs) tells UNIX to take the output and *append* it to the file. This allows us to keep putting new things into a file without destroying what was previously in it.

Redirection of the Standard Input

Let's go back to our sort example. The UNIX command **sort** would be relatively useless if we actually had to type our unsorted list into the computer every time we wanted to sort this list. No, the way **sort** is normally used is to sort the contents of an already existing file. So, we need a way to give a file to the **sort** command. One way to do this is to redirect the standard input. For example, suppose I had the to-be-sorted names sorted from above in a file called **names**. (In this case, I just used an editor to input the information into the file called **names**). I could then direct **sort** to sort this list of names with the following command:

```
$ sort < names
```

The less-than sign (**<**) tells UNIX to redirect the standard input to the file called names. Thus, the computer doesn't look at your keyboard for its input but rather looks at what is in the file.

The above command will output the sorted list onto your screen because although we have redirected the standard input, we have not redirected the standard output. If we want the contents of names to be sorted and sent to a file called **sorted.names**, we would issue the following command:

```
$ sort < names > sorted.names
```

Now for a question to highlight three more points:

Question: Which of the following UNIX commands would ***NOT*** work?

```
1) sort < names >> sorted.names
2) sort <names >>sorted.names
3) sort <names>> sorted.names
4) sort << names > sorted.names
5) sort >> sorted.names < names
6) sort>>sorted.names<names
```

Answer: All of them will work except number 4!

Point number one: The **<<** symbol is invalid. That is, there is no equivalent to the **>>** symbol which appends a command's input to a file.

Point number two: The order of the redirection of the standard output and standard input is not important. A command like `sort > sorted.names < names` may seem backwards to us, but UNIX is able to determine what we want.

Point number three: UNIX interprets certain characters in a special way and the following are special: **>**, **>>**, and **<**. This means that spaces are not necessary. Again, a command like `sort>>sorted.names<names` may look nearly unintelligible, but it is understood correctly by UNIX.

So, What's In Your Pipe, Anyway?

Up to this point, we've discussed how to redirect the output of a program to a file or to redirect the contents of a file to a program. There is one other type of redirection that is very common in UNIX and this type of redirection takes place with the use of *pipes*.

UNIX provides a mechanism for piping the output from one program to the input of another program. In order to illustrate this, let's review what the **who** command does. The **who** command simply lists the names of the people currently logged onto the computer. For example, if you type **who** on your computer, you might see something like:

```
mark     console Apr 22 12:39
john     tty02   Apr 22 12:21
alice    tty03   Apr 22 12:46
```

Now suppose we wanted to see this list in alphabetical order. Well, we could perform the following two commands:

```
$ who > who.list
$ sort < who.list
```

However, there is a more elegant way to accomplish our pedagogical task which uses a pipe. Look at the following command:

```
$ who | sort
```

Here the | symbol (a vertical bar, it is the upper character on the middle gray key on the left hand side of the Fortune keyboard) represents a UNIX pipe. This symbol instructs UNIX to take the output from the **who** command and *pipe* it to the **sort** command. So, in a sense, this one symbol has the effect of redirecting the standard output of the **who** command and also redirecting the standard input of the **sort** command - joining them together so that the one flows into the other. In the above command, we have only utilized the pipe and have not used redirection at all. Thus, the output from the **sort** command will appear on your screen. If you wanted to save the output, you would use a

Continued on next page

command like `who | sort > sorted.names`. Also note that the pipe symbol (|) is used *exclusively* as a means of connecting two commands. One would never try to pipe the output from a command to a file because this is the domain of redirection.

Using pipes can be extraordinarily useful and time-saving. The real beauty of pipes is the ability of piping together several UNIX commands. For example, we have in the past advocated that all owners make a printout of the *conf* block. The information in the conf block could save you hundreds of dollars if your hard disk experiences trouble. So, you really should print this information out and keep it in a safe place. The question then is, how can you print it out?

The **rdconf** command will print this information to the screen. Obviously, through redirection, we could send this information to a file and then print this file out. The following commands would accomplish this:

```
$ rdconf /dev/hd00 > conf.info
$ lpr conf.info
```

You could then take the printed output, date it, and then store it somewhere safe. However, why use two commands when one will do? The following has the same effect:

```
$ rdconf /dev/hd00 | lpr
```

Now, don't forget to date the output. But wait a minute, why should you date this output? We could use another program, called **pr**, to put page numbers and a date on the output. Try this:

```
$ rdconf /dev/hd00 | pr | lpr
```

Now the output on your printer will have the current date on it. Note that the above command was actually three separate commands. First, the output from the **rdconf** command was sent to the **pr** command. **pr** formatted this information and included the current date and then this information was piped to the **lpr** command which took care of the printing.

Conclusion

In this installment of our UNIX Basics series we discussed the notion of redirection in UNIX. The output from a UNIX program normally is sent to your terminal screen. This output can be redirected to a file. Also, most UNIX programs expect input to be entered from the keyboard. This input can be redirected so that the program will look for its input in a file. Finally, the output from one UNIX program can be redirected to the input of another UNIX program via a pipe. □

Mark Palmerino

TRI-TECH COMPUTER SERVICES, INC.

Tri-Tech Computer Services, Inc., has recently become part of the network of Authorized Fortune Systems Affiliate Dealers. We have offices in Newton, MA and Kensington, CT. We provide local on-site service as well as nationwide depot service by Fortune Trained personnel.

We offer the following to any and all Fortune Computer users who obtain either an On-Site or Carry In Maintenance Agreement with Tri-Tech Computer Services, Inc.:

1) A 15% Discount on all subsequent purchases of Fortune released Hardware products (Excluding Tri-Tech Specials).
2) A 20% Discount on all subsequent purchases of Fortune released Software products (Excluding Tri-Tech Specials).
3) Software Telephone Support (Fortune Products).
4) Unix Operating System Telephone Support.
5) Periodic product announcements and informative mailers.
6) FREE 24 month subscription to /u/fortune news.

If you are currently under a Maintenance Agreement, sign with us and we will extend our coverage to include the final (3) three months of that agreement at no additional cost.

Tri-Tech also offers Quarterly Specials. Our current Spring Specials are:

Micropolis 70MB Hard Disk Upgrade (*45% off List with this ad*).
H.P. Series II, LaserJet or LaserJet 500+ (*15% off List with this ad*).
Fortune Used Workstations (*Best price around, quantity discounts*).

Shop - Then Call Us

SERVICE SERVICE SERVICE
Professional Installations

CALL TRI-TECH COMPUTER SERVICES, INC. TODAY !!!
Massachusetts (617)965-4453 / Connecticut (203)828-9784

To receive periodic informative mailers and product announcements, call Victoria Cole in CT and she will put you on our mailing list.

/u/help!

Fortune Systems has released a number of Software Bulletins which may be of interest to our readers. Some are problems you may encounter, and others are useful tips. We are reprinting them here:

Problem: The HP Laser Jet Printer will not print in 16.66 pitch when using Multiplan. (P.S. If your version of **/usr/bin/mp68** looks like Figure 3, you may want to replace it with Figure 2 which will allow Multiplan to use the pitch, wheels, etc from the Multiplan Menu.)

Solution: Look at the end of the file **/usr/bin/mp68**. If it looks the same as the file listed on figure 1, then make the changes listed on figure 2. Add the section contained in the double outlined box, and move the `+P$pitch` to the end of the line as shown in the `exec` command. (Note that the **exec** line is very long and has been wrapped onto several lines in the figure. You should enter it all on one line, or if you need to continue it, put a backslash on the end of the first line.) In the second line of the addition where it says 1), you should use the printer # for your laser printer, e.g. 2), 3), etc.

Problem: The Basic4 to Fortune 32:16 conversion package (version 1.0) does not work after installation of Thoroughbred BASIC 6.5.12.

Solution: Follow these steps:

One: Log in as root.

Two: From the shell prompt, type the command: **`cd /b/B4X`**

Three: Add the following three lines to the beginning of the file **B4Xferimac**:

```
#KEYMAP, COLLATE variables suppor soft
k/b mapping
KEYMAP=${KEYMAP-BDOM} ; export KEYMAP
COLLATE=${COLLATE-NULL} ; export COLLATE
```

Figure 1.

```
case $papertyp in
        0)SHEET=-d;;
        2)case $bins in
                3)BIN=-S1;;
                4)BIN=-S2;;
                5)BIN=-S1,2,A;;
        esac
        ;;
esac

exec /m/mp/mp /m/mp/mp -p "-p
  $printnum $BANNER $SHEET -
  F$formlen $DRAFT +P$PITCH
  +W$prwheel +V$lnperinch" $*
```

Figure 2.

```
case $# in
        2)case $1 in
                -p)shift
                   shift;;
        esac
esac

if test -f .mprc
then
        MPRC=.mprc
        eval `cat $MPRC`
elif test -f $HOME/.mprc
then
        MPRC=$HOME/.mprc
        eval `cat $MPRC`
elif test -f /etc/.mprc
then
        MPRC=/etc/.mprc
        eval `cat $MPRC`
else
        pitch=10
        printnum=1
        banner=no
        papertyp=1
        bins=3
        prwheel=1
        formlen=66
        lnperinch=6
        draft=no
fi

case $banner in
        no)BANNER=-h;;
        *)BANNER=-b;;
esac

case $draft in
        no);;
        *)DRAFT=+D;;
esac

case $papertyp in
        0)SHEET=-d;;
        2)case $bins in
                3)BIN=-S1;;
                4)BIN=-S2;;
                5)BIN=-S1,2,A;;
        esac
        ;;
esac
```

```
case $printnum in
        1) case $pitch in
                15)pitch=17;;
                ;;      esac
        ;;
        $);;
esac
```

```
exec /m/mp/mp /m/mp/mp -p "-p
  $printnum $BANNER $SHEET -
  F$formlen $DRAFT +W$prwheel
  +V$lnperinch" +P$pitch $*
```

Figure 3.

```
case $# in
0) exec /m/mp/mp /m/mp/mp;;
*) exec /m/mp/mp /m/mp/mp "$@";;
esac
```

Problem: After installing **FOR:PRO 2.0** over **FOR:PRO 1.8** or **FOR:PRO 1.8.1.1**, the system will not allow a user to complete a system shutdown from the login screen (as opposed to the Global Menu).

Solution: If version 2.0 was installed over 1.8 or above, through the **S5** option in the Global Menu, and you want users to be able to shutdown the system from the login screen, then the shutdown entry (in the **/etc/passwd** file) should resemble the following line:

```
shutdown::0:1:& computer:/:/etc/shutdown
```

It is also highly recommended that the shutdown account be password protected.

If the above change is not made, then you will need to be logged in as **root** or **manager** in order for a system shutdown to take place.

Problem: (Not so much a problem as an important note.) During the installation of **Fortune:Word 3.0**, there is an elapsed time of 5-6 minutes from the time installation commences until the first dot (".") appears on the screen.

Solution: Wait the required amount of time necessary for the diskettes to install. DO NOT DEPRESS THE CANCEL/DEL KEY!!!! Doing so will cause the diskettes to be damaged and will prevent any further attempts at a successful installation. The diskettes will have to be replaced at an additional cost. □

But What I really Want to do is Direct. . .

Jeffrey Fiskin, who wrote the screenplay for the film ***Cutter's Way,*** *lives and works in Hollywood.*

You might think the problem is endemic to Hollywood. Naw. New York, Pittsburgh, Seattle, London. Even happened in Tunis once. They hear you're a screenwriter and sure as fruitcake turns blue at the back of the fridge they have a great idea for a movie. So you listen, hoping to hear an idea good enough to steal. Never happens. And nowadays technology has added an even more boring wrinkle: "Which word-processor do you use?" If you answer WORDSTAR, MicroSoft's WORD or WORDPERFECT (the word-processors of choice in Hollywood) you are in for a long night of techno-babble. But if you can say **FORTUNE:WORD**, you're home free. Nobody uses **FORTUNE:WORD** for screenplays. Except me. But then, as Thomas Carlyle said, "Every great idea is, at its inception, in a minority of precisely one."

Why **FORTUNE:WORD**? Glad you asked. When I first investigated computers some seven or eight years ago, the choice was simple. There was WANG and WORDSTAR. Wang was the obvious, if pricey, choice. WORDSTAR was ridiculous, an absurdly complicated amalgam of control codes and recondite mnemonics demanding months to learn and longer to get "up-to-speed." Unfortunately WANG was corporation-oriented and quite unprepared to service a single user. Besides the old Wang unit was ugly. So I kept looking. And then one day, I saw a photograph of MY computer in a "Buyer's Guide" magazine. No clumsy elephantine box this, but a graceful, stream-lined machine. I made a foolish presumption. Surely a unit as attractively designed as this must have an equally elegant word-processor to run. Nine times out of ten I would have been wrong. But **FORTUNE** was that all-important tenth time. Indeed, the word-processor–in fact a Wang clone–was as streamlined, graceful and elegant as the computer.

That was the start of beautiful relationship. Three generations of computers have come and gone. Hundreds of word-processors have emerged and faded. UNIX has NOT become the promised industry standard. But **FORTUNE:WORD** is still the best word-processor around. XYWRITE may be faster, NOTA BENE may do more stuff. But for sheer elegance, simplicity and ease, it's **FORTUNE:WORD**. You see the whole point of a great word-processor is to stay the hell out of the way of the creative process. If you've got millions of synapses busy remembering thousands of **ALT-CTRL-SHIFT** combinations, your mind isn't on story, character, description and dialogue. Paramount and Warner Brothers and Twentieth Century-Fox don't spend the big bucks for **ALT-CTRL-SHIFT**.

So let me tell you all you need to know to write a movie with **FORTUNE:WORD**. It's easy. The fame and fortune part is up to you.

* * *

The only tricky part of script-writing --aside from story, character, description and dialogue–is the screenplay form, the layout of the page. And that isn't very tricky either. There are just two modes: description and dialogue. You switch back-and-forth. That's where the computer comes in handy. Now if you want to get real technical, there are some differences between screenplay form and teleplay form, but nobody ever got turned down for confusing the two. Basically they both look like that shown in figure 1.

That's it. Piece o' cake. Two formats: one for description, one for dialogue. And FORTUNE:WORD'S glossary puts them only two keystrokes away. The glossaries are shown in figure 2. You can add a lot more of course, but this one gets you up and running.

Now the first thing you need to do is change the document's default format. The standard script "description" line runs 60 spaces. So hit **FORMAT**, run the cursor along the **FORMAT LINE**, stopping along the way to insert **TABS** at 11,26 and 48, then continue just past the "6" and hit **RETURN** and **EXECUTE**. Voila! That's "Format 1."

Next you need to set up the DIALOGUE FORMAT. So you hit **INSERT**, then **FORMAT**, then run your cursor to "48" and hit **RETURN** and **EXECUTE**. That will be "Format 2". Done. That's it. All of it.

If you want, you can store the formats in a prototype document called SCRIPT. Or do it from scratch each time you start a new script. Takes about a minute.

Then you attach your "script" glossary and get to work. Whenever you introduce a new character, you just add the name to your glossary like this: Hit the **MODE** key (**F 15**), the screen asks "What mode?", so you hit the **LF/GL** key and the computer flashes the "glossary entry" sign at the bottom of the screen. Now you add your entry: Hit **LF/GL** again, then hit **2**,–the computer fills in the dialogue format--then type in the new character's name, hit **RETURN** and **INDENT** (**F1**) and you're almost ready to go. Hit **MODE** again and when the computer prompts you for "Which entry", type in a letter that will remind you of the character's name. Say the name is SAM, so you hit **s** or **S**. The one trick here is to make sure no

See Directing, Page 20

Figure 1.

```
INT. AN OFFICE. NIGHT.

An extremely handsome YOUNG MAN called
CHARACTER NAME pours himself another
Perrier and offers a toast to SOMEONE OFF-
CAMERA.

                    CHARACTER NAME
          To us, my darling. I couldn't have
           done it without you...

PULL BACK TO REVEAL:

He is talking to his extremely handsome
FORTUNE 32:16 computer.

                    CHARACTER NAME
          ...And to think I almost bought a PC
           running WordStar.

                                   FADE OUT.
```

Figure 2.

```
entry 1
{
insert copy format "1" execute(2)
}

entry 2
{
insert copy format "2" execute indent(2)
execute
}

entry 3
{
gl "1" indent(3) "CUT TO:" return(2)
}

entry c
{
gl "2CHARACTER NAME" return indent
}
```


Our Memory Doesn't Cost You A Fortune

Would you like to:

Speed up your computer?
Add more users?
Use the latest software, like Fortune:Windows™?

Compatibility:	The F1MB is absolutely 100% compatible with Fortune's own 1 Meg memory board.
Affordability:	USA is proud to have equaled and exceeded the quality and performance of our competition, while offering substantial savings for both the dealer and end user.
Dependability:	All boards are burned in for 72 hours and tested for 48 prior to shipment. Our One Year Repair or Replace Warranty underlines our confidency in the product.
Repairability:	The F1MB facilitates repair in the field as it is 100% RAM socketed. Now you can repair a board by simply replacing a defective chip rather than face an expensive board exchange -- reducing maintenance costs and system down time.

If you have a Fortune SX, XP45 or recent model XP30, you can use our board by simply plugging it in. Older systems can be upgraded to use our board with a simple and economical modification.

For Details Call:

Upgrade Service Associates

850 Lexington Street, Suite 15, Waltham, MA 02154 Phone: (617) 239-8147

Our Guarantee: If you're not completely satisfied within 30 days, USA will gladly refund your money.

Directing

continued from page 18

more than two characters share the same first initial or you're back to worrying about too many mnemonics.

When you want to switch to "description mode", you just hit **LF/GL** and 1. You're there. Dialogue? Either hit **LF/GI** and **2** and add the new name, **RETURN** and **INDENT**, or **LF/GL** and your character name mnemonic. Back and forth until you hit page 120. That's more than enough pages. Studio executives are over-worked as it is and do not take kindly to long scripts.

Some script writers add a lot of other glossary entries for things like repeated location directions in "description mode" such as EXT. (Exterior) and INT. (Interior). But there's always a point of diminishing returns. Sooner or later you have to remember too many glossary entry mnemonics which is why you didn't buy WORDSTAR in the first place. So add them if you must–it's certainly easy enough–but don't forget, the whole point of this is story, character, description and dialogue. Anything that helps you get there is good. That's what **FORTUNE:WORD** lets you do. And stays the hell out of your way. The rest is just the agony of the blank page.

See you at the Academy Awards. □

Affordable laser printing for Fortune

Laser Printing! Kiss the expense good-bye.

QMS — KISS laser printer $1995.00

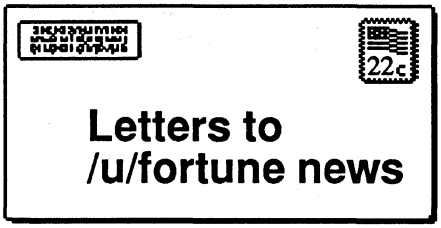
- cost per page 1.4¢.
- 9 resident fonts
- Print cap and print utility included.
- add-on sheet feeder provides total of 3 trays and envelope feeder.

THE GOOD FORTUNE TRADER

QMS - KISS from The Good Fortune Trader
1 Year Warranty

a division of Computer Services of Atlanta
(404) 392-9938

1534 Dunwoody Village Parkway, Suite 109, Atlanta, Georgia 30338



Letters to /u/fortune news

Dear Mr. Wannall,

Choice Systems, the systems division of Choice Retail Systems, was acquired by Choice Technical Services on September 1, 1986. Our address and phone number appear at the end of this letter.

BAS PAYROLL 2.0 is **not** a product of Choice Systems. Fortune had **PAYROLL 2.0** when we first talked to them about fixing the various bugs in their BAS product.

Our contract with Fortune specifically excludes PAYROLL 2.0. However, we received a copy of 2.0 Payroll in late February and have already documented quite a few problems including the difference in the file layouts. We are currently trying to determine the most expedient means of supporting 2.0 Payroll.

I feel it is important that you and other ISV's understand that both Fortune and Choice agreed that we would provide support for **non-modified** versions of the BAS packages. Our support contract is for **Payroll 1.2** and **BAS 1.3**, and is primarily for Fortune Dealers and VARs. We only provide support for end-users who are "orphans."

The BAS package provided a general accounting solution that was a **base** package for VARs, ISVs, etc., to provide specific solutions to their vertical markets. While there are quite a few non-modified installations of BAS, there are just as many **modified** installations. Those of you who provide software customization services or customer applications to BAS installations must re-apply your modifications to the new release of BAS, after you have determined that the changes that were made do not affect your modules.

The ease with which you can make program changes to BAS modules was one of the reasons for the popularity of BAS. That is also the reason why general update releases are not simple load and go procedures.

This applies to all BAS accounts on all hardware. We directly support installations on Fortune, Arete, Zilog, Pertec, Seiko, Altos, Basic Four, and Plexus computers.

We provide telephone support on an hourly basis to persons having a **signed support agreement** on file in our offices. We do **not** sell or distribute BAS modules for Fortune computer systems.

Ben Alexander, President
Choice Technical Services
2254-H Northwest Parkway, Marietta, GA 30067
(404) 980-1446

BASIC Advisor

continued from page 1

Dear Ray: I am a Fortune Systems dealer who, for one, was glad to see someone open a forum on the 2.0 version of the BAS software. I, too, have questioned the sanity of the upgrades and am looking forward to reading any responses you may get to your "open letter".

Dear Ray: As an authorized dealer for Fortune, I do not recommend any of the 2.0 applications to my prospects. The old versions were just fine.

Dear Dealers: Read on. The following came from **Mr. Jim Jones** of Jones Business Automation Center in Charlotte, NC.

Dear Ray: I believe you have been terribly misinformed about the authorship of the 2.0 BAS Payroll. In fact, it was not written by Choice, but by Fortune. We were a beta installation for the Payroll upgrade back in 1984. Allow me to straighten out your facts.

In late 1983 or early 1984 Fortune released their 1.1 version of the Payroll. There were many problems with it, so it was pulled and reworked. In April, 1984, release number 1.1.1 was put on the market. This, too, had problems and was pulled. This is when we came into the picture. We worked with a Fortune programmer in developing the 1.2 version, which was released in January, 1985. This version contained our upgrade to the Gross-To-Net processing program. We continued to work with the Fortune programmer, and version 1.2.1 was released on May 17, 1985 (after Fortune had fired the programmer). I believe that this 1.2.1 version is, in essence, what is now being called 2.0.

As far as the changes to the Payroll Time Sheet file are concerned, putting the payroll check number into the file key solves problems that existed in updating manual and canceled checks. The department code and worker's comp code in the key allow for a more detailed interface with General Ledger.

If you look back to the announcement released by Choice regarding upgrades to the BAS applications, you will see that Accounts Receivable, Accounts Payable, General Ledger, Order Processing, Purchase Orders and Fixed Assets were all involved. Payroll was omitted from the list. As I have explained, the reason for this omission is that 2.0 Payroll is from Fortune, not Choice.

Dear Mr. Jones: I want to thank you for filling in the holes in my history of Payroll. I discussed our conversation with another Fortune dealer who, over the years, has worked very closely with the Fortune technicians responsible for BAS upgrades and support. Here's his commentary.

*Dear Ray: I was glad to see your letter to Choice in /**u/fortune news.** I am as frustrated as you are with the new Payroll. As you know, I was present when the Fortune Systems executive quoted $2.3 million as the contract value for BAS*

continued on next page

G.T. Hawes & Co., Inc is BACK!!!

with prices that have never been:*cheap, cheap, cheaper!!!*
and delivery that is: *fast, fast , faster!!!*

G.T. Hawes & Co., Inc.

(No Longer an Authorized Fortune Master Dealer)
(No Longer an Authorized Fortune VAR)

G.T. Hawes & Co., Inc.

475 West Merrick Road
Valley Stream, NY 11580

(516) 627-5556 Call George Today (516) 568-0044

upgrades. It did seem awfully high at the time for the work required.

I was personally acquainted with the Fortune programmer who was the primary author of 1.1.1 Payroll. That version had it's bugs, but was a very workable product. I cannot believe this programmer would ever create the 2.0 monster. Putting the check number in the time sheet file key demonstrates a minimal comprehension of programming protocol.

Dear Dealer: Apparently the $2.3 million was a high figure, in spite of what you and I were told. The following letter came from Ms. **Elizabeth M. Muth**, Manager, Software Services, Fortune Systems Corporation.

*Dear Ray: I read, with interest, your letter in The Basic Advisor section of the February edition of /u/**fortune news**. For the record, I would like to correct some apparent misinformation regarding the agreement between Choice Retail and Fortune Systems Corporation.*

Your comment about a supposed 2.? million dollar contract caught my eye and prompted me to reread the contract one more time. In fact, the total agreement was for $42,000 which included both development and maintenance costs. This, as you can see, is a far cry from the information you received. Fortune Systems contracted with Choice Retail to make fixes to the software as well as provide support to the dealer network. I must say, however, that there were no requests, at least none that I can find, to change the payroll application.

If you wouldn't mind, I would appreciate getting a copy of any correspondence you might receive from Choice as a result of your letter. In order for me to resolve any problems the field is having getting support, etc. from Choice, it's necessary that I know the specifics. Thanks.

I hope the above clears some misunderstandings.

Dear Ms. Muth: I heard from your colleague, Mr. **Jim Dyson** of Fortune Systems in Monaco. As you know, Mr. Dyson has been with the company since 1982 and is one of only two people still with Fortune who were involved in negotiating the Choice contract. His comments follow.

Dear Ray: It has been a couple of years since I helped negotiate the contract with Choice, but I think I can help clear up some of the questions you raised in your letter.

At the time, we had already spent much time and money supporting the BAS business application packages. It was estimated that in order to continue, Fortune Systems would have to spend much more, maybe even the $2.? million you mentioned. I have a feeling you confused this number with the actual amount we agreed to pay to Choice, which, as I recall, was about $35,000 plus maintenance agreements. For the money, Choice was asked correct about 50 technical problems which were still unresolved. Choice was to benefit additionally by writing service contracts and maintenance agreements with Fortune Systems end users.

The 2.0 Payroll was ours. Payroll had been the most difficult application to support, so at least seven people, including the programmer Mr. Jones mentioned, produced the upgraded version. Multiple Workman's Comp codes were provided to solve problems users were having with different rates for different job categories. Another big problem at the time was the change in employer/employee FICA contributions. BAS Payroll was not originally designed to handle one rate for the employer and another for the employee. Our upgrade resolved this. But the biggest change was to the work files. 2.0 allows for multiple state and local taxing authorities and rates. I suggest you look into these files if you want to see where payroll was really overhauled.

Choice Systems provided fixes for all of the other applications. 2.0 releases of everything except Payroll are now, essentially, debugged versions of the original packages.

When the contract was signed, both we and Choice estimated that there would be a lucrative support and maintenance business for Choice. As it turned out, we were both wrong. There was extensive local support available for Fortune Systems users. All of you in the field were doing such a good job, Choice was unable to significantly penetrate the market. If it appears that Choice has fallen down on the job, this is probably why.

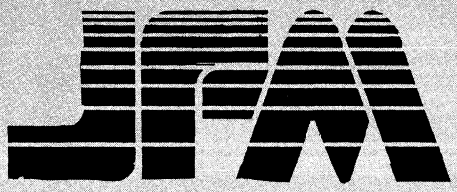
I want you to know that if you need answers in the future I will be happy to speak with you. I appreciate the efforts of /u/fortune news and the Basic Advisor and hope you will both continue your support of Fortune Systems.

Dear Mr. Dyson: It is nice to know that there is someone willing to give us a hand. Reliable information is sometimes very difficult to find.

***Question**: Were you also aware that the 2.0 Payroll does not provide a program to print 941 forms? I have been told by a Choice support technician that the Quarterly Tax Report gives all the information needed for the end user to type up his own 941. Don't you think this defeats the purpose of having a computer?*

Answer: Now that's a new can of worms! Maybe we should all lobby for a 2.1 Payroll. We could also request that our time sheet sequence counter be returned. The problem is, who could we get to do it?

(P.S. As /u/**fortune *news*** went to press, I received a letter from Ben Alexander, President of Choice Systems. I appreciate his repsonse and the text of the letter can be found in the *Letters from our readers* column.) □



Dear Fortune Owner:

As you have probably read in */u/fortune news,* Fortune Systems SX models are significantly more powerful and faster than the PS and XP models. If you have a PS or XP model system, you should take advantage of our special promotion on SX upgrades. For a limited time, and while our supplies last, we will upgrade your system, including the new SX Cabinet at no additional charge for the cabinet. With this special offer, you may choose any combination of the following upgrades and/or expansions:

- *SX Processor with Current Release of Multi-User For:Pro*
- *60 MB Internal Streaming Tape*
- *30 MB, 45 MB, 70MB High Speed Hard Disk*
- *6-Port Intelligent Asynchronous Controller*
- *Upgrade to Current Release of all Fortune Software*
- *Fortune:Windows (Included at NO Charge)*

Additionally, we can perform these upgrades within 24 hours of receiving your system. In fact, if you ship your system overnight on a Friday for Saturday delivery, we will have it back to you on Monday morning, all ready to plug in and run; not losing the use of your system for a single work day. And, we will issue you a full 90-day warranty on the entire system.

JFM provides services to Fortune Systems owners either directly or through our authorized Affiliated Dealers. If you haven't used our services, I invite you to discover JFM; you will be glad you did.

Sincerely,

Morris Frydman
President

Talk to us, then talk to our clients



JFM BUSINESS SYSTEMS
The Merchandise Mart
Lobby Suite 144
Chicago, Illinois 60654
Call (312) 644-0870

Authorized Fortune Systems Master Dealer

BACKUP AND RESTORE UTILITY

PROTECTION MADE EASY!

- o Back up all or selected user directories
- o Restore all or selected user directories or files
- o Format diskettes
- o Review backup volumes
- o Menu-driven, easily installed
- o Complete documentation

Also copies the files necessary to preserve your system configuration--a tremendous help if your system ever needs to be cold booted!

$150.00 Backup/Restore Utility
$200.00 Utility + 2 boxes 5 1/4" 96 TPI high-quality diskettes

For more information, call CTD at (313) 540-1282.

To order, make check payable to *Computer Trade Development Corporation.* Send to:

BACKUP/RESTORE UTILITY
Computer Trade Development Corporation
30600 Telegraph Rd, Suite 1386
Birmingham, MI 48010

CTD

Company:_____
Contact:_____
Address:_____
City:_____
State:_____ Zip:_____

DO YOU KNOW...

there is an organization that has a thorough knowledge of

FORTUNE BAS
MULTIPLAN IDOL

We have been providing widespread support to our Fortune clients for more than 4 years.

If you need:

- training
- support
- maximum computer performance
- peace of mind relating to processing
- customization
- application results
- minimum errors and confusion
- better ideas on how to increase the power of your Fortune

Call Steve Rosenfeld at (314) 837-7111

SUPERIOR COMPUTER SYSTEMS, INC.
917 N Hwy 67, Suite 105
P.O. Drawer T
Florissant, MO 63031
Making computers work for you

NOTICE: Many of you have noticed that our address is now 73 Jason Street in Arlington, MA. However, there are some stubborn souls out there whose computers still have 20A Prescott Street as our mailing address. The people at that location appreciate your cards and letters, and especially your checks, but we're getting poor. Please verify that the address below is in your computer for The Cambridge Consortium, Inc. Thanks. (P.S. We know how computers can be sometimes.)

/u/fortune *news*

73 Jason Street
Arlington, MA 02174

/u/fortune* news

The Newsletter for Users of the Fortune 32:16 Computer — May 1987/Volume 4 Number 5

New Features–New Functionality

A Review of Fortune:Word 3.0–Full of New Features!

New Fortune:Word 3.0 Debuts

As previously announced in this publication Fortune Systems released **Fortune:Word 3.0** this spring. The new version includes many enhancements to the word processor, including such features as **Autosave, Floating Footnotes, Right-flush Tabs, Sorting,** and **Wildcard Searches.** We have used the new product for several weeks and can report that many of these changes are very real improvements. This article contains a description of many of the new features.

What's new in 3.0

What follows is an alphabetical listing of the major new features of **Fortune:Word 3.0:**

Alternating Headers and Footers

If you want to have different headers and footers for the odd and even pages of a document, this can easily be done. This is helpful for documents that will be bound, such as this newsletter, where the page numbers should always be placed on the outside edge of the newsletter.

Autosave

The Autosave feature automatically saves your document after a predetermined number of keystrokes have been entered. You can set this number anywhere from 1 to 32,767. When the appropriate keystroke has been reached, the "writing pages" message will appear as the document is written to the hard disk. Although you will not see what you type while this is happening, your keystrokes will appear on the screen as soon as the writing is completed. The **Autosave** feature insures

The Autosave feature automatically saves your document and protects it.

that in the case of a system problem, the most you will loose is something less than the number of keystrokes you have set — e.g. what you have typed since the last save. This is a very useful feature.

Since the document is constantly being updated, it isn't possible to return to the version that existed at the start of the editing session. However, by simply hitting `F8 (STOP)` followed by `F11 (Copy)`, you can copy your document to another file with a new name at any time you desire — directly from the editing screen. This makes it easy to save several versions of a document as you make changes.

See **Fortune:Word,** Page 18

News From Fortune

SCI Update

In our last issue, we reported that **SCI Systems, Inc.** was negotiating to buy the computer operations of Fortune Systems. After due diligence with the two companies, an agreement was reached, which is now subject to approval by Fortune's shareholders. (*A shareholder's meeting will be held at 9am on June 29th in Redwood City, CA to vote on this issue.*) It is expected that this sale will be approved.

If the sale is completed, **SCI** will take possession of the computer operations of **Fortune Systems.** This includes their hardware, manufacturing, sales and marketing divisions. **Fortune Systems** will

See **News**, Page 2

In This Issue...

News from Fortune...Page 1

System Administration - Dave Kloes starts a new series in The UNIX Directory...**Page 4.**

UNIX Basics - Special characters in UNIX...Page9

The BASIC Advisor - LIFO/FIFO problems and more on error 103s...**Page14.**

/u/help - Data verification with **cp** and help for Fortune:Word glossary writers...**Page16.**

Review of Fortune:Word 3.0...Page 1.

*/u/fortune *news* is published by The Cambridge Consortium, Inc. It is not affiliated with Fortune Systems Corporation.

/u/fortune *news*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz

Technical Editor
Scott Bradner

Advertisting Manager
Mark Palmerino

/u/fortune *news* is published 12 times per year by The Cambridge Consortium, Inc. Neither **/u/fortune** *news* nor The Cambridge Consortium are affiliated with Fortune Systems Corporation.

Subscriptions cost $48 for one year within the U.S.A. For foreign subscriptions, add $27 each year for added postage and handling costs.

Back issues are offered at:

Vol 1, Numbers 1 - 3	$1/each
Vol 1, Number 4	$.50/each
Vol 1, Numbers 5 - 8	$3/each
Vol 2, Numbers 1 - 7	$3/each
Vol 3, Numbers 1 - 12	$4/each
Vol 4, Numbers 1 -	$4/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $10 each. Please add $5 per order for foreign destinations.

Although every attempt is made to insure that our contents are error-free and without risk, The Cambridge Consortium is not responsible for losses that occur due to correct or incorrect interpretation of **/u/fortune** *news*.

For more information, call or write:

/u/fortune *news*
73 Jason Street
Arlington, MA 02174
(617) 663-7700, (617) 646-5603

The text for this newsletter was prepared using a Fortune 32:16 for text input. The information was then transferred to an Apple Macintosh Plus with kermit and formatted for typesetting using the Aldus Pasgemaker program.

UNIX™ is a trademark of Bell Laboratories, Inc. Fortune 32:16™, Fortune:Word™ and FOR:PRO™ are trademarks of Fortune Systems Corporation. Multiplan™ and MS-DOS™ are trademarks of Microsoft. IBM-PC™ is a trademark of IBM. Macintosh Plus™ is a trademark of Apple Computer. Pagemaker™ is a trademark of Aldus Corporation.

©1987 by the Cambridge Consortium, Inc. 5/87

News

Continued from Page 1

retain their cash balance, tax-loss credits, and software company, **Tigera**.

We spoke with **Ronald Borelli**, Senior Vice President of SCI's Computer Division, who reported a strong commitment to the Fortune product line. "We intend to protect what we bought—we're in this business for the long term", he said. His attitude towards change was a careful one. SCI will spend some time getting to know the computer business ala Fortune Systems. They don't plan any imminent changes, and are open to suggestions about improving Fortune's existing products. As he stated in a recent press announcement, "Our plans are to supplement the Fortune Systems product line with SCI products and SCI's financial, manufacturing and technological strengths to enable us to grow this segment of the business two- to three-fold over the next few years. Our future products will draw from the strengths of both product line organizations to develop higher-connectivity systems with broad networking capabilities and improved application software. The immediate task at hand is to protect our investment by making the transition of the Fortune Systems operation to SCI's Computer Division as smooth as possible, and to effectively communicate to the Fortune distribution network our long-term commitment to this business." It is expected that the manufacturing division of Fortune will be moved to SCI's corporate headquarters in Huntsville, Alabama, while the sales and marketing divisions will remain in California.

Fortune Systems Corporation has announced its intent, subject to approval at the shareholder's meeting, to change its corporate name to Tigera Group, Inc.

SCI has announced the sale of Fortune Systems' international operations to a new company, Fortune Systems International, based in Monaco and headed by Robert Davis, Vice-President of Fortune Systems Corp, and the Brooke P. Taylor, former president of Fortune's international division. Taylor is chairman of the new company, with day-to-day executive responsibilities, and Davis is president. Bob Davis will help manage the domestic operations during the transition period.

As noted in the last issue, SCI Systems is a major manufacturer of computer products. They are among the top five purchasers of integrated circuits in the world. They produce many of the printed circuit boards for IBM's line of personal computers. In addition to the hardware they produce under government contracts, they make computers for the airline, banking, and process control industries. Many of these latter applications run on their line of Intel 286 and 386 based products, which operate under UNIX System V.3.

Primarily an OEM manufacturer, the purchase of Fortune Systems computer division by SCI represents one of their first ventures into the retail market. We have spoken to a number of dealers and Fortune employees who are all enthusiastic about the SCI purchase. A recent meeting that was held for the Master Dealers and VAR's in Huntsville generated enthusiasm and optimism amongst the attendees. In addition to the manufacturing expertise, financial resources and product line expansion, SCI's involvement may finally lift the cloud of uncertainty that Fortune Systems has operated under for quite some time. SCI's announcement of their third quarter results yields a different picture than the financial picture often chronicled in these pages. For the quarter ending March 31, 1987, they realized Net Income of $10,775,663 on sales of $387,258,676. It's clear that Fortune Systems is moving into a different league.

As more information becomes available on this issue, we will continue to keep you informed.□

New VAR's and New Software

Press Releases from Fortune Systems

Fortune Systems Introduces Two New Value-Added Resellers

The Logic Works, of Fairfax (VA)

The Logic Works is one of the first of seven value-added resellers (VARs) for Fortune Systems Corporation. In business for more than four years, The Logic Works specializes in custom-designed UNIX-based computer solutions for the administrative side of associations, political action committees and lobbying organizations.

"We chose to sign a VAR agreement with Fortune Systems," says **Nat Heiner,** vice president, operations and development for The Logic Works, "Because, after employing their products for nearly five years for our own software development, we feel that Fortune Systems has the best office automation hardware and software products on the market today. We're especially impressed with the superior quality of Fortune:Word, Fortune Systems' top-flight word processing package."

Heiner projects a minimum of 15% growth per year in sales of Fortune Systems products over the next three years. Says **Paul Olin**, vice president, sales for Fortune, "A Fortune VAR is a company that markets nationwide a specific software package which has been ported to Fortune Systems' hardware. The hardware and software are sold together as a total solution. We're particularly pleased to be able to offer The Logic Works' solution for associations to Fortune Systems dealers and users across the country."

ASI Computers, Inc (New York)

In business as a VAR for more than seven years, ASI specializes in custom-designed computer solutions for the education marketplace.

We're committed to UNIX," says **Henry Weingarten**, president of ASI, "and we feel that Fortune Systems has developed a stable implementation of UNIX in its supermicrocomputer line. We chose to sign a new VAR agreement with Fortune Systems because the company's products have an excellent price/performance edge over the competition." Adds Weingarten, "Fortune is a dealer-oriented company and we're very impressed with its responsiveness to the needs and concerns of its dealers and VARs." Weingarten projects a 50% increase in sales of Fortune products by ASI during 1987.

Fortune Systems and UNIFY Announce Agreement to Port UNIFY DBMS Software to Fortune FORMULA

Unify, which originally ported its DBMS to the Fortune 32:16 in early 1984, will port its software to the FORMULA. Under terms of the new agreement Fortune Systems will be an OEM distributor for Unify for both the 32:16 and the Formula products.

As a Unify distributor, Fortune can offer its customers both its proprietary office automation solutions as well as access to the numerous vertical applications that have been written by third parties on Unify DBMS. This is consistent with Fortune Systems' strategy of offering total solutions to the small organization and providing additional software to its installed base.

"Due to the popularity of UNIFY DBMS in the UNIX market," **Paul Olin**, "it was natural for Fortune to offer this database on our systems. Unify has been working with Fortune on selected accounts in the past, and now we can offer this product to our entire customer base."

Says **Anita Massey**, product manager for Unify, "We recognized Fortune Systems as an important player in the multi-user computer field and we're pleased that Fortune will now be able to offer Unify DBMS directly to their growing customer base."

Fortune Announces a new Job Costing Package from BaSiC Software Corporation (Baltimore, MD)

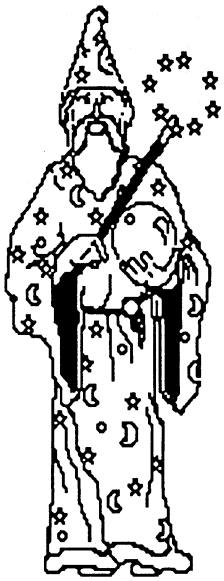
From residential construction to commercial landscaping, the financial success of many businesses depends heavily on meeting or beating estimated job costs. Often actual costs fluctuate continually throughout the course of a project. Equipment failure, delayed deliveries, labor shortages, work change orders, and uncooperative stretches of weather can lead to major job setbacks.

Tracking deviations from schedules and specifications can become a monumental task, even for companies billing no more than $200,000 to $5,000,000 annually. The ability to see, in a matter of minutes, how all of a project's interacting elements fit into place at a given time can spell the difference between heading off a potential problem profitably or facing a no win, no gain situation.

The BaSiC Job Costing software package can put vital job cost information at the business manager's fingertips. It's easy to understand and can generate a variety of project status reports, alleviating many expensive hours of tiresome manual record keeping, cross referencing and calculating.

For further information, contact **Lynn Crawford** or **Ray Wannall** at BaSiC Software Corporation, 301 448-9460.

*****



System Administration

No Longer Requires a Degree in UNIX Wizardry

Beginning with this issue, Dave Kloes starts a series on the art and skill of System Administration. This series is a must for all owners and users of the Fortune computer. Dave is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the Fortune Computer, and is president of the Houston UNIX User's Group. He contributes independently to ***/u/fortune news.***

For those of you that are **system adminstrators** out there, we present this series of articles that will hopefully make your job easier.

We are not going to teach you how to use the Fortune menu system - you can learn that by reading a Fortune manual. We also are not going to teach basic UNIX commands and concepts - we assume you are already familiar with these. What we are going to talk about is what goes on behind the scenes in **FOR:PRO** (UNIX) when you perform many of the functions on the menu system. In our experience with UNIX based systems, we have seen many menu systems that make the task of system administration *user friendly*. In many cases, the actions are performed without a clear understanding of what is being done. In addition, while menu systems are helpful, they cannot solve all of the problems that a system administrator may encounter.

Our goal is to help you understand the files, commands, and concepts that a system administrator needs to know to resolve problems without relying on the Global menu system. Unfortunately for you (and fortunately for us), this opens up a world of subjects that can be covered in this series of articles. The material presented is an accumulation of working with Fortune and other UNIX systems over the past six years. We hope the material will help you to know your system better. More importantly, we hope that by the end of the series, you will be able to resolve many of your own system problems.

Are you or have done any of the following on your Fortune system?

1. Add or maintain user login accounts.
2. System backup.
3. Restore system or user files.
4. Coordinate addition of peripherals (printers, terminals, etc.).
5. Insure system integrity and security.
6. Load software packages.
7. Monitor and coordinate computer usage.
8. Determine additional hardware requirements.
9. Coordinate computer training.
10. Coordinate computer consultant activities such as programming.
11. Coordinate internal programming, if applicable.
12. Load system updates.
13. Coordinate system hardware maintenance.
14. Resolve system related problems.
15. Bring the system up or shut the system down.

If you are doing or have done any of these activities, then you are performing some of the duties of the system administrator. As you can see from this list, the system administrator has a lot to be responsible for. In many organizations, the system administrator usually gets the job by default. It is usually a person who has an aptitude or interest in computer activities. In many organizations, no one is specifically assigned the system administrator duties. It may be several people who jointly do many of the tasks listed above. In some cases, no one in the organization is responsible for the duties of the system administrator. This is due in part to the transition from single user to multi-user systems. Many organizations are simply not aware of what needs to be done for a successful multi-user (UNIX) computer installation. The "PC mentality" holds where people think a system like the Fortune is as simple to operate and maintain as a personal computer. The truth is that while you get increased power and flexibility from a multi-user Fortune system, it also takes more to maintain it. It is amazing that someone is willing to expend thousands of dollars to purchase a computer system, yet they don't want to spend the money to properly maintain it. Therefore, for those of you that have not offically made someone the system administrator, you should. Depending on the size and number of systems you have, this will probably be only a part time

insures your computer operation will run smoothly in the absence of the regular administrator. By assigning these responsibilities and insuring the person is properly trained for the job, you reduce the chances of computer down time and can increase the productivity of your system.

We might also add here that many companies opt to use an outside consultant to perform many of the functions of the system administrator. Even if you have a system administrator, the outside consultant is used to solve problems that the administrator cannot resolve. They can also be used to train your administrator. In many cases, the money spent on the time it takes to train someone is less than having to call someone everytime a problem arises.

As a minimum, the system administrator should know the basic commands and concepts of **FOR:PRO**. This normally requires some sort of UNIX training course. It saves time and money in the long run if the administrator can communicate in general UNIX terms with a consultant for problems they cannot resolve. (i.e. there's a difference between having to tell someone to change directories to /usr/bin and list the files there and having to tell them to type in "cd /usr/bin" and hit the return key, then type in "ls" and hit the return key. In the latter case, you usually have to enunciate and spell each letter).

Our goal is to help you understand the files, commands, and concepts that a system administrator needs to know in order to resolve problems and to keep your system running properly.

Enough about stressing the importance of a system administrator. Before we get into the heart of the material we will be discussing, let's talk a little about each of the responsibilities outlined above.

As you already know, each user on the system must be assigned a login account. This login account should also be deleted when it is no longer needed. There are selections on the Fortune menu system for doing this. In this series of articles, you will learn how to do this without the help of the menu system. There are also good and bad ways to set up these accounts. We will show you *good* ways. The "manager", "root" and "newuser" accounts require some special attention.

System backup is the **most** important duty of the administrator. You don't know how many times we have gone in to fix a problem for someone and the task is complicated by the fact that a backup has not been done for months.

We are the *largest* Authorized Fortune Master Dealer in the Pennsylvania and Delaware area. As part of the **P**rofessional **S**olutions **G**roup of companies with over 35 years in business automation, we have the *stability* that our customers demand

Our services include a factory trained field engineering team, a complete parts inventory including system upgrade capability and a software support department.

For all of your Fortune needs contact:

Mark Shields of PSG at
2550 Eisenhower Avenue
PO Box 762
Valley Forge, PA 19482

or you can call Mark at (215) 666-1400

In the case of a hard disk that has gone bad, this means re-entering all of the data since that last backup. Even if a backup has not been done for a few days, this means lost time and revenue for your company. We will present some suggestions and hints for a good backup procedure.

If the system has a problem that requires a complete system restoration (such as losing a hard disk), you can save time and money by having someone who knows how to restore the system once the problem has been fixed. Occasionally, user files need to be restored (such as a Fortune:Word document that someone accidently deletes). We will talk about some practices and procedures for doing these tasks.

There are both hardware and software changes that have to be made when printers, terminals, modems, etc. have to be added or changed on the system. We will discuss some of the things that should be considered when doing this. There are also some special considerations for application programs such as BASIC/IDOL/BAS. Remember, the more you learn from these articles that you can do yourself, the more money you will save in outside costs. On the other hand, we will also talk about how some things are best left for someone else to do. Part of the system administrator's job to know *when* to do it and when to call for help (e.g. setting up printers).

In many organizations, the system administrator gets the job by default

For some companies, system integrity and security is not an issue. This is true if everyone can have access to all parts of the system. For most, however, they want to be sure that some people do not have access to some parts of the system. For those that have modems, security is even more important since people from outside the company may be able to access your data. The administrator is responsible for knowing how to set the system up to insure system integrity. We will talk about some of things you can do to insure that your system is secure.

Occasionally, you will purchase new software for your system. Sometimes the person you buy it from will install it for you or tell you how to install it. Sometimes it can be installed from the Global Menu and sometimes it can not. In any case, the administrator is responsible for insuring that it is installed properly either through coordinating the efforts of others or installing it themselves. This will also be discussed in these articles.

With the installation of a computer comes a demand for terminals, printers, software, etc. The system administrator is normally responsible for coordinating the use of the system and its peripherals. They make decisions like who should have a terminal or printer, and what software packages are needed by users to perform their jobs most efficiently. This may involve

Affordable laser printing for Fortune

Laser Printing! Kiss the expense good-bye.

QMS — KISS laser printer $1995.00

- cost per page 1.4¢.
- 9 resident fonts
- Print cap and print utility included.
- add-on sheet feeder provides total of 3 trays and envelope feeder.

THE GOOD FORTUNE TRADER

QMS - KISS from
The Good Fortune Trader
1 Year Warranty

a division of Computer Services of Atlanta
(404) 392-9938

1534 Dunwoody Village Parkway, Suite 109, Atlanta, Georgia 30338

DO YOU KNOW...

there is an organization that has a thorough knowledge of

FORTUNE BAS
MULTIPLAN IDOL

We have been providing widespread support to our Fortune clients for more than 4 years.

If you need:

- training
- support
- maximum computer performance
- peace of mind relating to processing
- customization
- application results
- minimum errors and confusion
- better ideas on how to increase the power of your Fortune

Call Steve Rosenfeld at (314) 837-7111

SUPERIOR COMPUTER SYSTEMS, INC.
Making computers work for you

917 N Hwy 67, Suite 105
P.O. Drawer T
Florissant, MO 63031

cost/benefit analysis to justify the purchase of new hardware or software or may be a simple decision like where to place a terminal or printer. We will talk about these types of decisions in more detail.

After the system is installed, there many be additional requirements for the system. Do I need power protection? Do I need a larger disk or more memory? Do I need to add more communication ports for additional terminals? Do I need a tape streamer backup instead of a flexible disk backup? Do I need a larger system? As a minimum, the administrator must coordinate and communicate these requirements through their dealers. They should not only be educated in what their system is doing, but what its potential is. Later, we will discuss what some of these terms mean and give you some basic system guidelines.

A major part of any successful computer installation is training. This not only includes training for the system administrator, but also includes user training such as Fortune:Word, Multiplan and Accounting. The administrator normally helps determine and coordinate this training. We will give you some guidelines on training.

The major expense of a computer installation is not the hardware and canned software. Over the life of the computer, it is the dollars expended for internal or external software maintenance and development. Because of this, many factors should be considered. This includes an initial decision of whether to develop and maintain software within the company or not. If outside software maintenance and programming services are used, some factors should be considered to insure you get the "most bang for your bucks". We will talk about these considerations.

After software and hardware are installed, upgrades will be required (you can count on it!). These may be upgrades to the operating system (**FOR:PRO**), application software (BAS, Fortune:Word, Multiplan, etc) or to the system itself (hardware fixes such as new WD controllers, motherboards, power supplies, etc). The administrator is responsible for coordination to insure the most current hardware and software are installed on the system.

Whether your system has a maintenance contract or is repaired based on time and materials, the administrator is responsible for knowing who to contact if hardware problems arise. The administrator should also know the major benefits of the different types of contracts available. We will discuss some of the contract types that are available and the advantages/disadvantages of each.

As you can see, the *system administrator* has a lot to be concerned about. Our goal in this series of articles is to try to pass on as much information as possible to make the job easier. This information will come from a variety of sources - hardware manuals, software manuals, tech tips, field service notices, and most importantly - experience.

Before we leave this session, let's bring up our first security issue. There are three login accounts on your system that should, without a doubt, be password protected. In fact, all three of them should normally have the *same* password. They are **manager**, **root** and **newuser**. The manager and root accounts in **FOR:PRO** have unlimited power and access to the system. If they are not protected, normal users can login and remove, view, etc. any system file or data file. The system administrator normally logs into the system using one of these two logins to make changes or fix system problems. Since we will be giving you information that can be dangerous if misused by people not authorized to login using these account names, we suggest you password protect them right now using the **S2** selection on the Global menu.

The newuser login account is used to add new users to the system. Since this is a duty of the system administrator, he should be the only one that has access to it. If it is not password protected, any user who uses it can create a login account and get into the system even though they are not authorized. We suggest you assign the same password to the manager, root and newuser accounts. The password for these three accounts should be changed occasionally to insure they have not been compromised.

In the next issue, we begin to discuss "system administration" by going through the **FOR:PRO** (UNIX) directories and discussing what files and commands are contained in each of the directories. Half of the solution to a problem is to know where to go to fix it. In subsequent issues or where it is applicable, we will be discussing a variety of subjects which are important to the system administrator including some of those mentioned above.

It is amazing that someone is willing to expend thousands of dollars to purchase a computer system, yet they don't want to spend the money to properly maintain it. A well-trained system administrator is an investment in your computer.

By the way, if you have any comments or suggestions for this series or articles (or the previous "Introduction to Shell Script" series), you can write to:

Uni-Komp
9803 Rocktree
Houston, Texas 77040

*****

*** See the NEW product DOS LINK ***

SOFTWARE FOR THE FORTUNE 32:16

NEW PRODUCT!!! DOS LINK ($495.00) - provides easy-to-use menus and programs that transfer data between any PC (including Laptops) running DOS and the Fortune system. Here are just a few examples of how this product can be used:

- Transfer Multiplan or Fortune:Word Documents from or to your DOS system.
- Transfer other ASCII or binary files.
- Use the PC or Laptop as a remote terminal on the Fortune system.
- Transfer THOROUGHBRED (SMC) BASIC data that has been extracted using one of our other products

Documentation and software for both the DOS and Fortune system are provided. Data may be transferred via a direct connection (9600 baud) between the two computers or via modem (at modem speed). You may also order additional hardware for your PC that may be required to accomodate the transfer.

If you are interested in purchasing a Laptop computer that can be used with this product for people on the road or at home, give us a call!! We also provide consulting services for interfacing your DOS system to the Fortune ($50/hr).

MULTIPLAN LINK - can pass data from BAS/IDOL/BASIC files to Multiplan and back. For example, it could be used to send data from any of the BAS files (Chart of Accounts, Customer Master, Inventory Master, etc). Allows selection of fields to be sent and provides logical retrieval and key range selection. **$295.00**

RECORDS PROCESSING LINK - can pass data from BAS/IDOL/BASIC files to Records Processing and back. For example, it could be used to take customer address data from the Customer Master and will automatically make the Records Processing List document for mail/merge. Allows selection of fields to be sent and provides logical retrieval and key range selection. $295.00

ASCII LINK - can pass data from BAS/IDOL/BASIC files to ASCII and back. Use to send data between machines or databases. For example, it could be used to send data from any of the BAS files (Customer Master, Inventory Master, etc). Allows selection of fields to be send and provides logical retrieval and key range selection. **$295.00**

TERMINAL/PRINTER LINK - provides capability to print data on a printer connected to a workstation. Can print BAS/IDOL/BASIC reports, Multiplan and Fortune:Word documents. BAS/IDOL reports can also be viewed before printing, converted to Fortune:Word format; accumulated for mass printing; backed up to diskette; printed on any system printer. An option to print BAS/IDOL reports on a LASER PRINTER is $50 extra. Many other options are included. **$195.00** (Fortune:Word printing on a printer connected to a workstation is $50 extra)

TELEPHONE LINK - on line telephone message, telephone directory and inter-office mailbox/menu system. Telephone message entry resembles typical pink slip. Telephone directory can search by name or company. Mailbox/message system can send and receive Fortune:Word documents or Multiplan documents. Does much more! Give us a call. Written in Business Basic. **$295.00**

CALCULATION LINK - provides programs for amortization, depreciation, loan repayment, averaging (with graph), linear correlation and breakeven analysis. Written in Business Basic. **$95.00**

KOMPACT PERSONNEL ACCOUNTING - provides for data capture of personal, wage/salary, job, education, salary/wage, appraisal training, dental, medical and life insurance data. Includes over 80 pre-defined reports/worksheets. Written in Business Basic/IDOL. **$995.00**

PRINTER LINK - provides printcaps for Okidata 84, 2410, TI 855, Anadex 6500 and Printronix. Includes shell script for changing pitch and LPI. Allows these printers to be used in Fortune:Word. **$95.00**

MODEM LINK - allows Hayes 1200/2400 modem to be used on one port for both incoming and outgoing calls. **$95.00**

WESTERN LINK - provides instruction on how to use your Fortune for sending/receiving telex messages using Western Union's Easylink. Can replace your telex! Fortune ITE or Handshake software required. **$95.00**

UNI-KOMP 9803 Rocktree Houston, Texas 77040 (713) 937-7782

UNIX Basics 6

Special Characters

This month we discuss the topic of *special characters*. Special characters are those which are interpreted by the UNIX shell in a special way. Knowing what these special characters are and how to use them will provide you with two main benefits. First, you will understand more completely how the UNIX shell processes commands. Second, you will be able to use the shell more effectively and efficiently.

In last month's article on redirection and pipes we discussed several special characters. As you will recall, the greater-than sign (**>**) and the less-than sign (**<**) are special characters that are used for redirection. Whenever the UNIX shell encounters these characters anywhere on a command line (unless they are *escaped*, something we will speak of later), UNIX will try to interpret them as redirection signals. Last month we also introduced the vertical bar (|) which is the pipe symbol. So, three special characters are **<**, **>**, and |.

We will discuss in this article several more special characters including the ampersand (**&**), the semi-colon (**;**), the asterisk (*****), the question-mark (**?**), the square-brackets (**[]**), and the back-slash (\).

Let's begin with a general caveat. Special characters are just that - *special*. They are interpreted by the UNIX shell in a special manner. This means that you should **not** use them in file names. For example, if you wanted to get a list of the users currently on the system, and you liked file names with punctuation in them, you might be tempted to type something like this:

```
$ who > who_is_there?
```

If you tried this, the shell would respond with:

```
who_is_there?: No match.
```

The shell is basically telling you that it couldn't find the file called who_is_there? The reason this is a problem is because we tried to include a question-mark as part of the file name. (As an aside, the use of the under-line character caused no problems. In fact, since spaces are not allowed in filenames, many people include the underline character to produce somewhat more readable names.)

The question-mark is a special character and should not be used as part of a file name. This also goes for any other special character we will discuss in this article. Thus, as a general rule, do not use special characters in the names of files.

The Ampersand (&)

The ampersand character is used "to put UNIX commands in the background." *In the background*, what does that mean? You know that when you type a UNIX command to your prompt, UNIX will start that command running. When the command is finished, UNIX will redisplay the prompt character, which is UNIX's way of saying, "Ok, I'm ready for another command." The important thing to note here is that UNIX will not redisplay the prompt character until the command is **finished**. The ampersand, however, when appended to the end of your command line, tells UNIX to start that command running and then *immediately* redisplay the prompt character so that you can enter more commands. Thus, the command with the ampersand is *put in the background*, that is, it is running in the background of the computer, sort of silently out of sight. At the same time, you can be running more commands in the *foreground* which is roughly synonymous for running commands "interactively".

The real benefit for the ampersand, or running commands in the background, is when a particular command will take a long time to complete. In a case like this, it really doesn't make much sense to start that command running "interactively" and then tie up the terminal for 5 or 10 minutes (unless, of course, it is time for a coffee break).

See **Basics**, page 11

VITAL RESOURCES

YOUR GUIDE TO AUTHORIZED FORTUNE SYSTEMS MASTER DEALERS

At Micro Resources--Ohio's Master Dealer of Fortune Systems products--we value the network established to link the manufacturer with you, the end user. We believe that--quite simply--your most cost-effective and reliable service comes from the Master Dealer/Affiliate Dealer network.

So, to help you find the most service oriented resource, Micro Resources is sponsoring this listing of every authorized Master Dealer in the United States.* Because, to be honest, the stronger the Fortune Systems foundation is nationwide, the stronger we are.

AUTHORIZED MASTER DEALERS AND VARS

ADVANCED INDUSTRY SYSTEMS, INC.
9045 Seminole Blvd.
Seminole, FL 33542
813-397-3535

APEX BUSINESS COMPUTERS, INC.
210 O'Keef Avenue Suite 400
New Orleans, LA 70112
504-524-0648

ATEK COMPUTER DISTRIBUTORS INC.
3017 Kilgore Road, Suite 180
Rancho Cordova, CA 95670
916-638-4744

AUTOMATED OFFICE SYSTEMS
2228 El Cajon Blvd.
San Diego, CA 92104
619-542-0144

BOSTON BUSINESS COMPUTERS, INC.
72 River Park Suite 1
Needham Heights, MA 02194
617-449-6980

CTS COMPUTER SYSTEMS
405 114th Ave. S.E., Suite 337
Bellevue, WA 98004
206-453-5411

COMPUTER PARTNERS INC.
2995 N. Cole Road, Suite 260
Boise, ID 83704
208-377-2070

COMPUTER SALES CO. INC.
14111 E. Freeway Drive
Santa Fe Springs, CA 90670
213-921-8825

COMPUTER TRADE DEVELOPMENT
30600 Telegraph Road, Suite 1386
Birmingham, MI 48010
313-540-1282 Bermingham tech line
313-271-3020 Dearborn office
313-567-3331 Bermingham office

DYNAMIC BUSINESS SOLUTIONS, INC.
4744 N.W. 167th Street
Miami, FL 33014
305-625-6012

GENERAL INFORMATION SYSTEMS, INC.
1200 Bigley Avenue
Charleston, WV 25302
304-342-7108

J.F.M. BUSINESS SYSTEMS
144 Merchandise Mart Plaza
Chicago, IL 60654
312-644-0870

J.J.J. INC.
834 Tyvola Road
Charlotte, NC 28210
704-525-0582

MICRO RESOURCES, INC.
255 Bradenton Avenue
Dublin, OH 43017
614-766-2335

NEW JERSEY BUSINESS COMPUTERS, INC.
301 Route 17 North
Rutherford, NJ 07070
201-438-1410

PROFIT MASTERS, INC.
P.O. Box 2799
231 E. Ruby Street, Suite E
Kissimmee, FL 32742
305-933-1525

PSG COMPUTER CENTERS
2550 Eisenhower Avenue
P.O. Box 762
Valley Forge, PA 19482
215-666-1400

QBC SYSTEMS, INC.
375 North French Road
North Tonawanda, NY 14120
716-691-5201

STRATEGIC NETWORKS, INC.
360 Bloomfield Ave.
Windsor, CT 06095
203-683-1070

SUMNER GROUP
2121 Hampton Avenue
St. Louis, MO 63139
314-647-2500

SYSTEMS ASSOCIATES, INC.
8118 Philadelphia Road
Baltimore, MD 21237
301-866-8900

VARS

ASI COMPUTERS
63 West 38th Street
New York, NY 10018
212-719-2919

BUSINESS EQUIPMENT CENTER
2991 Directors Row
Memphis, TN 38131
901-345-5650

INTEGRATED BUSINESS SYSTEMS
2200 E. River Road, Suite 120
Tucson, AZ 85718
602-299-7660

KNOWLEDGE ENGINEERING CORP.
8597 Harborview Drive
Blaine, WA 98230
206-371-2031

P & ID
1700 Enterprise Way, Suite 110
Marietta, GA 30067
404-952-1572

SOFTWARE ENGINEERING ASSOCIATES
P.O. Box 771
301 E. Greenville Street
Anderson, SC 29621
803-225-0101

THE LOGIC WORKS
10009 Blue Coat Drive
Fairfax, VA 22030
703-591-2345

UNISOFT, INC.
14100 East Jewell Ave. Suite 27
Aurora, CO 80012
303-671-8600

So, if you have a product question, need an upgrade, or are looking for a reliable service arrangement, call a Fortune Systems Authorized Master Dealer. If you are unable to find a resource to meet your needs, call Micro Resources. Our dedication to Fortune Systems and to complete service is simply unsurpassed.

Micro Resources

255 Bradenton Avenue, Dublin, Ohio 43017

FORTUNE SYSTEMS
Authorized Fortune Systems Master Dealer

*This is the most current listing as presented to Micro Resources by Fortune Systems on June 2, 1987.

Basics

Continued from Page 9

As an example, then, let's say we wanted to copy all the files from some directory onto our floppy disk (for this example, we will assume that the floppy drive is already mounted). If this is a large directory, then it could take several minutes for the copy to take place. We could put this command into the background as follows:

```
$ cp /u/john/* /f  &
801
```

Let's look at what we just did. First, the **cp** command should be understandable. We are copying the contents of the directory, **/u/john**, to the floppy drive (**/f**). The ampersand (**&**) on the end of the line instructs UNIX to put this command into the background. The **801** is the *process number* of this particular command. You, as the user, would not type this in. Rather, the system prints this out whenever it sees the ampersand. The process number, or *process id* (sometimes abbreviated **PID**) can be very useful if you want to track the progress of some process. This is especially handy if the job you put in the background will take a very long time and you are interested in knowing when it is finished. You can periodically check on its progress with the **ps** command. For example, if after a short period of time you were interested in seeing if our **cp** command was complete, you could type:

```
$ ps 801
```

The output would look something like if the command were still running:

```
  PID TTY   TIME CMD
  801 p01   1:06 cp ...
```

Otherwise, just the line that starts with PID would be printed out and then you would see your prompt character. In any case, if you want to keep track of when your job is finished, it will be necessary to somehow remember the process number that UNIX prints out when you use the ampersand.

The Semi-colon (;)

The purpose of the semi-colon is to separate different UNIX commands so that they can all be typed on one line. Let's return to our previous example of copying the files from some directory to the floppy drive. Recall that we assumed that the floppy drive was already mounted. In the following example we will not assume that. So, to copy files from some directory to the floppy drive would require the following commands:

```
$ mount /dev/fd02 /f
$ cp /u/mary/* /f
$ umount /dev/fd02
```

The semi-colon allows us to type more than one command per line as in:

```
$ mount /dev/fd02 /f; cp /u...
```

You might ask, if the semi-colon is only used to allow one to type multiple commands on the same line, then what real use is it? After all, I could just type each command on a line, couldn't I?

Thus, as a general rule, do not use special characters in the names of files.

Well, there might be some occasions when knowing about semi-colons could come in handy. First, you will undoubtedly run across semi-colons in the way other people use UNIX (perhaps you will read an article in some other journal where they are used), so it won't hurt to know something about them. Second, there are some cases where they simplify one's life. For example, suppose you need to run two separate programs and you want to capture the output in a single file. Let's also suppose that the name of the first program is **p1** and the name of the second is **p2.** We could get all the output from these two programs into a file called p.out by doing the following:

```
$ p1 > p.out
$ p2 >> p.out
```

Or, we could type:

```
$ (p1;p2) > p.out
```

So, we've somewhat simplified the amount of information that we type in. Also note that the parentheses are necessary in the command which contained the semi-colon. For example, the following command

```
$ p1;p2 > p.out
```

would not put the output of both commands into the file p.out but rather would run p1 whose output would go to the terminal and then run p2 and redirect the output from just p2 to the file called p.out.

Wildcards

Several special characters act as wildcards. We've touched upon one of these wildcard characters, the asterisk, in an earlier installment of this Basic UNIX series (see Volume 4, Number 1). We will now discuss this character along with the question-mark (**?**) and the square brackets (**[]**). Each of these characters are used as wildcards. Those of our readers who have even a vague knowledge of the card game called poker will be familiar with the notion of a wildcard. A wildcard is a card, usually designated by the dealer, which can be used as any card that the owner of the wildcard wants. In the same way, UNIX recognizes wildcards, which can take the place of any character.

We can start with the asterisk (*). The asterisk matches any character and also any number of characters. For example, if you want to see a directory listing of all the files in your directory, you would type:

```
$ ls *
```

The asterisk in the above command literally matches all the file names in the current directory. (Technically, this is not true. The asterisk used in the above command matches all files except those that begin with a period (.). In order to get a listing of the files that begin with a period, you would type: `ls .*`). One point to make is that the **ls** command never sees the asterisk! The UNIX operating system takes

the command you typed in and then looks at it. Before UNIX ever starts the **ls** command running, it sees the asterisk and then *substitutes* for the asterisk all the names in the current directory. After it does this substitution it starts the **ls** command running which no longer has an asterisk to look at, but a long list of filenames that it will process.

To further understand this point, let's suppose that we are in the directory called **/u/fred**. In this directory, let's further suppose, there are four files called **chap8**, **chap9**, **chap10** and **notes**. Given this, the following two commands will give you the exact same output:

```
$ ls *

$ ls chap10 chap8 chap9 notes
```

Though, I think you will admit, the first is much easier to type. What is interesting is that the ls command, *in both cases*, sees only the second command. This is because the UNIX operating system *expands* the asterisk into, in this case, `chap8 chap9 chap10 notes`.

There is a UNIX command that will expand our wildcards so that we can see how UNIX will interpret an particular wildcard. This UNIX command is called **echo**. For example, if we are in /u/fred and we type:

```
$ echo *
```

we will see the following:

```
chap10 chap8 chap9 notes
```

The **echo** command is sometimes useful for testing whether or not a wildcard construction will have the effect that we desire. You may wish to learn about its use so that you can check whether or not a wildcard construction that you come up with will do what you think it will do.

Returning to our discussion of the asterisk, we should note that the asterisk does not have to be used alone. For example, look at the effect of the following command:

```
$ ls chap*
chap10 chap8 chap9
```

In this case, **chap*** matches all the filenames in the directory that begin with the four letters **chap**. The asterisk can be used anywhere in a filename and will match one or more characters. In the above example, the asterisk matched the single character **8** in the file called **chap8**. It also matched the two characters **10** in the file called **chap10**. If we had a file in that directory called **chapter**, then **chap*** would have matched it, too.

If we want to match just one character, then we use the question-mark. For example:

```
$ ls chap?
chap8 chap9
```

Note the difference between how the asterisk and the question-mark works. **chap*** matched **chap10** but **chap?** didn't. The question-mark will match *one* and *only one* character. However, what would **chap??** do? It would match **chap10** and *not* **chap8** or **chap9**.

Let's continue. The square brackets are used to match a *class* of characters. In other words, any characters within the brackets are matched. It is like the question-mark in that only one character is matched. For example,

```
$ ls chap[678]
chap8
```

The above command which includes the **[678]** would match **chap6**, **chap7** or **chap8** but since our imaginary directory does not include files called **chap6** or **chap7**, only **chap8** is printed out. A dash (-) can be used to specify a *range* of characters. As an example, `ls chap[6-8]` would have the same effect as `ls chap[678]`. As another example, `ls [a-z]*` would match all files in your directory that began with a *lower* case letter.

Escaping Special Characters

Sometimes it is necessary to use one of the characters that the shell thinks is special in a non-special way. Thus, we need a way of *signaling* the shell to treat this character as a normal character. There are two basic ways of accomplishing this. By the way, this process of telling the shell to ignore the specialness of a character is called *escaping* that character.

The first way involves the use of the backslash (\). Any character preceded by the backslash is not treated in a special way. Thus, it is possible to create a filename that includes a special character. For example, if we want to see the contents of a file called **what?**, we could do the following:

```
$ cat what\?
```

The second way of escaping special characters is to surround them with single quotes. So, we could type:

```
$ cat what'?'
```

in order to see the contents of the file called **what?**. The shell will not try to expand anything that is between single quotes. Thus, it is possible (though not advisable) to create filenames that look very strange. As an obviously inane example, the following command:

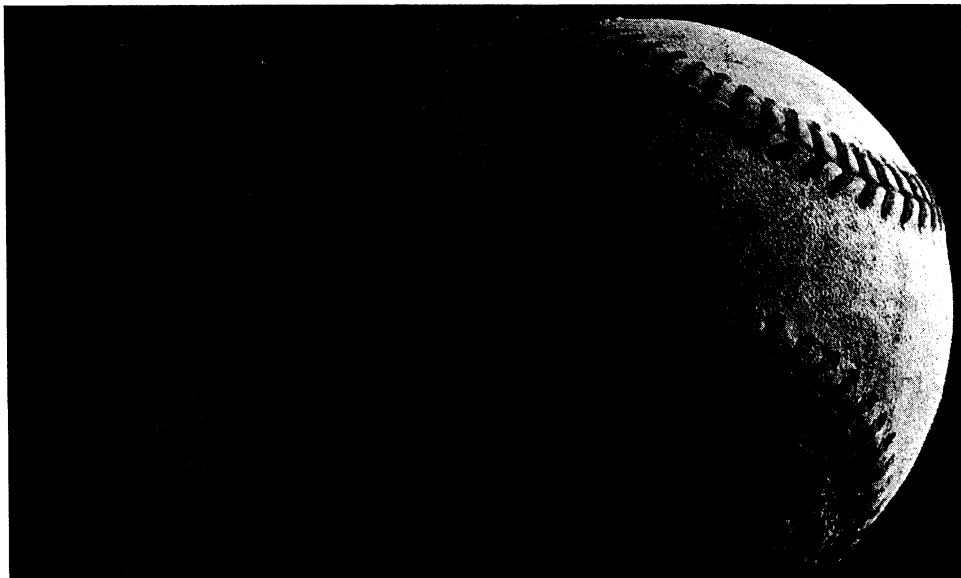
```
$ cat '&;*?[]\'
```

would instruct UNIX to try and type out the contents of the file called &;*?[]\. But please, don't ever name a file like that!

Conclusion

In this article, we've discussed the notion of special characters in UNIX. Certain characters are interpreted in a special way by the UNIX operating system. Knowing what they are and how to use them will allow you to use your Fortune more effectively. Please write or call us if you have any questions about this topic. Also, let us know you like this series of articles and whether or not you would like to see any other topics covered. We appreciate your input! □

Mark Palmerino



IF YOU WANT TO WIN WE'RE READY TO PLAY!

You may have seen our ads. You may have heard about us. But you haven't heard anything yet. Because we're getting ready to leave the competition behind. We're **Tri-Tech Computer Services**–part of the Network of Fortune Authorized Affiliate Dealers whose personnel are Fortune trained and have been servicing Fortune Computer products for several years.

In the late summer of 1986 we began marketing Fortune computer equipment and our expertise supporting them. Today, we're in the big league of players with Fortune Systems products and services.

Because we are a service organization first, we use our technical expertise and resources for add-ons to and upgrading of the products we support. We do this better than the vendor whose primary purpose is to sell you a product. With **Tri-Tech**, you don't ever get the "pushing" typically associated with sales.

With offices in Connecticut and Massachusetts and clients on both coasts and in-between, we support the complete line of Fortune's multi-user computer equipment. We have the people, the expertise and the resources to meet your computer needs today, and tomorrow–with superior local as well as nationwide service and support.

We're going for it! We are challenging the competition for the No. 1 spot. If you need Fortune Computer equipment or services, consider the company that many corporations rely on.

Tri-Tech Computer Services, Inc.

If you want to win, we're ready to play.

For the Tri-Tech representative nearest you, call (203) 828-9784 or write Tri-Tech Computer Services, Inc. 106 Cynthia Drive, Kensington, CT 06037. Associate Dealer inquiries invited.

Destined to be No. 1!

Kensington, CT (203) 828-9784 Newton, MA (617) 965-4453

The BASIC Advisor

Ray Wannall is President of BaSiC Software Corporation in Baltimore, MD, and is contributing independently to this publication.

Question: We use our 32:16 for billing and business accounting only. We have General Ledger, Accounts Receivable, Order Processing and Purchase Orders. I have learned about using the Fortune System primarily on my own through trial and error with many evening hours during the first year we had the system.

Concerning Order Processing/Inventory Control (?), there have been many occasions where I pitied anyone with a large inventory and trying to use this system, especially when I would discover a bug by accident. For example, our inventory items are on a FIFO basis. Using the Debit/Credit Memo selection on OP, I issued a credit by-passing the invy. qty. field because I did not want the item back in stock. The inventory item master record quantity on hand did not increase; the qty. returned field stayed the same; BUT the LIFO/FIFO record increased by one!!! Surprise, surprise! So I subsequently decided to use the Debit Memo for re-billing at the correct price of the item, again by-passing the invy. qty. field. Lo and behold! Sure enough the quantity field in the FIFO record was reduced. For all intents and purposes, in no way should any inventory record be affected when by-passing the invy. qty. field during credit or debit memo entry; nor when the memo is printed; nor when the sales journal is printed. (Sales journal printing does somehow tie in with sales analysis reports which we do need in order to get our sales category breakdown.)

I feel I have to back-track here and re-organize my thinking. I printed the credit memo, the sales journal, the inventory movement report, sales analysis report which showed the quantity I sold but no dollar amount, which would have been correct because I did not put the item back in stock, but I did clear out the dollar amount of the billing. I then checked the inventory item master record and the LIFO/FIFO record and found the discrepancy in the qty. fields. Something is eluding me here. I do know that sales journal printing and approving it as correct is what ties in to the LIFO/FIFO records of inventory items and posts these dollar amounts to the general ledger inventory and cost accounts.

Our inventory items are on a FIFO basis. We seem to get an error in the LIFO/FIFO record which increases when it shouldn't.

Here it is step-by-step. During Direct Sales Entry, inventory quantity on hand in the Item Master Record is immediately reduced, but NOT in the LIFO/FIFO Record. After the sales journal is printed the LIFO/FIFO record is then affected. These are the cost figures that are posted to the General Ledger inventory and cost accounts. (The cost dollars on the sales analysis report never tie-in with the sales journal postings to the general ledger accounts.)

I think it best that I stop here because I feel I am going to go off on a tangent and whoever reads this is going to feel the way I did when I tried going through the instruction manuals: "what are they saying?"

Check out the following: Inventory Item Master Field No. 21, Cost Method Ind. Problems are caused when using LIFO or FIFO, i.e., field no. 8 in the Item Master, Accounting Cost, is averaged based on the number of lots in the LIFO/FIFO Lots File and the quantity and dollar amounts in each lot.

How is the LIFO/FIFO Lots file tied in during Debit/Credit Memo entry and/or printing? Credit memos are issued for defective merchandise which is naturally not added back into our inventory stock for resale. Because of the current problem the cost accounting field is way out of line based on what I mentioned above when using LIFO or FIFO as the Cost Method Indicator.

There will be more later.

Answer: Thinking back over the last decade or so, I can recall coming across only one installation of hundreds which actually used LIFO/FIFO inventory accounting. (I admit, however, that my vertical concentrations have not usually included Inventory Control as a primary application.) If you are, in fact, a member of a small minority group, it is no wonder there has been no grease applied to your wheel. I'll bet there **is** more!

You included in your letter the FOR:PRO release on your system, but did not mention the versions of the accounting modules. I assume from the age of your Fortune that you are in one of the pre-2.0 Order Processing packages (as am I). If this is the case, I'm afraid you are mistaken about the sales journal printing affecting the LIFO/FIFO records. That file, called CINV3, is never opened by the sales journal print/update programs, so quantities cannot be adjusted there. Excluding the Purchase Order, Physical Inventory and O/P File Mainte-

nance systems, the only time the LIFO/FIFO records are altered is during Inventory Adjustments Entry and Journal, and Invoice/Memo Printing. We know from your diligent efforts that the Invoice/Memo Printing has to be the culprit.

The program which updates inventory quantities as well as LIFO/FIFO Lots is named CBIUA2. This update occurs just after you answer affirmatively to the question "INVOICES CORRECT (Y/N)": memos and invoices are printed and updated by the same program. Whoever wrote this gem went to great pains to insure that quantities were handled properly in the Inventory Item Master when memos were being processed. But there is no acknowledgment of memos in the LIFO/FIFO Lots File update portion of the program. And it is the memo quantity, not the invy. qty. (as you called it), that updates LIFO/FIFO Lots.! Who would have guessed?

You can fix this "bug" yourself by adding a single line of code to CBIUA2, and I encourage you to do so. To be completely safe, do an "END OF DAY" backup of your program files before you conduct the edit. If something goes wrong, you can drop back ten and punt with a "START OF DAY" restore of program files.

Follow this procedure, remembering to use caps and that <CR> means "press the RETURN key": From any BAS accounting selector type

```
BASIC <CR>
```

At the > prompt type

```
LOAD "CBIUA2" <CR>
```

Then type

```
LIST 7520 <CR>
```

you should get another prompt. (If you do not, type RUN "DOL" <CR> and call for assistance.)

At the prompt type the following command:

```
7520 IF A1$(2,1)="C" OR
A1$(2,1)="D" THEN LET C=A2
<CR>
```

(The above should be all typed on one line.) If you get an error message you probably typed the command wrong. Don't panic, no damage has been done yet. Simply retype the command, being sure it is exactly as it appears above and press the RETURN. If the machine persists in giving you errors, type `RUN "DOL" <CR>` and find help. Otherwise:

At the prompt type

```
SAVE <CR>
```

Then type

```
RUN "DOL" <CR>
```

As they say in the Fortune Technical Tips, "The procedure is complete". Happy computing, and I hope to hear from you again.

Continued on next page

G.T. Hawes & Co., Inc is BACK!!!

with prices that have never been:*cheap, cheap, cheaper!!!*
and delivery that is: *fast, fast , faster!!!*

G.T. Hawes & Co., Inc.

(No Longer an Authorized Fortune Master Dealer)
(No Longer an Authorized Fortune VAR)

G.T. Hawes & Co., Inc.

475 West Merrick Road
Valley Stream, NY 11580

(516) 627-5556 Call George Today (516) 568-0044

Question: I have some follow-up for you concerning error 103, which you discussed briefly in the March issue of /u/fortune news (volume 4, number 3). I am not familiar with Fortune:Windows but have had 103's from time to time. I have found that they occur most frequently when either ipl file ownership and read-write permissions are not set properly or devices in /dev have lost their character status or read-write availability for all users. All tty's in /dev should have the following attributes:

```
crw-rw-rw-  root
```

Or better, the owner needs to be root or bin. If the system has been back loaded rather than installed, the owner may be different. Also, be sure in ipl files (is that 1401 or 360 initial program load carry over?) that the terminals are linked properly and the owner is correct.

Also, I imagine that you could get an error 103 if you ran out of swap space, or something like that for which UNIX would issue an error that BASIC would capture and pass to the user. In essence, BASIC must trap most UNIX errors and convert to error 103. One time a bad sector in the keys area of a file gave me a 103.

Keep up the good work! - Mike Smith, Athens, TN

Answer: Sometimes I feel so silly when I call these things "Questions" and "Answers". Thanks once again, Mike, for your input.

*****

We, at **/u/fortune** *news*, appreciate the fine job Ray Wannall is doing with The BASIC Advisor column. Please keep the questions and comments coming–Ray can only answer your question if he knows about it. You can write directly to him at BaSiC Software, 5201 Powhatan Street, Baltimore, MD 21207, or call him at 301 448-9460. □

Vantine Business Systems

- System Sales
- Consulting
- Support
- Onsite and National Depot Service

Over Three and half years supporting Fortune Users

Vantine Business Systems
70 Walnut Street, Wellesley, MA 02181
(617) 239-8084

Authorized Fortune Systems Dealer

/u/help

A forum for our readers' questions and comments.

Dear /u/fortune news,

You suggested that if any of your readers felt that you skipped a particularly useful option for the **ls** or **cp** command that we should drop you a not and let you know.

I believe that perhaps one of the most useful options which, as far as I know, remains undocumented, is the **X** option. If you use this option on a backup command, it will verify the copy and make certain that it is readable.

I came across this option several years back when we were having a great deal of trouble with hard disks and, on one occasion, after the disk had crashed, discovered that our backup disks could not be used. Needless to say, I was more aggravated than usual at the system failure and the person sent by Fortune suggested that we, in the future, use the **X** option in order to verify the backup. Since beginning the use of that option, we have never had a problem with our backup disks.

While it takes a little bit of extra time to use the verification option while doing the backup, it only takes one time of discovering that your backups are not readable to repay the time many fold.

Sincerely,

Thomas L. Grisham

Editor's Note: Thank you, Mr. Grisham, for bringing up a very useful option for the **cp** command. According to the FOR:PRO manual, the **-X** option "... verifies that the backup media is readable after each disk is written. Does not check file system integrity or do any data verification, however, This option is ignored when not used with -B."

This option obviously provides a cer-

tain amount of protection which, as Mr. Grisham notes, is well worth the added time it takes to do this check. However, our readers should realize that this option does not provide *complete* protection. It would be possible to have a disk that is readable but still have problems with the copy. In fact, we recently ran into this very problem.

We discovered that some of our customers who purchased our software diskettes were having some problems with them. Upon investigation, we discovered that when we copied from one of our hard disks to our floppy drive, that some files were being corrupted. We currently have three hard disks connected to our Fortune. When copying from one of them to the floppy drive we got errors but when copying from the others to the floppy drive, we didn't see any errors. We also noted that when copying from this hard disk to our other hard disks, we would also get similar error.

Our short term problem became, "How can we copy the information from this hard disk to another new hard disk and check that the information was being copied correctly?" The normal **cp** command will not do this type of data verification so we had to turn to other UNIX commands to perform the proper data verification.

The basic process is as follows: First, copy a file from one place to another; Second, compare the two files; Third, if they are the same, fine. If not, then redo the first and second steps as many times as necessary until the two files are exactly the same.

There is a very useful UNIX command that will compare two files. This command is called **cmp**. If two files are the same, **cmp** says nothing, if the files are different, **cmp** announces the byte and line number at which the difference occurred.

We set up a shell script that sequentially copied each file from our defective disk to another one, did the compare and if there was a problem redid the copy as many times as necessary until we got a clean copy on our new disk. We intend to polish up this shell script and include it on an upcoming software disk so that it will be available for our readers.

The upshot of this discussion is to highlight the need for ensuring corruption-free copying. The **-X** option on the **cp** command provides an important level of protection. However, if you need more data verification, it is possible to use existing UNIX tools to build shell scripts that will give you what you need.

Dear /u/fortune news,

I am the Word Processing Manager for this firm. We are using the Fortune 32:16 (upgraded) and 30 MB Expansion Cabinet. I have been receiving your magazine for one year. I find it extremely helpful, particularly in the area of Glossary and Unix. I have acquired most of my knowledge through "hands-on" experimentation, learning most of the glossary work by trial and error. Due to the lack of information I have received in writing glossaries from the Fortune manuals, I am constantly in the process of self-education when it pertains to UNIX, records processing and glossary. This firm produces large quantities of paperwork, predominantly pleadings, so I am always interested in simplifying the word processing tasks of the users.

In one of your issues there was mention of the availability of a Fortune Advanced Glossary Use book. At present I have the Fortune:Word Records Processing manual, Fortune:Word Advanced Features manual and the Records Processing & Glossary Workshop manual. I would be interested in purchasing the Fortune Advanced Glossary Use book if it details the more advanced stages of glossary writing on the Fortune System and is not a repeat of the manuals I have mentioned above. Please let me know how I can order this book and its cost. I would be interested in any other books you may have that show samples of advanced glossary writing on Fortune.

Please let me know if there is available for purchase (on floppy disk) glossaries for the following tasks:

1. Automatic footnotes (system automatically places footnotes at bottom of pages and numbers them);

2. Automatically pulling legal cites from pleading documents. The table of contents and/or index generator are not feasible for this particular task.

3. Automatic numbering of paragraphs (we have a crude but workable glossary at this time).

I have created a number of glossary items for the users, some of which I am sure can be improved upon. I would appreciate any information you could give me regarding glossaries already created and available on floppy and/or programmers I could contact for help in creating such glossaries.

Sincerely,
Sally A. Henderson

Editor's Note: The Fortune Advanced Glossary Use book is available from your local dealer and/or Fortune Systems Corporation. It contains information on writing glossaries as well as numerous examples of actual glossaries. However, you should be aware that this book, and a floppy that contains all the glossaries, is included with the new version of Fortune:Word.

With regard to automatic footnotes, the new Fortune:Word handles them quite nicely. (For more on the new features in Fortune:Word 3.0, see the review article elsewhere in this issue). As for your second question concerning pulling legal cites from documents, we don't really know how to do that in Fortune:Word. We would suggest constructing a special program that keys on some character which would surround the cite. This program would simply search through the document and extract the information between the marker character. Perhaps several existing UNIX programs could be put together in a shell script to do this or even a simple C program could be written.

In any case, these problems may have already been solved by some of our other readers. If you feel you can help Sally out, please contact her at Lurie & Hertzberg, 213 274-8700, ext. 38.□

Fortune:Word

Continued from Page 1

Block Copy/Move/Delete

The Block functions make it possible to highlight rectangular columns of text which can then be copied, moved, or deleted. Normally with the copy/move/delete commands, text is highlighted line by line, making it impossible to work with a vertical column in the middle of several lines. Block highlighting overcomes this problem. It is useful for working with columns of numbers, columns of data, or columns of text. The program is intelligent enough to deal with each of these different kinds of blocks in the appropriate way, although we found that it was necessary to spend some time playing with this feature in order to be able to perform the tasks that we desired. Once mastered, it is a handy feature for special applications. We suggest that if you plan to use the Block functions that you should consult the reference manual for all of the options that can be employed.

Case Conversion

Fortune:Word 3.0 can automatically convert text to upper case and lower case, a feature found in many DOS word processing packages. In addition, it allows you to make the first character of each word capital, or the first letter of every sentence capital. This latter option is very useful for converting from all upper case to the normal lower case format, with sentences beginning with capitals.

Case Sensitive or Insensitive Search

When searching your text for a particular word or phrase, **Fortune:Word 3.0** will normally ignore the case, or capitalization of the text. If you want to match the capitalization exactly, precede your search text with an =.

Cursor Movement

Using the GO TO PAGE key, it is now possible to move directly to the next Tab, Indent, Decimal Tab, Center, Right-flush Tab, Required Space or Discretionary Hyphen.

When we first started using 3.0, we thought that an old bug had been left in. Usually, when defining text, you can have the cursor jump ahead to the next occurrence of a particular character, e.g. if you hit a period, the cursor will jump to the end of the sentence. However, hitting the TAB key while defining text produces a beep, and the cursor just sits there. This was true before and is still true. If you hit the GO TO PAGE key and the TAB though, the cursor will skip forward as expected. This is true for any of the characters mentioned above.

Severel Help functions have been added. Besides improving the help text, it is now possible to get help from any menu, while doing spelling correction, or from within a document.

Document Assembly

Many professions require mix and match documents — documents that are a collection of a stock of canned text. **Document Assembly** lets you combine many individual documents into one master document automatically, without using the SHIFT COPY (super copy) function to copy each one manually. In addition, unique text can be included in the master document surrounding each of the canned pieces. To include another document, all that you need to do is include a line like <include=para1> in your document. The sophisticated user (frustrated hacker) will appreciate the fact that includes can be nested to 16 levels.

Document Password Protection

In addition to the standard **For:Pro** password and account protection afforded any file, **Fortune:Word 3.0** allows you to assign a password from within **Fortune:Word** to any particular file. Once this is done, it is not possible to edit, copy, or print a document unless you know the password. If you use this feature, don't forget your montra, or you won't be able to work with the document in **Fortune:Word**.

Floating Footnotes

Fortune:Word does Footnotes! This feature allows you to print your footnotes at the bottom of each page, or at the end of a document. You can assign them numbers, letters, or *'s. You can have them set off from the text by any character you want, have a minimum number of text lines per page, continue long footnotes across pages, have prefixes for footnote references — like III-a, III-b, automatically number footnotes and more. In short, although we haven't used footnotes with other word processors, it seems that **Fortune:Word 3.0** does a good job in handling footnotes with a great deal of flexibility. (Note: When using footnotes it is best to use a single space format line. Also, be certain to turn off any attributes, such as undelining, that may have been turned on in the footnote.)

Forms Processing

Forms Processing is used when you have a canned document that just needs certain fixed information filled in — e.g. a name and address on an invoice.

Improved Help Screens

Several Help functions have been added. Besides improving the help text, it is now possible to get help from within any menu, while doing spelling correction, or from within a document. If you want help while you're editing a

Continued on page 20



"I checked. For us, it made more sense to buy a multi-user Fortune Systems computer than try to link up a bunch of independent PCs."

"We're a ten-person office. We needed an affordable computer system that everyone can use at the same time for different purposes yet allow us all to share information.

"We need a system that gave everyone computer access but that could also integrate the PCs we currently had.

"With a Fortune System from Micro Resources, everyone who needs it has instant access to information, word processing, financial programs and a central printer. All at a fraction of the cost of buying personal computers and software for everyone in the office."

From their inception, Fortune System computers have been designed to serve multiple users. There's no jerry-rigging of connections, no patch-work cables and "interfaces" that can break down and increase costs.

For years, Micro Resources has been designing, installing and servicing office automation systems for virtually any size business, from Mom & Pop stores to Fortune 500 office clusters. We apply the latest technology to meet the practical needs of business.

So if you're thinking about expanding the power of computers throughout your business, investigate a Fortune Systems multi-user computer from Micro Resources. We'll gladly show you how to speed work, lower costs and prepare for tomorrow's growth.

Micro Resources

535 Metro Place South, Dublin 766-2335

FORTUNE SYSTEMS

Authorized Fortune Master Dealer

Fortune:Word

Continued from Page 18

help screen on the bottom. One particularly helpful feature of help is the ability to get help for the COMMAND mode while editing. Many of the special features of **Fortune:Word** use the COMMAND key to initiate them. If you press COMMAND ?, each of the possible command choices will rotate through the top of the screen. When the one you desire appears, you can select it by pressing EXECUTE.

Insert/Delete within Insert

Have you ever been in the middle of inserting text and wanted to go back and edit the text you just typed in? Now you can.

Interface Enhancements

You've heard of "user friendly" right? Well **Fortune:Word 3.0** is a little more friendly than previous versions. The changes here seem to be primarily in the area of deletion protection. If you try to CANCEL an insert, or delete your changes, you will be given an explicit message asking you to confirm that you will lose changes.

Mathematical Calculations

Math capabilities have been added to **Fortune:Word**. Performing any of the four basic operations (+, -, * and /) is possible using numbers from the keyboard or within your text. Percentages can also be calculated. These functions required some experimentation to master. Although this is useful, it does have some limitations. Each number has to be selected one at a time (rather than defining an entire column for instance). Although it is possible to place the results into your text, you may have to reformat them.

Multiple Filing from Index

From within an index, it is possible to select several files to be archived, copied, printed or deleted.

Optional Foreign Language Modules

Language modules are available for German, French, Spanish and Swedish in addition to the standard English. Each module contains the necessary help files, message files, and dictionary files to operate in another language. If desired, **Fortune:Word** can be configured to print all menus and messages in one language, while performing Spelling Checking in another language.

Page Break without Insert

It is no longer necessary to press INSERT PAGE EXECUTE to insert a page break. Simply pressing the PAGE key will enter the page break. Likewise, INSERT SHIFT PAGE will produce a required page break.

Language modules are available for German, French, Spanish and Swedish in addition to the standard English. Each module contains the necessary help files, message files, and dictionary files to operate in another language.

Print from End of Edit Options

Another option has been added to the **End of Edit Option** which saves the document and goes directly to the **Print Document Screen**.

Print Documents from an Index

From any index screen, it is possible to print a document directly. By simply positioning the cursor on the document name and pressing SHIFT FORMAT, you will instruct the computer to print the document.

Print Queue Access while Editing

During an editing session, pressing COMMAND Q will bring up the **Printer Control** screen. It is possible to alter the queue without leaving your document.

Right-flush Tab

Right-flush tabs cause text to be aligned at the right. For instance the date in the following Memo format is Right-flush:

M e m o r a n d u m

May 15, 1987

The Right-flush tab defaults to the right margin, although these tabs can also be set on the format line, as is possible with any other tab.

Save Changes while Editing

This feature is almost as important as Autosave. At any time during an editing session, you may save your document by pressing COMMAND RETURN EXECUTE. Since it is so important to write to a disk to save your work, this is a very valuable feature. We suggest saving your document before performing any functions that might cause unexpected results. Then, if there's a problem, you can always revert to the earlier version.

Search with Attributes

You can search specifically for text with any attributes. If you just want to find the occurence of the word special that has been underlined, you can do that. However, since attributes put special hidden codes into the text, it is necessary to enter the MODE attribute command and the text for the search exactly as it was entered in the text in order to be successful in your search. Using the wildcard search described below, it is possible to find all of the text that is underlined, or double-striked, etc.

Shell Escapes from Menus

Since the advent of **Fortune:Word 2.0** it has been possible to escape to the shell while editing by pressing COMMAND !. We find this feature useful for accessing our **Electronic Rolodex**, or

it has been possible to escape to the shell while editing by pressing COMMAND !. We find this feature useful for accessing our **Electronic Rolodex**, or **Multiplan**, etc. from within **Fortune:Word**. **Fortune:Word 3.0** allows the same capability from any **Fortune:Word Menu**.

Shell Pipes into a Document

The COMMAND | (pipe) sequence sends data from a UNIX command into your document.

Sort Records in a Document

Fortune:Word 3.0 can now sort "records" within a document. In **Fortune:Word**, a record is defined as text that ends with a hard return. This could be single lines of text, such as a column of names or numbers, or it could be paragraphs that have been numbered. If you incorporate a blank line between paragraphs, all of the blank lines in the sorted material will be moved to the top of the document. Sorted material must also be contained on a single page. The sort command will sort in numeric or case-insensitive ASCII sequence, or in any Non-ASCII sequence that you specify (e.g. case-sensitive).

Spelling Checker Improvements

Fortune:Word 3.0 includes a new spelling dictionary and program which should result in better performance, both in speed and accuracy (making correct suggestions for words). The new spelling check also ignores the CONTROL-Y commands that are used with the HP Laserjet printers.

Suppression of Status Line Display for Faster System Throughput

That's a mouthful, but an important one. If you look at the status line on the top of the editing screen, you will note that the Status Line continually updates the Page, Line, Position and ruler crosshair to let you know where you are typing on the screen. As we have noted in the past, this means that for every key you type, as many as 5 characters may be written to the screen. Fast typists may find that this slows them (and the system) down. There is now an option to easily turn the status line update on or off. (COMMAND FORMAT toggles the display.)

This feature may be particularly helpful for modem users. In the past, **Fortune:Word** has been unworkable over a 1200 baud modem. If the display is turned off, it is possible to work on documents over a phone line. It's still slow for certain operations, particularly where the whole screen is rewritten. We found that if we typed in the INSERT mode, input was considerably faster.

View Index from Editing Screen

By pressing COMMAND i from the editing screen, you will be presented with a short index of the documents in your directory. (COMMAND I produces a long index.) You can use the GO TO PAGE feature to edit a document from this index, although the **Fortune:Word Windows** feature will

UNISOFT INC.

Experts in Computer Consulting - Sales - Service

Presents

FORTUNE SYSTEMS HELP

We Provide:

- Expert Trouble-shooting
- Years of programming and accounting experience
- *Free Initial Consultation*
- Training Available for:
  - * BASIC/IDOL/BAS and Multiplan
  - * UNIX and Ext. Fortune:Word
- Hardware Repairs & Upgrades
- FORTUNE trained experts at a cost that will surprise you!
- *Trade-Ins Welcome*
- Specializing in:
  - * Hard Disk Trade-In
  - * Hard Disk Upgrades

Authorized Fortune Systems Repair Depot
Fortune Systems National Value Added Reseller (VAR)
CALL: (303) 671-8600 EMERGENCY HELP: (303) 649-8505

UNISOFT INC. 14100 East Jewell Avenue Suite 27 Aurora, Colorado 80012

probably prove to be an easier way to edit multiple documents.

View Sublibrary Index

Fortune:Word 3.0 has the option to view an index of sublibraries, including all sub-directories and also the parent directory, without changing working libraries.

Wildcard Searches

Often times you want to search for a word that might have small variations within it. For instance, you might frequently enter John Jacob Jingleheimer Schmidt's name into your letters. However, you sometimes spell it Jinglehiemer, sometimes Jinglehimer and sometimes Jingleheimer. In the old days, you'd have to know every way you possible mispelled John's name. Now you can use wildcard characters to catch all of the names. When it says "*Search for what?*" you can enter `Jing*mer` and the computer will find all words that begin with `Jing` and end with `mer`. That encompasses all of our misspellings. You can even use the Search and Replace function to correct these spellings.

The above list summarizes the new features of **Fortune:Word 3.0**. We have found that many of these features simplify the use of **Fortune:Word**, like Right-Flush tabs, while others allow functions that were nearly impossible before, such as Floating Footnotes.

Performance

Fortune has stated that **Fortune:Word 3.0** includes performance enhancements. We did some tests on loading files, reformatting, searching for words, going to pages, etc., and found that in our tests **Fortune:Word 3.0** was not actually faster than **Fortune:Word 2.0**. In fact, in some cases it was slightly slower. However, because of the Status Line Update option, normal typing is probably faster. We also found that with certain functions, such as typing in columns of numbers, input was considerably faster if we entered the data in `INSERT` mode, rather than then normal mode.

We feel that any minor changes in performance are a minor consideration compared to all of the added features in the new product.

A New spelling dictionary and program should result in better performance, both in speed and accuracy.

Documentation

Fortune:Word 3.0 comes with a completely new set of documentation. The following manuals are included:

Self-paced Learning Guide
Reference Guide
Glossary User's Guide
Records Processing User's Guide

In addition to these manuals, a floppy disk containing the sample glossaries presented in the Glossary User's Guide is also included. These glossaries can be used as what Fortune Systems calls "productivity glossaries" — that is they can be used as is to perform many helpful functions, such as inserting the date into your document. These should be helpful for all Fortune:Word users. We wish Fortune had included an index of the entries by function, however.

These manuals have been completely redone for **Fortune:Word 3.0**. Much of the information contained in them is an exact replica of the HELP screens, although the Reference Guide is sometimes more elaborate. The new manuals are fairly complete and easy to read, however we sometimes found it difficult to find information using the indices. The Reference manual contains Appendices which list Error Messages and techniques for recovering corrupted documents. These are both useful additions.

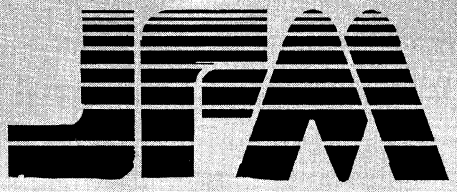
Conclusions

Fortune:Word 3.0 contains significant enhancements, particularly for power users who need things like footnote capabilities. The autosave features and ease with which you can work with your documents — printing from an index, multiple filing from an index, etc. — will be helpful for all users. As one user we spoke with commented, "It's full of all sorts of helpful goodies". Most of the features available in top word processors are included in this product, and we still find Fortune:Word to be the most "friendly" and easy to use word processor available. Features we look forward to in future releases are an Outline Processor, an Undo Command, a Thesaurus and terminal independence.

Josh Lobel

The following is the HELP screen for the COMMAND functions in Fortune:Word 3.0

TO ACCESS THIS FUNCTION: COMMAND, THEN PRESS:	PRESS
Block Copy (within a document)	COPY
Block Copy (between documents)	SHIFT/COPY
Block Move (within a document)	MOVE
Block Move (between documents)	SHIFT/MOVE
Block Delete	DELETE
Bookmark (0)	NOTE
Bookmark (1-9)	Number, NOTE
Case Conversion	c or C
Footnoting (enter a footnote)	n or N; number, n or N
Footnoting (return to a ref.)	n or N
GL (attach a glossary document)	GL
HELP (full-screen)	HELP
Horizontal Scroll	number, h or H
Hyphens (optional or required)	- (hyphen)
Index (short form)	i
Index (long form)	I
Length	PAGE, number, EXECUTE
Math	m or M
Printer Queue Control	q or Q
Required Space	Space bar
Restore Page	r or R
Right-flush Tab	INDENT
Save Changes While Editing	RETURN, EXECUTE
Screen Symbols	REPLACE
SEARCH (backward)	SEARCH
Shell Escapes (access shell)	!
Shell Escapes (redirect output)	|
Sort (ascending)	MERGE
Sort (descending)	SHIFT/MERGE
Spacing (line spacing)	s or S, letter or number
Status Lines (updating on/off)	FORMAT
TAB (next tab-stop position)	TAB
Vertical Scroll	number, v or V
Windowing	w or W
Flip between windows	f or F
Jump between windows	j or J



Dear Fortune Owner:

As you have probably read in */ulfortune news,* Fortune Systems SX models are significantly more powerful and faster than the PS and XP models. If you have a PS or XP model system, you should take advantage of our special promotion on SX upgrades. For a limited time, and while our supplies last, we will upgrade your system, including the new SX Cabinet at no additional charge for the cabinet. With this special offer, you may choose any combination of the following upgrades and/or expansions:

- *SX Processor with Current Release of Multi-User For:Pro*
- *60 MB Internal Streaming Tape*
- *30 MB, 45 MB, 70MB High Speed Hard Disk*
- *6-Port Intelligent Asynchronous Controller*
- *Upgrade to Current Release of all Fortune Software*
- *Fortune:Windows (Included at NO Charge)*

Additionally, we can perform these upgrades within 24 hours of receiving your system. In fact, if you ship your system overnight on a Friday for Saturday delivery, we will have it back to you on Monday morning, all ready to plug in and run; not losing the use of your system for a single work day. And, we will issue you a full 90-day warranty on the entire system.

JFM provides services to Fortune Systems owners either directly or through our authorized Affiliated Dealers. If you haven't used our services, I invite you to discover JFM; you will be glad you did.

Sincerely,

Morris Frydman
President

Talk to us, then talk to our clients



JFM BUSINESS SYSTEMS
The Merchandise Mart
Lobby Suite 144
Chicago, Illinois 60654
Call (312) 644-0870

Authorized Fortune Systems Master Dealer

Experienced BAS/Thoroughbred Specialists needed for:

- *Customer Support*
- *Programming*
- *Implementation*

Westchester County, NY

Chemical Background a plus.

Call or Write:
Dave L. Stein, V.P. of Operations
DataLogix Formula Systems, Inc.
200 Central Park Ave. Ste. 200
Hartsdale, NY 10530
(914) 997-1627

Foremost Supplier of Software for Formula Based Manufacturers.

Expand your Fortune® without spending a fortune.

New equipment costs big bucks. The Good Fortune Trader sells fully tested burned-in equipment, refurbished to Fortune® standards at prices that won't make waves with your cash flow. Call or write today for a listing of our specials.

THE GOOD FORTUNE TRADER

60 DAY WARRANTY

a division of Computer Services of Atlanta

(404) 392-9938

1534 Dunwoody Village Parkway/Atlanta, Georgia 30338

/u/fortune ***news***

73 Jason Street
Arlington, MA 02174

/u/fortune* news

The Newsletter for Users of the Fortune 32:16 Computer — June 1987/Volume 4 Number 6

SCI Buyout of Fortune Completed

Fortune Announces Formula 4000

Faster machine in familiar desktop box

(Fortune Systems recently announced a new product, the Formula 4000, a desktop version of the original Formula 8000 product. What follows is their announcement and press release for this product.)

Fortune Systems announces the latest addition to the Formula family, the **Fortune Formula 4000**. Like the **Fortune Formula 8000**, the **Formula 4000** offers full 32-bit architecture, featuring the Motorola MC68020 microprocessor operating at 16.5 MHz. The **Fortune Formula 4000** also utilizes the same **FOR:PRO** operating system, and will run the same applications as the larger **Formula 8000**. When **UNIX System V.3** is released, it will be available on both the **Formula 4000** and **8000**.

By virtue of being part of the same family as the **Formula 8000**, the **Formula 4000** will use the same memory cards and central logic bus cards, which include the intelligent asynchronous and communications controllers. Memory cards are currently available in 1 MB and 4 MB sizes, for a maximum RAM of 8 MB in the 2 memory slots.

The **Formula 4000** is available in 85 MB and 145 MB hard disk configurations, with a 40 MB configuration to follow. The 40 MB and 85 MB configuration will come standard with 1 MB RAM installed, while the 145 MB configuration will come standard with 4 MB RAM installed. The 85 MB and 145 MB configurations come standard with a 60 MB streaming tape, the streaming tape is optional in the 40 MB configuration. An added configuration is the 145 MB O/A system, which comes standard with **Fortune:Word** and **Fortune:Windows**. All systems come standard with the **FOR:PRO 3.0 Operating System**.

The **Fortune Formula 4000** will support a minimum of 4 devices, and may be extended to 10, 16 and 22 devices with multiple Intelligent Asynchronous Controllers. The **Formula 4000** is configured with one full-height hard disk module and one streaming tape module, both interface with the **SCSI Host Adapter**. The **SCSI Host Adapter** provides support for up to eight **SCSI** devices.

> Like the **Fortune Formula 8000**, the **Formula 4000** offers full 32-bit architecture, featuring the Motorola MC68020 microprocessor operating at 16.5 MHz.

The **Central Logic Bus (CLB)** includes the **SCSI Host Adapter**, for the winchester drive and tape interface. The **Formula 4000** provides space for three additional CLB controllers, which are identical to those used in both the **32:16** and the **Formula 8000**.

Fortune offers a 145 MB universal expansion cabinet. The expansion cabinet's versatility allows it to function with the **SCSI SX** product and both **Formulas**. By adding the expansion cabinet, maximum storage capacity

See News, page 2

The BASIC Advisor

Ray Wannall is President of BaSiC Software Corporation in Baltimore, MD, and is contributing independently to this publication.

Question: *I just received my copy of* ***/u/fortune news,*** *volume 4, number 5, and noticed that you referred to "Jim Dyson of Fortune Systems in Monaco, CA." For the record, Jim is in Monaco, the country, not in California.*

P.S. I hear the weather there is great!

Answer: My heart bleeds for him.

Question: *Is it possible to have an*

See **BASIC**, page 21

In This Issue...

Formula 4000 introduced, **SCI** buy out approved–Read all about it in **News From Fortune**...**Page 1**

Customizing report totals for BAS Inventory program, in the **BASIC Advisor**...**Page 1**

Do you need to transfer files between computers more quickly? **SST's Blazer** product can help...**Page 4**

In **Part 2** of **Dave Kloes' System Adminstrator's** series, he discusses system files and updates...**Page 10**

Verified Copy–a shell script to insure error-free copying from disk to disk...**Page 16**

*/u/fortune *news* is published by The Cambridge Consortium, Inc. It is not affiliated with Fortune Systems Corporation.

/u/fortune *news*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz

Technical Editor
Scott Bradner

Advertisting Manager
Mark Palmerino

/u/**fortune** *news* is published 12 times per year by The Cambridge Consortium, Inc. Neither /u/**fortune** *news* nor The Cambridge Consortium are affiliated with Fortune Systems Corporation.

Subscriptions cost $48 for one year within the U.S.A. For foreign subscriptions, add $27 each year for added postage and handling costs.

Back issues are offered at:

Vol 1, Numbers 1 - 3	$1/each
Vol 1, Number 4	$.50/each
Vol 1, Numbers 5 - 8	$3/each
Vol 2, Numbers 1 - 7	$3/each
Vol 3, Numbers 1 - 12	$4/each
Vol 4, Numbers 1 -	$4/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $10 each. Please add $5 per order for foreign destinations.

Although every attempt is made to insure that our contents are error-free and without risk, The Cambridge Consortium is not responsible for losses that occur due to correct or incorrect interpretation of /u/**fortune** *news*.

For more information, call or write:

/u/**fortune** *news*
73 Jason Street
Arlington, MA 02174
(617) 663-7700, (617) 646-5603

The text for this newsletter was prepared using a Fortune 32:16 for text input. The information was then transferred to an Apple Macintosh Plus with kermit and formatted for typesetting using the Aldus Pasgemaker program.

UNIX™ is a trademark of Bell Laboratories, Inc. Fortune 32:16™, Fortune:Word™ and FOR:PRO™ are trademarks of Fortune Systems Corporation. Multiplan™ and MS-DOS™ are trademarks of Microsoft. IBM-PC™ is a trademark of IBM. Macintosh Plus™ is a trademark of Apple Computer. Pagemaker™ is a trademark of Aldus Corporation.

©1987 by the Cambridge Consortium, Inc. 6/87

News

continued from page 1

can be increased to 435 MB.

An upgrade kit is required to connect the **Formula 4000** to the 145 MB universal expansion cabinet. This kit contains a connector, a ribbon cable and installation instructions.

Software applications which run on the **Fortune 32:16** family under **FOR:PRO 1.8** and **2.0** will need to be recompiled before they will run on either the **Formula 4000** or **8000**. *(Editor's note: This means that if you currently own a **32:16** machine you will need to obtain new versions of your software to run on the **Formula** line.)* All software which will run on the **Formula 8000** will also run on the **Formula 4000** without modification.

Product Comparison

It is important to differentiate between the **Formula 4000** and **8000**. The **Formula 4000** is positioned as an entry level MC68020 system. It fulfills the need of end-users who have limited upgrade requirements, but require the speed and power of a MC68020 system. For expansion, and higher-end needs, the **Formula 8000** is the answer. The **Formula 8000** has larger memory capabilities, greater connectivity, and a gateway to standard architectures. The **Formula 8000** provides room for two full height hard disk drives, and in the future will have larger capacity drives than the **Formula 4000**. Note the table for a concise product differentiation.

	4000	8000
Processor	MC68020	MC68020
Speed (MHz)	16.5	16.5
Max. RAM (MB)	8	16
Max. Devices	22	28+
I/O Slots	4(3)	5(4)
32-bit Slots	0	4
Disk	40/85/145	85/145/+
Internal Disk	1	2
Tape	60	60+
VME Gateway	NO	YES

* * * *

SCI Buyout of Fortune approved by stockholders

The last Annual Meeting of Stockholders of the **Fortune Systems Corporation** took place on June 29, 1987. A number of significant proposals were passed at the meeting including the proposal to sell substantially all of the operating assets of its computer hardware business to **SCI Technology, Inc**. These assets include manufacturing, research and development, support, marketing, and

A number of significant proposals were passed at the meeting including the proposal to sell substantially all of the operating assets of its computer hardware business to **SCI Technology, Inc.**

sales functions as well as the "Fortune" trade name and logo. Under the terms of the transaction, Fortune Systems Corporation will continue as a corporation under a new name and will retain its cash and cash equivalents, tax attributes, accounts and notes receivable, certain other assets and liabilities, and its software subsidiary, **Tigera Corporation**. As indicated in another proposal that was passed, the new name of the parent corporation will be **Tigera Group, Inc.** (TYGR). The subsidiary was originally formed to modify Fortune:Word and Fortune:Windows and to develop other office automation software applications all of which will run on machines manufactured by other vendors (including, possibly, existing Fortune equipment). These products are tailored to run in UNIX environ-

ments. Agreements have already been signed with several OEM companies including **Convergent Technologies, Inc.**, **Plexus Computers, Inc.**, and **Pyramid Technology Company**. The new parent company, Tigera Group, Inc., intends to emphasize and expand their software business, while seeking out attractive acquisition opportunities in a number of areas that would combine synergistically with current assets.

The transaction with SCI contains a special software agreement between SCI and Tigera which details the rights to each of the parties as they relate to current Tigera products. In the event that SCI or one of its authorized resellers ports and sells Fortune:Word or Fortune:Windows for use on non-SCI/Fortune equipment, royalties must be paid to Tigera for a three-year period. Tigera will pay a similar royalty for a similar period to SCI in the event that it licenses its **WORD ERA** product for use on SCI/Fortune hardware. Each party is obligated to protect the security and confidentiality of the base product source code and each party is entitled to develop and market enhanced and modified versions of the software.

The stockholders also elected a new Board of Directors. The new board consists of six directors, two of whom are principals of a new, major stockholder in what was Fortune Systems Corporation — **Citicorp Capital Investors Ltd.** (See the article below for details on this development). In addition, proposals were passed that limit the legal liability of directors and that make an unfriendly takeover of the company more difficult.

In a development related to these changes, **Paul Olin** has ended his association with Fortune Systems. Originally, Olin was slated to direct, for SCI, the marketing and distribution of Fortune hardware through the master dealer network. **Robert Davis** has agreed to manage these efforts until a new person has been recruited for this position. As noted in the last issue of */u/ fortune news*, Davis and **Brooke "Pete" Taylor** have agreed to purchase **Fortune Systems International** (FSI), the international marketer of Fortune equipment, from SCI. FSI will retain the exclusive right to distribute Fortune and multi-user SCI equipment outside of North, Central, and South America along with the non-exclusive right to distribute other SCI products.

The effects of these significant changes for Fortune equipment end-users remain to be seen. Clearly, the future of Fortune end-users lies in the hands of SCI. As a well-endowed company that has put its money on the line, we have reason to be optimistic about SCI's commitment to this equipment and its continuing development. But, seeing is believing, as we will monitor SCI's policies. In the future, the actions of the old Fortune Systems Corporation (now, the Tigera Group, Inc.) will have less significance for us. However, the software applications developed there will surely be relevant for maintaining the effectiveness of Fortune hardware.

* * * *

Citicorp buys Fortune Systems Stock

On April 24, **Citicorp Capital Investors Ltd.** bought an option to acquire 3.94 million shares of Fortune Systems Corporation for $7.9 million from **Alcatel** USA Corporation, a subsidiary of the French nationalized conglomerate **Compagnie Generale d'Electricite (CGE)**. This option to buy 18.9% of Fortune Systems cost $100,000. On May 29, the Citicorp group exercised the option and purchased the stock. In earlier years, CGE had had a marketing agreement with Fortune Systems to distribute Fortune equipment in certain areas of the world, including France. With the transfer of Fortune's hardware assets to SCI, CGE's interest in the restructured Tigera Group, Inc. clearly diminished. The character of the Citicorp group's interest in Tigera remains to be clarified. However, the group's two representatives on the new Board of Directors may show something about the restructured company's posture.

Information in the proxy statement for stockholders from Fortune Systems Corporation indicates that the two new directors are **Mart E. Bakal**, 36, an attorney and investment banker and **R. Peter M. Elrick**, 41, an investment banker. Mr. Bakal is a faculty member of the Harvard Business School in Boston and also serves as a First Vice President of the investment banking firm Drexel, Burnham, Lambert in New York. He also presently serves as a board member for Morison Asset Management Fund. Mr. Bakal was previously Senior Vice President of the investment banking firm Bear, Stearns in New York from 1985 to 1986. Prior to that he was a Visiting Scholar at Harvard Law School from 1984 to 1985 and a corporate attorney with the law firm of Fillmore & Riley from 1981 to 1984.

Mr. Elrick has been president of Elnet & Co., Inc., an investment banking firm, specializing in small and medium size public companies, since January 1987. He also serves as a board member for Masstor Systems Corporation, Britton Lee Inc., Intelligent Business Communications Corp. and other private companies. From 1978 to 1986 Mr. Elrick was Vice President, Allen & Co., Inc., specializing in financing and advisory activities for small and medium size private and public companies. □

Benzion Chanowitz

SST Data Releases BLAZER

Performs high speed file transfer with ease

Lightning Communications

We have frequently discussed communications programs in these pages. These packages are used to share information between computers located in the same office or across the country. We frequently use a modem to access our office computer from home, or to send electronic mail to Fortune Systems, or to download a public domain program from a user in California. We discuss communications regularly because we think that they simplify many things.

Unfortunately, setting up communications devices (wires and/or modems) and software is not always easy. Fortune's basic communications packages, **ITE (Interactive Terminal Emulator)** and **FFCP (Fortune-to-Fortune Copy)** are functional, but not particularly friendly. The **kermit** program that is widely distributed by us and others is very functional, basically free, and moderately friendly. This article is a discussion of a new software product called **Handshake Blazer** from **SST Data Corporation**, which is both functional and very friendly. It is also very fast.

Before we begin, we'll present a short review of how communications programs usually work. Normally when you work on your computer you are computing in isolation, your keystrokes go into the computer, and the letters appear on your screen. In order to communicate with the outside world, you need a program to "redirect" what you type and what the computer puts out to a modem or to a wire. Typically, you will have a port set up to communicate, and you will direct the information to flow in and out of that port.

Programs like **kermit**, **ITE**, and **Handshake ITE** all provide terminal emulators which allow you to log in on a remote computer. Once you log in, it is just the same as if you were using their computer in your office, with one caveat — typically you will be communicating at slow speeds, like 1200 or 2400 baud, compared to the usual 9600 or 19200 that are used with a direct wire connection. These programs can also be used to send files between two systems, but the transfer may be unreliable. If there is noise on the phone line for instance, some of your file may be lost, and new garbage characters may be introduced. In addition, these programs can only transfer ASCII text, which means that programs, **Multiplan** spreadsheets and **Fortune:Word** formatting information cannot be transferred with them.

If you are interested in error-free file transfer you will need to use a program that has error correction.

If you are interested in error-free file transfer, you will need to use a program that has error correction. Typically this means that you have a program running on both computers talking to each other. One sends a "package" (also known as a "packet") of information, and the other responds confirming that what was sent was the same as what was received. If there was a problem, the same packet is sent again. The packets are sent one after another until the entire file has been transferred. (You may wonder how the receiving program knows how to verify the accuracy of what was sent without actually knowing what was sent. At first it may seem like a computer Catch 22. In fact, there are many schemes that allow computers to have this intuition. One method is to use what is called a checksum. This is a number which is a sum of all of the bytes in the packet. The sending computer sends the packet followed by the checksum. The receiving computer receives the packet, calculates the checksum, and compares it to the checksum it received. If the two checksums don't match, there was a problem and the data is retransmitted. Simple, right?)

As we just stated, the communicating programs have to be working on both machines for the error checking to take place. With the **kermit** program, the user typically manually logs in on the second machine, and starts the remote **kermit** going in either the send or receive mode. Once that is done, he/she returns to their own machine and starts **kermit** in the appropriate mode there. This gives the user flexibility about just where the files should go, how they might be manipulated, etc. because everything is done in an interactive manner. The disadvantage of this approach is that the user has to stay with the transfer from the beginning to its completion.

Other programs, like **FFCP** and **Blazer**, operate in an automatic mode. Once a command is given to copy a file, the initiating machine finds the phone number for the remote system, dials the modem, logs in, starts the remote program, transfers any pending files, logs off and hangs up the phone. The advantage of this approach is that once the command is given to transfer the files, the user is done and the computer takes over.

Handshake Blazer

Handshake Blazer was designed to duplicate all of the functions of **Fortune-to-Fortune Copy (FFCP/UUCP)** while adding several enhancements:

- User Friendly Interface
- High Speed (9600 baud+) support
- Friendly front-end to FFCP
- Partial File Transmission Recovery

We received a demonstration copy of **Blazer** and have been using it for our internal communications for the past two months. We have found that it fulfills its goals well.

Installation

Blazer is installed through the **Fortune Global Menu**. Like other Fortune

software, each copy can only be installed on one machine. The installation is a relatively straightforward procedure. You are given the choice for the **Global Menu** location for **Blazer**. Once the main program is installed, it is also necessary to configure it for your system and all of the systems you will be communicating with. With **FFCP**, it is necessary to use an editor to make changes in several systems files for each system you want to include. **Blazer** uses menus to simplify this process, making it possible for less sophisticated users to maintain the product. The menus are consistent with the **Fortune:Word** bar menu so that anyone familiar with that system will feel at ease. These menus also have a striking similarity to other **Handshake** products.

Even with the menus, it may be necessary to experiment some to get your communications established. Since **Blazer** will automatically log in to the remote machine, it is necessary to duplicate the log in process exactly. Confirming the phone number and log in procedure may take more than one try, although if you are only talking to standard Fortune computers, you won't have much difficulty.

When you need help

Blazer incorporates the use of the `HELP` key. When you get stuck in any of these phases, you can always get the appropriate `HELP` by gently pressing that key. Sometimes it even gives you hints about the best way to use the system. We found the help screens helpful.

Using Blazer

Once installed, **Blazer** can be used either through the **Global Menu** or from the shell. The main **Blazer** menu looks like Figure 1.

As you can see, the options are clearly presented. After choosing either Send or Receive, you will be prompted to enter the system name and files that you want to send. From the local computer, it is permissible to use wild-

continued on next page

FORTUNE USERS:

SIMPLIFY SYSTEM-TO-SYSTEM FILE TRANSFER
TRADE UP AND SAVE MORE THAN 75%

Is your Fortune to Fortune Copy cumbersome, difficult to set up and use?

Do you want easy to understand menus so anyone can send files without shouting for help?

Are you frustrated by the complex setup and maintenace required by Fortune to Fortune Copy?

Would you like to have an easily maintainable Fortune to Fortune network with solid security?

YOU NEED HANDSHAKE BLAZER!

TRADE UP your Fortune to Fortune Copy
for HANDSHAKE BLAZER and save $$$$

HANDSHAKE BLAZER standard list is $695

Limited Time Offer
ONLY $150

This special offer good thru August 31, 1987.

For more information call:

SST DATA INC. 250 S. Main St. #201
Thiensville, WI 53092
(414) 242-3999
or your LOCAL FORTUNE DEALER

Figure 1

HANDSHAKE-BLAZER (TM) by SST DATA, INC.
HIGH SPEED DATA TRANSFER SYSTEM

Please select next activity Mon Jun 29, 1987 5:05 PM

Send File To Remote System
Receive File From Remote
Send Mail To Remote System
File Transfer Activity Logs
Background Activity Status
Parameter Maintenance

You are: root
on directory: /u/newsletter

Blazer

continued from page 5

card characters such as * or ? to specify file names. If you use wildcard characters on the sending end, the **Blazer** menu system will find all the files that match the wildcard and display them on the screen. On the remote end, filenames must be provided explicitly.

The third option raises a use of communications we haven't discussed yet — **Electronic Mail**. **Electronic Mail** is used to send messages to users at remote locations. In order to use **Electronic Mail** with **Handshake Blazer**, you must own **Fortune's Development Utilities** which include the **mail** command. **SST Data** incorporates a text editor that is used to create messages which works acceptably for simple messages. Unfortunately its commands are not like **Fortune: Word**, but with a small effort you will be able to use it without any major problems. If you have something complicated to send, you can always use **Fortune:Word**, and send it as a complete file. (When we mentioned this limitation to the people at **SST Data**, they explained to us how to change the default editor to **Fortune:Word**, or any other editor of our choice.) The **mail** option allows you to specify several users at a particular site, but you can only specify one site at a time. Future releases of **Blazer** will incorporate support for groups, aliases and distribution lists, which will simplify the distribution of information in a corporate setting.

Maintenance

There are several options on the menu that allow you to monitor the **Blazer's** activity. These report statistics on what calls were made and which files were transferred. If you sent a file to a remote system, **Blazer** logs how much time was needed to make the transfer.

If the file(s) was not successfully transferred on the first try (perhaps the line was busy) Blazer will repeatedly try to send the file. You can monitor this activity by viewing the **File Transfer Activity Log** and the **Background Activity Status** from the main **Blazer** menu. You can view all activity, or easily select a particular site name.

The **Parameter Maintenance** selection from the menu is used to define ports, install new remote sites, tell the computer about your modem, etc. The menus are generally well prepared and easy to follow. You can see a sample of them in Figure 2.

These menus make Blazer an easy product to use.

Blazer's Lightning Speed

The **Blazer** product was written to perform high-speed data transfers. This is accomplished in two ways: 1) **Blazer** supports high-speed modems and 2) **Blazer's** error checking protocol is very efficient. We explained in the section on general communications that error checking involves sending messages back and forth to confirm that the data was received

COMMUNICATION GAP

Is your Boss difficult to talk to?
Maybe with all the MONEY and TIME you save on data transfer
using TELEBIT MODEMS, you can get him to speak your language.

ABSOLUTELY The Lowest Prices on TELEBIT MODEMS and COMMUNICATION SOFTWARE

COMPUTER TRADE DEVELOPMENT
(313) 271-3020 EXT 34
One Parklane Blvd., Suite 1430 East
Dearborn MI 48126

correctly. **Fortune-to-Fortune Copy** was designed to send error-free data, which it does. However, as **Michael Myers** of **SST Data** says, "**UUCP** (the generic, AT&T version of **FFCP**) was designed assuming that there will be a noisy line. This was often true in the old days of acoustic modems. It's not true any more. We designed **Blazer** assuming that the line would be good — that allowed us to not use up error checking time unless we need to".

If time is a consideration, **Blazer** will outperform either **kermit** or **FFCP** in terms of transfer time. This is especially true with high speed modems

FFCP does the traditional thing, which is to send a packet, verify it, and send the next. With each packet, the computers have to pause to receive the verification before sending more information. **Blazer** sends packets for about a minute and then pauses an instant to check the correctness of all the packets it just received. If any are not right, their position is recorded in a error table. Once the file transfer is completed, any packets that were incorrect are sent again. This means the data comes almost in a steady stream rather than the usual back and forth with other communication programs. Since the sending computer does not have to constantly wait for an acknowledgment, it sends the data much more quickly. If the line is extremely bad causing an excessive number of bad packets, **Blazer** falls back down to the more traditional error checking method.

What does this mean?

It means that **Blazer** is very fast at transmitting files. We did a sample test at 1200 baud between two Fortune computers using **Blazer**, **FFCP** and **kermit**. We transmitted a file that was 33,395 bytes long. The following is a chart that represents the total line charge — from the moment the phone went off-hook until the line was hung up.

Blazer	5:34
FFCP	6:06
kermit	9:02

As you can see, Blazer was the winner for this file. These times include the time it took to dial the modem, make the connection, log in and transfer the file. With **Blazer** and **FFCP**, this all happened automatically, while with **kermit**, the process was all done manually. With longer files, the transfer would be proportionately faster since the dial and log in process would only need to be done once for the entire session. Blazer typically sends information with about 3-4% overhead for checking. This means if your modem is operating at 1200 baud, or about 120 characters per second, the files will get sent at roughly 116 cps. **Kermit** typically operates at about 65 cps.

We tried to do the same test using **TrailBlazer** modems running at 9600 baud. Unfortunately, we could not do the comparison because **kermit** and **FFCP** did not work properly at that speed. The people at **SST Data** felt the reason that these programs did not handle the high speeds was that they would transfer the packets so quickly, that they would end up waiting for the verification too long, and end up timing out. Since **Blazer** was designed to operate at high speeds, it supports 9600 baud plus. In fact it has been used to transfer information via satellite, which is even more demanding than ground transmission.

At 9600 baud, **Blazer** transmitted the 39,000 byte file in 1:39. The actual time to transmit the file was 36 seconds. The rest of the time is for modem dialing, sign in, log out, etc.

File Recovery

Suppose you are sending a 5 meg file

continued on next page

Figure 2

HANDSHAKE-BLAZER (TM) by SST DATA, INC.
Parameter Maintenance

Please select next activity Tue Jun 30, 1987 3:08 PM

Systems File Maintenance
Devices File Maintenance
Permissions File Maintenance
Other Parameters
Activity Logfile Clearing
Remove Unwanted Jobs

You are: root

HANDSHAKE-BLAZER (TM) by SST DATA, INC.
Systems File Maintenance

Enter system name: _____

Every computer has a unique system name.
Each remote computer that you with to queue work for must be entered into the 'Systems' file.
This record identifies another computer and indicates how to access that computer.
Enter a system name that you wish to maintain (or add).
You may also enter '?' to see an index of all known systems.

Blazer

continued from page 7

across country. After sending 90% of the file (which at 9600 baud only takes 78 minutes — at 1200 baud it would take over 10 hours) the line drops. With **FFCP** or **kermit**, you would have to start from scratch and retransmit the entire file. With **Blazer**, the program keeps track of how many packets have been sent, and only sends the missing end piece. With long files, this is obviously a valuable feature.

Using Blazer from the Shell

Blazer has commands that correspond to most **FFCP** shell commands. For instance, the **pass** command replaces **uucp**; **blazer** functions like **uucico**, etc. One useful command is called **blaztry**, which can be used for debugging **Blazer** sessions. This command prints out what **Blazer** is doing while a transmission is going on. That information can be very useful if you're having trouble establishing a connection with another computer. The other commands function as any **uucp** user would expect.

Using Blazer with uucp

Some people may establish a network where some computers use **FFCP** (or **uucp**) and other use **Blazer**. You can use the **Blazer** menu system to talk to any of these computers. If **Blazer** cannot find a system name in it's files, it will check the **L.sys** file. If the system is found, the transfers will be done using **FFCP** (or **uucp**). The main advantage to this approach is that the **Blazer** menu system is easy to use. Although the **Blazer Menu** works with **FFCP/UUCP**, please note that to actually transfer files you need to have a matching pair of programs on both computers--either **Blazer** or **FFCP/UUCP**. Blazer can only be used to communicate with **UUCP** if **UUCP** is installed on your system.

Documentation

Blazer is packaged with a small looseleaf manual that is now the standard with many software products. We found the manual to be clear and complete, and were able to start using **Blazer** without any problems.

Conclusions

For users who are sending files between computers on a regular basis, **Blazer** provides a friendly way to accomplish this. If time is a consideration, **Blazer** will outperform either **kermit** or **FFCP** in terms of transfer time. This is especially true when high speed modems (9600 baud and higher) are used to communicate. The file recovery feature for partial transfers is also a marked advantage. In order to use **Blazer**, you must have the program running on both computers that are communicating with each other. It is currently available for all Fortune computers and the AT&T 3B line. Ports are in progress for PC DOS machines.

TrailBlazer Modems

In order to fully appreciate **Blazer**'s speed over phone lines, you will want to use a 9600 baud modem. We were loaned two **Telebit TrailBlazer** modems for our tests. (Note that the word ***Blazer*** in both product's names is a coincidence, **Telebit** and **SST** are separate companies.) These modems can communicate at 300, 1200, 2400 and 9600 baud. They operate somewhat differently than slower modems. For one thing, it is possible to set one speed for the interface to the computer, and another for the modem speed over the phone line. This means that you can set up the modem to always log into your system at 9600 baud. However, if someone calls with a 1200 baud modem, the **TrailBlazer** will talk to the other modem at 1200 baud, but still talk to your Fortune at 9600 baud. This simplifies interfacing problems with the Fortune if people call in at varying speeds.

Our experience with the **TrailBlazers** showed them to be very efficient for file transfers. At high speeds, they lived up to their specs. We don't have the technical means to evaluate them, but we are told that these modems are very good at handling noisy phone lines. Rather than fall back from 9600 to 7200 to 4800, etc., the **TrailBlazer** modems fall back 100 bits per second at a time, until a successful communication can be achieved. This makes them very robust when dealing with problem lines. They also employ a unique scheme which divides the audio spectrum of the phone line into 500 sections, and they can avoid specific frequencies that are particularly noisy.

In order to fully appreciate **Blazer**'s speed over phone lines, you will want to use a 9600 baud modem.

The modems weren't as satisfactory for interactive communications, such as using **Fortune:Word** over a phone line. Because of their high speed sophistication, they have a long "line turn around time". We won't go into a long explanation about why this is, but the effect is that you will be able to type ahead of the modem. Telebit is apparently working to improve their turn around time while maintaining their superior high speed transfer capabilities.

One other caution about 9600 baud modems in general. At the moment, there isn't a protocol standard for 9600 baud modems. This means that it is necessary to use like-branded modems for both sides of the communication at 9600 baud. If you are setting up a network, you will have to purchase only one brand of modem.

Given these cautions, the **TrailBlazer** seems to function very well with **SST Data's Blazer** software. If you want to move files quickly, you won't be disappointed. □

Josh Lobel

MAKE YOUR FORTUNE GROW

One of the greatest assets of your Fortune Systems 32:16 is its built-in ability to expand to match your need.

Whether you want to increase your number of workstations, upgrade system capability, or tie-in your Fortune Systems cluster to a mainframe, Micro Resources can give you the prompt, complete service you need.

As *the* Fortune Systems Master Dealer in Ohio, we have the hardware and software you need available for immediate shipment. From Fortune:Link to uninterruptible power supplies, from storage expansion systems to Development Utilities, we have the exact blend of hardware, software and systems expertise needed to make your Fortune Systems 32:16 do exactly what you want it to in the most reliable and cost-efficient manner possible.

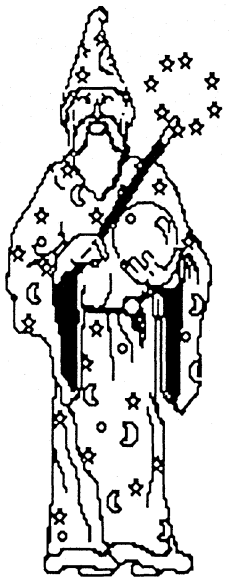
And our genuinely friendly staff programmers and systems analysts are available to provide you with whatever additional support you may need, from software consultation and customization to integrated systems design and specification.

So whether you need a quick question answered or a new service agreement, Business Applications Update or just some networking cable, call Micro Resources.

We'll help your Fortune Grow.

Micro Resources

535 Metro Place South Dublin, Ohio 43017 (614) 766-2335



System Administration

No Longer Requires a Degree in UNIX Wizardry

This series is a must for all owners and users of the Fortune computer. Dave is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the Fortune Computer, and is president of the Houston UNIX User's Group. He contributes independently to ***/u/fortune news.***

It seems like we just finished writing one of these segments and it's time to write another. In any case, here we are again with Part 2 of the System Administration series. In this segment, we will take a walk through each of the *directories* in **For:Pro** (**UNIX**) and discuss the information and commands that are stored in each. This will serve as groundwork for some of our more in-depth discussions later on. We will also talk about some other issues as appropriate.

Before we begin, however, we need to talk about a point we made in Part 1. One of the responsibilities of the system administrator is to insure that the software on the system is up-to-date. Each software package that is loaded on the system has a *version number* assigned to it. The version number can be identified in a number of different ways. It should be on the **master** diskette that was loaded on the system (a master diskette generally has a Fortune label and packaging). It is sometimes displayed while the software program is running. If all else fails, the **what** command can be used to get the version number. The current version of For:Pro is 2.0 for the Fortune 32:16. To find out your version, login as **root** and enter the following command:

```
# what /unix
/unix
                For:Pro 2.0
```

Here are a few other commands to find current release levels:

```
what /m/wp2/wp
```
(Current version of Fortune:Word)

```
what /m/mp/mp
```
(Current version of Multiplan)

```
what /b/basic.psd
```
(Current version of Basic Interpreter)

Normally, system administrators will want to keep each software package on their system current. Occasionally, **/u/fortune** *news* will publish a list of the current software versions for you to compare against. Deciding when or whether to order a software update or upgrade is not always cut and dry. Here are some of the considerations:

1. Sometimes you will not want to order an update immediately. You may want to wait a month or two to make sure that any major "bugs" in it have been worked out. We normally run a major release on our system before we contact our clients for orders. If in doubt, contact your dealer to see if they have used it or talk to someone else who may have already installed it.

2. Does the update have any major enhancements in it or does it fix bugs in previous versions? Enhancements may make the software run faster or provide new utilities. If you are having problems with a current version, the update may fix those problems. Software that runs faster or has new utilities may be worth the cost in terms of increased efficiency.

3. Does this release affect other software on your system? For example, when For:Pro 2.0 is installed, you must also install the new Thoroughbred Basic Interpreter (6.5.12).

4. The decision to upgrade some software packages such as the Business Accounting System (BAS) are not made easily. These should always be done in consultation with the company you are using to support those applications. Installing these new releases may overlay custom program modifications or may impact the structure of the Database files on the system. As you have probably already witnessed by reading the **BASIC Advisor** column in this publication, some real problems can surface. Newer does not always mean better.

5. We almost always recommend that, as a minimum, you stay current with the latest For:Pro operating system. First of all, at least until For:Pro 2.0, you will get increased (faster) performance. Secondly, Fortune normally does not support previous versions of the operating system. Thirdly, all new software releases by Fortune are based on the current operating system release. For example, a new release of **Fortune:Word** may not work on any version of For:Pro except the current version. Sometimes there is "good news" and "bad news" associated with a new release of software. Take For:Pro 2.0 for example. What we were told about For:Pro 2.0 was that "For:Pro 2.0 improves performance for random disk reads by approximately 10% over For:Pro 1.8 ...". What we were not told was that the size of the *kernel* (that part of For:Pro that is always resident in memory) increased from 132K (132,000 characters) to 216K (216,000 characters). This means the new kernel uses up almost one 256K memory board by itself and what is left over is used to process our work. The net result is that your system will run slower when For:Pro 2.0 is installed *unless* you add more memory. Granted there were other enhancements made to For:Pro that may justify the upgrade, but the decision becomes more difficult when we are comparing the cost of For:Pro 2.0 by itself against the cost of For:Pro 2.0, the added memory, and the cost to install it.

6. Fortune has recently announced that "upgrades" and "updates" will only be available at a reduced price for 90 days. If you do not order within 90 days of the release date, you will have to pay the full price for the update. This may, for example, be the difference between $495.00 for the upgrade to Fortune:Word 3.0 versus the full price of $1495.00.

7. The cost of the update may be a major concern for you. We all have our business cycles to contend with. When the business is "up" rather than "down", ordering an update may not be a major problem. When the business is "down", you sometimes have to decide whether the advantages of the upgrade justify the cost based on your current economic environment. In Houston right now, as an example, the decision to buy a $150.00 update package is a major one for some companies. This is especially true if you also have to pay someone to come out and install it for you. These are judgments the administrator has to make. The decision to postpone the purchase of a new release of software may, at times, be a valid one.

We bring up the considerations about upgrades at this point for two reasons. *One is that we assume that you are all running the current version of 2.0 For:Pro and therefore all of the material in this series is based on this version.* If you have

Our Memory Doesn't Cost You A Fortune

Would you like to:

Speed up your computer?
Add more users?
Use the latest software, like Fortune:Windows™?

Compatibility:	**The F1MB is absolutely 100% compatible with Fortune's own 1 Meg memory board.**
Affordability:	**USA is proud to have equaled and exceeded the quality and performance of our competition, while offering substantial savings for both the dealer and end user.**
Dependability:	**All boards are burned in for 72 hours and tested for 48 prior to shipment. Our One Year Repair or Replace Warranty underlines our confidency in the product.**
Repairability:	**The F1MB facilitates repair in the field as it is 100% RAM socketed. Now you can repair a board by simply replacing a defective chip rather than face an expensive board exchange -- reducing maintenance costs and system down time.**

If you have a Fortune SX, XP45 or recent model XP30, you can use our board by simply plugging it in. Older systems can be upgraded to use our board with a simple and economical modification.

For Details Call: ***Upgrade Service Associates***
850 Lexington Street, Suite 15, Waltham, MA 02154 Phone: (617) 239-8147

Our Guarantee: **If you're not completely satisfied within 30 days, USA will gladly refund your money.**

other versions of For:Pro, the material may not apply. The other is that, as you can see, many factors are taken into account when deciding to purchase new software or software upgrades. We do not expect you to keep up on all this information because that's what dealers are for. You should, however, be aware of the considerations mentioned above in order to make a more intelligent decision.

Let's return to the material at hand and talk about the For:Pro directory structure and the files found there. If you login as root and do an **ls** in the root directory (/), you should see something like this:

```
# cd /
# ls
.profile    dev         h
sa          unix        b
etc         lost+found  tmp
usr         bin         f
m           u
```

Depending on what software is loaded onto your system, your listing may or may not be identical to this. These files and directories, however, will be the basis of our discussion for this issue. First of all, a small lecture is in order.

The **root** and **manager** login accounts are intended to be used by the system administrator for system-related work and problems. *You should never do your daily computer activities (i.e. Fortune:Word, Multiplan, Business Accounting) while you are logged in as "root" or "manager".* Each system administrator should also have their own login account (i.e **/u/davek**) where this type of work should be done. There are a number of reasons for this:

1. Files may be created in places other than your login home directory. If for example, you login as manager and then select **Fortune:Word** from the Global Menu, your word processing files will be created in the root directory (/). These types of files should be in your login home directory. Part of what makes the system administrator's job easier is knowing where certain files and directories are. We should be able to go to the / directory and do an **ls** and, at a glance, tell if something is missing or if something is there that does not belong there. This is true of other system directories as well. When you did the **ls** command above, if you find files that have **.MP**, **.dc**, or **.fr** after the names, that means someone has been doing **Multiplan** or **Fortune:Word** work while they were logged in as root or manager. These files should be moved to the appropriate **/u** directory.

2. Whenever you are doing work as a normal user, you should be bound by whatever restrictions a normal user has. In other words, don't abuse the privileges that the system administrator has.

3. Files may be created with per-

UNISOFT INC.

Experts in Computer Consulting - Sales - Service

Presents

FORTUNE SYSTEMS HELP

We Provide:

- Expert Trouble-shooting
- Years of programming and accounting experience
- *Free Initial Consultation*
- Training Available for:
  * BASIC/IDOL/BAS and Multiplan
  * UNIX and Ext. Fortune:Word
- Hardware Repairs & Upgrades
- FORTUNE trained experts at a cost that will surprise you!
- *Trade-Ins Welcome*
- Specializing in:
  * Hard Disk Trade-In
  * Hard Disk Upgrades

Authorized Fortune Systems Repair Depot
Fortune Systems National Value Added Reseller (VAR)
CALL: (303) 671-8600 EMERGENCY HELP: (303) 649-8505

UNISOFT INC. 14100 East Jewell Avenue Suite 27 Aurora, Colorado 80012

missions that are inappropriate for the type of file you are creating. For example, if you login as root and create a word processing file in your normal login directory (/**u/davek**), that file will not have the same ownership and permissions set that it would have if you had created it while you were logged in as **davek**. In fact, if you created it as root and then login as **davek** to try to change it, you won't be able to.

4. If files and directories are created in places other than where they are normally created, you take the risk that they will not be backed up.

One of the responsibilities of the administrator is to make sure that files and directories on your system are where they are supposed to be and that the permissions on them are correct.

Let's continue on with our tour of your system. The first file in the / directory listed above that we will talk about is the **.profile** file. In the Bourne Shell (**sh**), if a **.profile** file exists in a user's home directory, the commands and instructions in that file are executed *for that user only* whenever that user logs in. This is different from the **/etc/profile** file which is executed for *all users* when they login. We will talk about the **/etc/profile** file later on. The **.profile** in the / directory is executed for root or manager whenever they login. You could have a different **.profile** for each user on the system which would be located in each of their home directories (i.e. /**u/davek**). The **.profile** file in the / directory is automatically created when a "cold boot" (when the system is initially installed or reinstalled) is done.

Here's what the /**.profile** file looks like on a For:Pro 2.0 system, assuming you have not changed it:

```
# cat /.profile
:
# .profile for root/manager
# logins, set defaults, etc here.
# default LANGUAGE is set by the
# kernel, default terminal type
# is set by login from /etc/ttytypes
```

This particular **.profile** just has comment lines in it (lines preceded with a **#** are comments). The comments tell us that this **.profile** is for the root or manager logins. It also indicates that you can set default parameters and variables here. To find out the default parameters and variables on your system, use the **set** command as shown in Figure 1.

```
# set
HOME=/
(Home directory is /)
IFS=
(Field separator is <CR>)

LANGUAGE=EN
(Language is English)
LPRFLAGS=-h
(No banner page for printer)
PATH=:/bin:/usr/bin:/usr/
ucb:/etc:  (Search path for
commands)
SHELL=/bin/sh
(Bourne Shell)
TERM=FT
(Fortune terminal)
USER=manager
(Login name is "manager")
```

Normally you will not want to change these default variables. However, if you did want to change them, the command lines would be in /**.profile** to change the parameters for root or they would be in a **.profile** file in a user's home directory to change them for that user. Remember that the **.profile** file should be in the user's home directory and only affects that user.

Let's talk about a couple of these defaults that you may be interested in changing. If we put the following line in a **.profile** then the banner page with the users name will be printed for any print job:

```
LPRFLAGS=-b
```

Have you ever logged into a system as a normal user and tried to execute a command that you know is on the system and received the message **Command not found**? Whenever you execute a For:Pro command, the system looks through a pre-selected list of directories for that command. This list is contained in the **PATH** variable listed above. For example, the following directories would be searched if root or manager executes a command:

current directory you are in
/bin
/usr/bin
/usr/ucb
/etc

They are also searched in the order that they are listed in the **PATH** variable. The directories that are searched are quite different for a normal user. If you login as a normal user and enter the **set** command, the **PATH** variable would look like this:

```
PATH=:/bin:/usr/bin:/usr/
ucb
```

Notice that unlike root or manager, the /**etc** directory is not searched. That is why when a normal user tries to execute the **shutdown** command, for example, he will get the message **command not found**. The **shutdown** command is located in the /**etc** directory which is not searched for a normal user. The reason the system is set up this way is that most of the commands in the /**etc** directory are system administrator commands and should not be executed by the normal user. To demonstrate how the **.profile** file works, we will show you how it can be changed. One way a normal user can execute these commands is to put in the full path name. For example, we could enter:

```
# /etc/shutdown
```

By putting in the full pathname, the **PATH** variable does not apply. If we always wanted a particular user to have the /**etc** directory searched, we would create a **.profile** file in his home directory and enter the **PATH** variable the same as it is for root or manager. Of course, we assume you know how to use one of the For:Pro editors (**vi**, **ed**, **screen**) to create or modify the **.profile** file. This is more frequently used when we want a particular directory

searched for a command. Some system administrators create their own directory of commands (c programs, shell scripts, etc.) which they want searched for each user. Let's say, for example, that we create a directory called **/usr/local** and put a shell script called **backup** in it. To have this directory searched for a particular user, we would create or modify that user's **.profile** file to read:

```
PATH=:/bin:/usr/bin:/usr/ucb:/usr/local
```

Here we have added **:/usr/local** to the end of the **PATH** variable which means that this directory will be searched whenever a user executes a command.

Enough about the **.profile** file. You can get more information about the **set** command, variables, and the **.profile** file by looking at the **sh** command in your For:Pro manuals.

In Part 3, we will continue by talking about the other files and directories in the / directory. □

Affordable laser printing for Fortune

Laser Printing! Kiss the expense good-bye.

QMS — KISS laser printer $1995.00

- cost per page 1.4¢.
- 9 resident fonts
- Print cap and print utility included.
- add-on sheet feeder provides total of 3 trays and envelope feeder.

THE GOOD FORTUNE TRADER

QMS - KISS from
The Good Fortune Trader
1 Year Warranty

a division of Computer Services of Atlanta

(404) 392-9938

1534 Dunwoody Village Parkway, Suite 109, Atlanta, Georgia 30338

We are the *largest* Authorized Fortune Master Dealer in the Pennsylvania and Delaware area. As part of the **P**rofessional **S**olutions **G**roup of companies with over 35 years in business automation, we have the *stability* that our customers demand

P S G

Our services include a factory trained field engineering team, a complete parts inventory including system upgrade capability and a software support department.

For all of your Fortune needs contact:

Mark Shields of PSG at
2550 Eisenhower Avenue
PO Box 762
Valley Forge, PA 19482

or you can call Mark at (215) 666-1400

*** See the NEW product DOS LINK ***

SOFTWARE FOR THE FORTUNE 32:16

NEW PRODUCT!!! DOS LINK ($495.00) - provides easy-to-use menus and programs that transfer data between any PC (including Laptops) running DOS and the Fortune system. Here are just a few examples of how this product can be used:

- Transfer Multiplan or Fortune:Word Documents from or to your DOS system.
- Transfer other ASCII or binary files.
- Use the PC or Laptop as a remote terminal on the Fortune system.
- Transfer THOROUGHBRED (SMC) BASIC data that has been extracted using one of our other products

Documentation and software for both the DOS and Fortune system are provided. Data may be transferred via a direct connection (9600 baud) between the two computers or via modem (at modem speed). You may also order additional hardware for your PC that may be required to accomodate the transfer.

If you are interested in purchasing a Laptop computer that can be used with this product for people on the road or at home, give us a call!! We also provide consulting services for interfacing your DOS system to the Fortune ($50/hr).

MULTIPLAN LINK - can pass data from BAS/IDOL/BASIC files to Multiplan and back. For example, it could be used to send data from any of the BAS files (Chart of Accounts, Customer Master, Inventory Master, etc). Allows selection of fields to be sent and provides logical retrieval and key range selection. **$295.00**

RECORDS PROCESSING LINK - can pass data from BAS/IDOL/BASIC files to Records Processing and back. For example, it could be used to take customer address data from the Customer Master and will automatically make the Records Processing List document for mail/merge. Allows selection of fields to be sent and provides logical retrieval and key range selection. $295.00

ASCII LINK - can pass data from BAS/IDOL/BASIC files to ASCII and back. Use to send data between machines or databases. For example, it could be used to send data from any of the BAS files (Customer Master, Inventory Master, etc). Allows selection of fields to be send and provides logical retrieval and key range selection. **$295.00**

TERMINAL/PRINTER LINK - provides capability to print data on a printer connected to a workstation. Can print BAS/IDOL/BASIC reports, Multiplan and Fortune:Word documents. BAS/IDOL reports can also be viewed before printing, converted to Fortune:Word format; accumulated for mass printing; backed up to diskette; printed on any system printer. An option to print BAS/IDOL reports on a LASER PRINTER is $50 extra. Many other options are included. **$195.00** (Fortune:Word printing on a printer connected to a workstation is $50 extra)

TELEPHONE LINK - on line telephone message, telephone directory and inter-office mailbox/menu system. Telephone message entry resembles typical pink slip. Telephone directory can search by name or company. Mailbox/message system can send and receive Fortune:Word documents or Multiplan documents. Does much more! Give us a call. Written in Business Basic. **$295.00**

CALCULATION LINK - provides programs for amortization, depreciation, loan repayment, averaging (with graph), linear correlation and breakeven analysis. Written in Business Basic. **$95.00**

KOMPACT PERSONNEL ACCOUNTING - provides for data capture of personal, wage/salary, job, education, salary/wage, appraisal training, dental, medical and life insurance data. Includes over 80 pre-defined reports/worksheets. Written in Business Basic/IDOL. **$995.00**

PRINTER LINK - provides printcaps for Okidata 84, 2410, TI 855, Anadex 6500 and Printronix. Includes shell script for changing pitch and LPI. Allows these printers to be used in Fortune:Word. **$95.00**

MODEM LINK - allows Hayes 1200/2400 modem to be used on one port for both incoming and outgoing calls. **$95.00**

WESTERN LINK - provides instruction on how to use your Fortune for sending/receiving telex messages using Western Union's Easylink. Can replace your telex! Fortune ITE or Handshake software required. **$95.00**

UNI-KOMP 9803 Rocktree Houston, Texas 77040 (713) 937-7782

A Verified Copy Command– making sure that there are no ERRORS in your files.



We had a problem with our hard disk recently. The way that we discovered this problem was quite disconcerting. We heard from some of you that our software diskettes weren't working properly. Puzzled, we looked more closely at our disks and, sure enough, we found that some of the files appeared to be randomly corrupted.

Since we use our trusty Fortune 32:16's floppy drive to copy the disks, our first thought was that it was becoming a bit flaky. Further tests, however, revealed that it wasn't the floppy drive but one of our hard disks that was acting schizophrenic.

We found that errors were sometimes produced when copying files from the bad disk to either another disk or the floppy drive. Ok, we found out why some of our software diskettes had problems. But now we were faced with the larger problem of how to get the information off this flaky disk.

The easiest solution would be to simply copy everything from this now flaky disk to another new disk. Ah, but there's the rub - the problem in the first place was that to copy files from this disk to another disk was inherently fraught with errors.

What we needed was a way to do a verified copy ... make a copy and then compare that copy to the original.

What we needed, we thought, was a way to do a *verified* copy. That is, we wanted to copy a file from the bad disk to the new disk. Once the copy was made, we then wanted to compare this copy with the original on the flaky disk to see if it was a good (uncorrupted) copy. If it was a good copy, then we would be set. If, on the other hand, it was a corrupted copy, then we would want to do the copy over and then do another comparison. This process would be repeated as many times as needed to insure that an uncorrupted copy of the file was obtained on the new disk.

We naturally turned to UNIX.

No single UNIX command will do exactly what we wanted. We looked at the **cp** command and found that it did not verify the copy it made. Yes, there is an option, **-X**, that when used with the **-B** option, will check to see if the media is readable after a disk is written. However, this is not really verification on a file-by-file basis because it is quite possible to have a disk that is readable and yet it will still have one or more files that are corrupted.

As far as we knew, there is no single UNIX command on the Fortune that would do what we wanted. However, we had confidence that we could use the tools that are a natural part of UNIX to construct something that would do what we wanted.

What we do in the remainder of this article is to present you with our version of a copy command that verifies, on a file-by-file basis, that the copy of a file is the same as the original.

Figure 1, on page 18, shows our UNIX shellscript which we will call **vcp** for **v**erified **cp**y. We have chosen to keep this script as simple as possible for two reasons. First,

pedagogically it will be easier to explain. Second, we generally agree with the underlying philosophy in UNIX that it is better to build tools that are simple. If more complex functions are required, simple tools can be put together to perform this more complex task.

The shellscript is simple. It accepts two arguments. The first argument is the name of the file that you wish to copy. The second argument is the name of the file to which you wish the copy to go. As an example, if I wanted to copy the file called **fubar** to the floppy drive and I was in the directory that **fubar** was in, I would type:

```
$ vcp fubar /f/fubar
```

Notice that I could not simply type `vcp fubar /f`, as we do with the **cp** command, because **vcp** wants two arguments that are file names. **/f** by itself is the name of a directory and this would cause our shell script to print an error message and then exit.

Look again at Figure 1. The very first thing we do is check to make sure that the number of arguments typed in is correct. The **case** statement does this. The **$#** represents the number of arguments given to **vcp**. If that number is 2, then the **case** statement is left in a normal fashion and we continue with the line after the **esac**. If that number is anything but a 2 (which is what ***)** matches), then an appropriate error message is printed and the shellscript is exited.

Next, we do some preliminary checks on the arguments. The first **if** statement, the one that says `if test ! -f $1`, checks to see if the first argument (**$1**) is not a file (**test ! -f**). If the first argument is not a file, then we print out an error message and exit. This test safeguards against putting in the name of a file that doesn't exist or putting in the name of a directory. In either of these cases, our command should fail.

The second **if** statement checks to make sure that the destination file is not a directory. If we didn't check for this possibility, our **cp** command would work just fine. In other words, it would copy the file to that directory. However, the verifying loop that uses **cmp** would do something strange; namely, it would try to compare the source file with the directory, rather than the copy of the file within the directory. The **cmp** would always fail and our script would loop infinitely.

You might ask, shouldn't you also check to make sure that the destination does not yet exist? Shouldn't we safeguard against inadvertently overwriting an already existing file? The answer is yes, we should. However, this safeguard is accomplished with the **-i** flag on the first **cp** command in our script.

NJBC

New Jersey Business Computers

Authorized Fortune Systems Master Dealer

Authorized Fortune Service

Authorized Fortune Training Center

NEW JERSEY BUSINESS COMPUTERS has the Largest Affiliate Dealer Network in the New York and New Jersey Metropolitan Area.

For The Most Reasonably Priced TRAINING, SERVICE and SUPPORT Available.

301 Route 17 North, Rutherford, New Jersey, 07070

call

201-438-1410

Next, we set the value of a variable called **ntries** to 1. This variable simply keeps a count of the number of times we have to recopy the file. It is not really essential to the working of the script. We felt it was useful to print out on the screen the number of times it required to recopy a file. Thus, a quick look at the terminal might tell you of some problem. For example, if it is taking hundreds of tries to recopy a file, you might be justified in suspecting a more serious problem.

Now, we do the first copy. This **cp** command will make a copy of the source file to the destination file. As previously mentioned, the **-i** flag will cause the **cp** command to prompt the user if the destination file already exists. This prevents the accidental erasure of an already existing file.

After the **cp** command we come to the interesting part of our shell script. We have programmed something called an **until** loop. The **until** loop has the form:

```
until command
do
        statements
done
```

The statements between the **do** and the **done** are executed as long as the command exits *unsuccessfully*. When the command exits *successfully*, control is sent to the first statement after the done. In our shellscript, there is nothing after the done statement; thus, our shellscript finishes.

The command that we execute is our compare command (**cmp**). The UNIX command **cmp** will compare two files. If the files are identical, then the command ends "successfully". If the files are not identical, **cmp** will print out the byte and line number at which the difference occurred. When a difference is found, **cmp** exits "unsuccessfully".

Thus, the guts of our **until** loop will be executed as long as there is a difference between the copies of our files. As soon as we get an uncorrupted copy, the **cmp** will exit "successfully," and our shellscript will finish.

The guts of the until loop are also straightforward. Keep in mind that the only way any of these commands get executed is if the **cmp** *fails*. If the **cmp** fails, we print a message announcing that we are recopying the file. Then we use the **cp** command again to make the copy. Note that this time we do not use the **-i** flag. This is because we already know that our destination file exists! However, it is a corrupted version, so we don't care if it gets overwritten. After the copy we add 1 to our variable ntries. (For more information on the **expr** command, see the UNIX Directory in Volume 3 Number 11 of **/u/fortune** *news*).

That's basically it. Our source file gets copied to the destination that we desire. Once it is there, we use the **cmp** command to make sure that it is uncorrupted. If it is corrupted, we simply copy and check it again, and again, and again, until it gets to the destination in an uncorrupted fashion.

And Now For The Rest of the Story...

The **vcp** shellscript we have presented thus far is a self contained program which will perform the duty we have described: copy a file from one place to another and make sure it gets there accurately.

The problem we had with our hard disk that prompted this program involved another step in its solution. We wanted to copy *all* the files from this flaky hard disk to another hard disk. Therefore, we needed a way to feed the names of the files from this disk to our **vcp** program so that

Figure 1.

```
:
case $# in
      2) ;;
      *) echo "usage: vcp source_file destination_file"
      exit;;
esac

if test ! -f $1
then
      echo $1 "is not a file. vcp can only be used on
single files"
      exit
fi

if test -d $2
then
      echo $2 "is a directory. vcp can only be used on
single files"
      exit
fi

ntries=1
cp -i $1 $2

until cmp $1 $2
do
      echo "^GRecopying..." $ntries
      cp $1 $2
      ntries=`expr $ntries + 1`
done
```

WE wouldn't have to type **ALL** the names singly!

The shellscript that does this task for us is presented in Figure 2, which we call disk_vcp.

This script takes two arguments, the first is the source disk and the second is the destination disk. For example, if we wanted to copy all the files from our root disk to a disk we call **h1**, we would type:

```
$ disk_vcp / /h1
```

In principle, the script is comprised of three steps. First, we create a list of all the files on the source hard disk. Second, we modify that list so that it is composed of two columns. The first column would be the name of the source file and the second column would be the name of the destination file. Third, feed this list, which is composed of two columns, to the **vcp** command that we described above.

We will now talk about each of these steps in a bit more detail. The first step of creating a list of names is accomplished with two **find** commands. The first **find** command will list out the names of all the files that are on the source disk, with the exception of files that begin with a period. Hence, we need the second **find** command. Thus, after these two commands are performed, the file **/tmp/tmp.lst** will contain the names of all the files on the source disk.

The second step is to create a file with two columns. The first column will contain the names of the files on the source disk including the disk designation. The second column will contain the names of the files on the destination disk. This will be an exact copy of the first column, except for the name of the disk. For example,

```
/u/john/junk  /h1/u/john/junk
/u/john/cal   /h1/u/john/cal
/u/john/misc  /h1/u/john/misc
```

This process, of creating two columns, is accomplished by using **sed**. **Sed** is a UNIX command which will edit "on the fly," so to speak. (**Sed** is a part of the **Development Utilities**. You could do a similar thing with ed without too much trouble if you don't have Fortune's **Development Utilities**). The command that begins:

```
sed "s/\\$1/\\$2\//" ...
```

is the **sed** command that will change the source disk designation to the destination disk designation. We put this new listing in a new temporary file called **sedtmp.lst**. We then *paste* together our two separate columns so that we end up with a file that has the two columns that we need. (The UNIX command **paste** is available on our **UNIX Tools** diskette, along with

Figure 2.

```
:
case $# in
      2) ;;
      *) echo "usage: disk_vcp source_disk
destination_disk"
      exit;;
esac

#
# get list of names of all files on source_disk
#
echo "Creating list of filenames..."
find /$1 -name "*" -print > /tmp/tmp.lst
find /$1 -name ".*" -print >> /tmp/tmp.lst

#
# change the source disk designation to the destination
# disk designation and then create a tmp file with
# two columns
#
echo "Modifying the list..."
sed "s/\\$1/\\$2\//" < /tmp/tmp.lst > /tmp/sedtmp.lst
/h1/unixtools/bin/paste /tmp/tmp.lst /tmp/sedtmp.lst > /
tmp/2col.lst
sed "1,1d" < /tmp/2col.lst > /tmp/2col2.lst

#
# feed the file names to vcp
#
echo "Copying the files from $1 to $2..."
echo
IFS='
'                    # set IFS to just newline
for i in `cat /tmp/2col2.lst`
do
IFS=' '   # set IFS to space and TAB
set `echo $i`
echo "vcp $1 $2"
#vcp $1 $2
IFS='
'                    # set IFS back to just newline
done
rm /tmp/tmp.lst /tmp/sedtmp.lst /tmp/2col.lst /tmp/
2col2.lst
```

many other useful UNIX utilities and games.) A second **sed** command is performed that deletes the first line of this new file because this line is unnecessary. Thus, our two-column list ends up in the new file called **/tmp/2col2.lst.**

The next part of the script handles feeding this two column file to the **vcp** command so that each file in turn can be copied from the source disk to the destination disk and then verified. The real trick in this portion of the script is the use of the **IFS** shell variable. **IFS** stands for Internal Field Separator. The **IFS** controls how the shell splits things into fields, or words. Usually, the **IFS** is set to a space, a TAB and a RETURN.

For our application, we want the **IFS** to be just a RETURN for part of the time and a space or TAB for the rest of the time. We first want it to be just a RETURN because we want to grab an entire line from the **2col2.lst** file. After we get the line, we want to process the two separate elements, so we set the **IFS** to a space and a TAB. Then we use the **set** command to force the two filenames we've just grabbed to be considered arguments which we then feed to the **vcp** command. After the **vcp** command, we set the **IFS** back to just a RETURN in preparation for the next iteration of the for loop that will grab the next line from the **2col2.lst** file.

The final **rm** command simply cleans up our temporary files.

We were successful in transferring our files from a flaky hard disk to a better disk while ensuring that we had uncorrupted files. By using UNIX and shellscripts, we were able to automate this process quite a bit. In the process, we put together a new UNIX tool, which we call **vcp**. It is useful for creating copies of files that we can be positive have no errors in them.

Conclusion

We certainly invite any readers to respond to this article and offer suggestions for improvements on our scripts. Also, the scripts have worked for us during the limited time that we've used them, but perhaps you see a bug or can conceive of a situation in which the scripts wouldn't work. Finally, perhaps you've already solved this problem in a slightly different way. We would be interested in publishing your version. Write us and tell us what you think. □

Mark Palmerino

FORTUNE SYSTEMS
an SCI Systems Company

No matter what your product or service, staying competitive requires control

Let QBC, your company's one source computer supplier, provide your business with solutions to:

Order Entry	Payroll
Accounts Payable	General Ledger
Accounts Receivable	Sales Analysis
Invoicing	Legal Time Billing
Inventory Control	Manufacturing Software
Office Automation	(MRP II Plus)

QBC is your one source for hardware, software, maintenance, customer training and data conversion.

MASTER DISTRIBUTOR FOR UPSTATE NEW YORK AND ERIE, PENNSYLVANIA

QBC

QBC Systems Inc.
375 NORTH FRENCH ROAD, AMHERST, NEW YORK 14120
(716) 691-5201 **1-800-443-5770**

Expand your Fortune® without spending a fortune.

New equipment costs big bucks. The Good Fortune Trader sells fully tested burned-in equipment, refurbished to Fortune® standards at prices that won't make waves with your cash flow. Call or write today for a listing of our specials.

THE GOOD FORTUNE TRADER

60 DAY WARRANTY
a division of Computer Services of Atlanta
(404) 392-9938
1534 Dunwoody Village Parkway/Atlanta, Georgia 30338

BASIC

continued from page 1

IDOL report maintain running totals? We have a daily inventory report which uses the product codes to show the dollar amounts entered and shipped in three broad categories. I would like to maintain a running total of each category for the month, setting it back to zero after month's end. Which brings me to another question: can an IDOL report be written so that it is interactive? What I mean is, can it ask "Is this month's end?" and, if a positive response is given, reset the monthly totals to zero?

Answer: Yes and yes, provided you do not mind writing some BASIC code.

I am going to assume that you need to capture three report totals (as opposed to item totals or report subtotals). I will also assume that you want the totals from each daily report to add into the saved totals from all previous daily reports. This requires a new file with a single record or a new single record in an existing file. I love to live dangerously, so let's play with **CCNVZ**.

CCNVZ is a single control file which is subdivided by IDOL into "mini" files for the purpose of conducting multiple file maintenance operations. IDOL does this by recognizing only the first five characters of the file name when defining or opening a channel to the file. This means that we can add a character to a five-character file name to create a new file maintenance screen for that file. We'll see how this works in a moment.

First, we need to find a CCNVZ file name and number which is not currently in use on your system. I suggest you try CCNVZ%. From any BAS or IDOL selector, type in DFN (which means Display File Numbers) and <CR> (carriage return). At the "Enter Starting File Number" question, press <CR>. Go through all pages of the display looking for a free file number (for example, number 25) and the file name CCNVZ%. If you find CCNVZ% you must come up with another name, such as CCNVZ* or CCNVZW, and go through the display again.

We will now assume that both CCNVZ% and file number 25 are available for use. Now we must IDOL-Define file number 25.

Go back to the selector and enter BASIC <CR>. At the BASIC prompt, type in OPEN (1) "CCNVZ" <CR>. At the prompt, type READ (1,KEY="%") <CR>. If you get an error 11, you are safe. (Otherwise type READ (1,KEY="*") <CR>. Keep changing the character in the quotes until you get the error 11. Make a note of the first character that gave you the error. We will call this character the "error 11 key".) Type RUN "DOL" to exit the console mode.

At the selector, enter 798 <CR>. This will bring you to the DEFINE A RECORD FORMAT utility. Enter 1 <CR> to ADD, and 25 <CR> (or whatever free number you found in the DFN screens) for the FILE NUMBER. Then use the following responses for the remainder of the header information:

1. FILE NAME:
enter CCNVZ% (or substitute name).
3. ELM DESC LN:
enter 15.
5. NO FM COLUMNS:
enter 1
COL POS: 6.
enter 25
COL LN: 7.
enter 3
COL LEN: 8.
enter 4
COL POS: 9.
enter 20 and press <F3> to skip
20. SCREEN HEADING:
enter INVENTORY TOTALS FILE
21. PASSWORD:
enter 26 and press <F3>
26. KEY DESCRIPTION:
enter KEY = "%" (or the error 11 key)
27. USER ID:
press <CR>

The computer says, ADD (Y/N) OR ENTER FIELD NUMBER AND RETURN. Look over the screen and, if everything looks good, press <F1> to continue.

The screen will now display the Add Detail screen and the cursor will be under the 1. ELEMENT NAME field. Enter GLCTL REC KEY and <CR>. If this element is in your data element dictionary, values will be printed on the screen for some of the fields numbered 2 through 20, and the computer will say, ADD (Y/N) OR ENTER FIELD NUMBER AND RETURN. If the element is not in the dictionary, the cursor will be under field 2, EL LN. In this case you will have to add all of the values (pressing <CR> for field numbers we will not be using). Here are the values for the fields we need:

2 EL LN:1
3 KI: A
4 ET: C
6 PS: A
7 SC: C
15 PRE-SET VALUE: % (or your error 11 key)
17 VARIABLE NAME: AN$

If you are in the change mode and find values in fields other than the ones above, change those fields to a single space.

Once you have told the computer to add this element, you will be back to the 1. ELEMENT NAME field. We need to add three more elements. I will call them TOTAL 1, TOTAL 2 and TOTAL 3. You may wish to call them something else, but if you do, be sure to keep the descriptions contained within the 15 character limit. The other fields needed for these elements are:

2 EL LN: 11 (to allow up to $99,999,999.00)
8 PR: 2
17 VARIABLE NAME: AN (for TOTAL 1), BN (for 2), CN (for 3)

From the 1. ELEMENT NAME field, press <CR> and <F4> to exit. You do not need to create the file because we have added a sixth character to the name of the existing file CCNVZ. Nevertheless, you should check CCNVZ with the BASIC Utilities (UTL <CR> from any selector), 6 *FPSD - DISPLAY FILE PARAMETERS, to insure the file is not full. (If it is, use 17 *TPSD - FILE TRANSFER AND EXPAND to increase the number of records in CCNVZ.)

From the BASIC utilities, select number 13, DEFINE A FILE DATA AREA, and enter P for a program. Disk number is 0 (zero), number of pages is 1 and file name is UPDATE. (I'm sure that you can find an appropriate program name to use if "UPDATE" is already defined on your computer. Just be sure to use exactly six characters for your program name.)

Exit to your BAS selector and define your IDOL report as usual until you get to DEFINE END OF REPORT LOGIC. (Since you are a reader of **/u/fortune** *news* and know how to define IDOL reports, I am skipping the basics.) Add the following line:

```
5500 LET A5=A(J+1),B5=A(J+2),C5=A(J+3)
```

Be sure to Save Report Parameters because we will need to run the report from a menu.

Now our holding file is ready, our update program is defined, our IDOL report is coded, and it's time to write our program. From a BAS selector, type BASIC <CR> to go to the console mode. See Figure 1 for the coding to enter (be sure to hit <CR> after each line).

Type SAVE "UPDATE" (or your program name) and <CR>. Note that we didn't have to open a channel for our update because channel 7 is always opened to CCNVZ in IDOL reports. (Wasn't that clever of us?) After your report has printed you will get the question "END OF THE MONTH (Y/N)". If you answer N or press <F2>, the report totals will be added to your CCNVZ record. A response of Y or <F1> will cause the three totals fields in the file to be reset to zero, and a response of END or <F4> will cause all updating of the record to be bypassed. (By the way, please don't criticize my error routine, I am trying to be brief.)

Type RUN "DOL" <CR> to return to the selector, and find a selector number for the new report. (The selector number is in the upper left corner of each menu screen.) I will explain how to add the report to the FORTUNE SYSTEMS BUSINESS APPLICATIONS selector (number 10) after 8. BUSINESS UTILITIES.

From your chosen selector, enter 799 (for the Selector Dictionary Records) and <CR>. Enter 4, for Inquire, and <CR>. The INDEX to enter is 10 (or your chosen selector number). Answer No to the hard copy question and, at the INDEX field type *8 (or * + the number of the selection just before where your report will be) and <CR>. Answer No again for the hard copy question and <CR> for the INDEX. At the Option line enter 1, for ADD, and <CR>. At INDEX press <F2> for next record. A number will be printed next to INDEX and you need to make the following field entries (skip missing numbers with the <CR>):

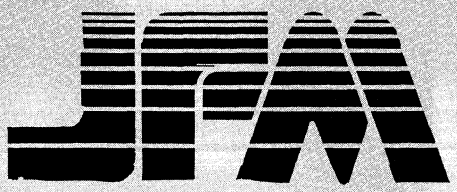
6. PRINTER IND: 1
9. OPEN FILE IND: R
10. SELECTION DESC: INVENTORY UPDATE REPORT
12. FILES TO OPEN: 095 R095XXUPDATE
13. PROG TO RUN: CUTFA0
15. PASS PARM: AP

Field 10, SELECTION DESC, is what will appear on your selector as the new selection description. You may not wish to call it Inventory Update Report. Field 12, FILES TO OPEN, is critical. There must be exactly 3 spaces between 095 and R095XX and no spaces before UPDATE. For the XX in R095XX substitute your defined report name, and if your program is not named UPDATE, substitute your six-character program name. By entering AP for field 15, PASS PARM, we are telling the report to go to the printer. If you wish to see it on the terminal, enter a space for 6. PRINTER IND and AT for field 15.

Once you have entered this data, answer Yes to the ADD question, No to the Hard Copy question, <CR> at INDEX and <F4> at the Option line to exit the utility. The next time you access the selector, IDOL will have to redefine it. After that you will be able to run your report from the menu and you will have your totals file. □

Figure 1.

```
0010 REM "UPDATE - ADJUST TOTALS"
1000 READ (7,KEY="%",ERR=8000)*,A4,B4,C4
1010  INPUT  (0,ERR=1010)'CS',@(10,10),"END  OF  MONTH  (Y/N):
",X7$:
      ("")=1030,"Y"=1040,"N"=1060,"END"=9000,LEN=0,3)
1020 GOTO 1010
1030 ONCTL GOTO 1010,1040,1060,1010,9000,1010
1040 LET A4=0,B4=0,C4=0
1050 GOTO 1070
1060 LET A4=A4+A5,B4=B4+B5,C4=C4+C5
1070 WRITE (7,KEY="%",ERR=8000)"%",A4,B4,C4
1080 GOTO 9000
8000 PRINT @(0,22),'CL','RB',"ERR ",ERR," (HIT 'RETURN'): ",
8010  INPUT *
8020  IF CTL=4 THEN GOTO 9000
8030 PRINT @(0,22),'CL'
8040 RETRY
9500 RUN "CUTSA0"
```



Dear Fortune Owner:

As you have probably read in */u/fortune news,* Fortune Systems SX models are significantly more powerful and faster than the PS and XP models. If you have a PS or XP model system, you should take advantage of our special promotion on SX upgrades. For a limited time, and while our supplies last, we will upgrade your system, including the new SX Cabinet at no additional charge for the cabinet. With this special offer, you may choose any combination of the following upgrades and/or expansions:

- *SX Processor with Current Release of Multi-User For:Pro*
- *60 MB Internal Streaming Tape*
- *30 MB, 45 MB, 70MB High Speed Hard Disk*
- *6-Port Intelligent Asynchronous Controller*
- *Upgrade to Current Release of all Fortune Software*
- *Fortune:Windows (Included at NO Charge)*

Additionally, we can perform these upgrades within 24 hours of receiving your system. In fact, if you ship your system overnight on a Friday for Saturday delivery, we will have it back to you on Monday morning, all ready to plug in and run; not losing the use of your system for a single work day. And, we will issue you a full 90-day warranty on the entire system.

JFM provides services to Fortune Systems owners either directly or through our authorized Affiliated Dealers. If you haven't used our services, I invite you to discover JFM; you will be glad you did.

Sincerely,



Morris Frydman
President

Talk to us, then talk to our clients



JFM BUSINESS SYSTEMS
The Merchandise Mart
Lobby Suite 144
Chicago, Illinois 60654
Call (312) 644-0870

Authorized Fortune Systems Master Dealer

SUPPLEMENT TO ALL VERSIONS OF BAS AT A REASONABLE PRICE, TOO!

JOB COSTING

The Right Tool for All Trades

BaSiC SOFTWARE CORPORATION

For More Information Call or Write...
BaSiC Software Corp.
5201 Powhatan Street
Baltimore, MD 21207
301-448-9460

Available on:
IBM AT & compatibles
UNISYS/SPERRY
Altos, All Levels

DO YOU KNOW...

there is an organization that has a thorough knowledge of

FORTUNE BAS
MULTIPLAN IDOL

We have been providing widespread support to our Fortune clients for more than 4 years.

If you need:

- training
- support
- maximum computer performance
- peace of mind relating to processing
- customization
- application results
- minimum errors and confusion
- better ideas on how to increase the power of your Fortune

Call Steve Rosenfeld at (314) 731-3636

SUPERIOR COMPUTER SYSTEMS, INC.
917 N Hwy 67, Suite 105
P.O. Drawer T
Florissant, MO 63031

Making computers work for you

/u/fortune news

73 Jason Street
Arlington, MA 02174

Address Correction Requested.

/u/fortune* news

The Newsletter for Users of Fortune/SCI Computers

July 1987/Volume 4 Number 7

F:W 3.0 Includes Glossary Disk

The Glossary Entry

Making Your Fortune Do the Work

Once again it has come time for us to extol the virtues of **Fortune:Word Glossaries**. If you don't use glossaries now, and you happen to have a copy of **Fortune:Word 3.0**, you'll find it's easier than ever to get in the swing of things because Fortune has included a floppy disk which contains all of the examples that are used in the **Fortune Glossary User's Guide**. Without doing any typing, you'll find some ready to use entries. (If you don't have **Fortune:Word 3.0**, don't worry, you can still use the entries in this article, you'll just have to do a little typing.)

For the truly uninitiated, we will explain briefly why you might want to use a glossary entry. The glossary functions simplify tasks that involve repeated keystrokes by automatically executing a series of keystrokes or commands, just as if you had typed them in from the keyboard. For instance, you can have an entry to automatically underline or bold a word, sentence or paragraph. Or you might want an entry which inserts the name of your company, or even your own name so that it doesn't have to be retyped every time it's used. Glossaries are useful for setting up default formats for letters, memos, reports, etc. They are invaluable for entering information into a **Records Processing List Document**. Having made our point, hopefully, we will draw your attention to some examples provided in the User's Guide.

Please note that glossary entries can be very simple, or very complex. Simple entries can be created and used by anyone. These entries insert text into your work automatically, such as you company name. The easiest way to create these entries is "*by example*", which basically means that you have the computer copy every keystroke that you make. More complicated entries can involve elaborate cursor movement, operator queries and input, decision-making capabilities, etc. We have always found that

> Fortune has included a floppy disk which contains all of the examples that are used in the **Fortune Glossary User's Guide.**

one of the easiest ways to learn glossary functions is to start with a glossary that works, and then modify pieces of it to suit your needs. That's why we're presenting these entries.

Using Fortune's Glossary Disk

As mentioned above, Fortune has provided a disk filled with the glossaries used in their Glossary Manual. There are four separate glossaries on the disk, **gloss1**, **gloss2a**, **gloss2b** and **gloss3**. In Chapter 13 of the manual, it lists where each of the examples is contained. To use these examples, choose **Retrieve from Archive** from the **Filing Menu** of **Fortune:Word**. Insert the examples

See **Glossary**, page 18

News From Fortune

Frank Watts hired as Vice President of Sales

Fortune Systems/SCI has recently announced that **Frank Watts** has joined the Fortune/SCI marketing team as Vice President of Sales, replacing Paul Olin.

For the past two and a half years Mr. Watts has owned his own consulting company, **F. Watts & Associates**. His impressive list of hi-tech clients includes such companies as Wang, Apple, Honeywell, Point Four, Pertec, and a variety of different VARs and dealers. Another distinguished member of that list is Fortune Systems. Frank has conducted several semi-

See **News**, page 2

In This Issue...

New Software and New Vice President–Read about it in **News From Fortune...Page1**

UNIX mail- how to use this effective means of sending messages to other users of the Fortune...**Page 4**.

More on system files in our series on System Administration...**Page 9**.

Conferences and string lengths in the BASIC Advisor...**Page14.**

/u/help–grep, appt, lpr and system hangups...**Page19**.

Overview of **FOR:PRO 2.0**...**Page21**.

*/u/fortune *news* is published by The Cambridge Consortium, Inc. It is not affiliated with Fortune Systems/SCI Technology, Inc..

/u/fortune *news*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz

Technical Editor
Scott Bradner

Advertisting Manager
Mark Palmerino

/u/fortune *news* is published 12 times per year by The Cambridge Consortium, Inc. Neither **/u/fortune** *news* nor The Cambridge Consortium are affiliated with Fortune Systems/SCI Technology, Inc.

Subscriptions cost $48 for one year within the U.S.A. For foreign subscriptions, add $27 each year for added postage and handling costs.

Back issues are offered at:

Vol 1, Numbers 1 - 3	$1/each
Vol 1, Number 4	$.50/each
Vol 1, Numbers 5 - 8	$3/each
Vol 2, Numbers 1 - 7	$3/each
Vol 3, Numbers 1 - 12	$4/each
Vol 4, Numbers 1 -	$4/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $10 each. Please add $5 per order for foreign destinations.

Although every attempt is made to insure that our contents are error-free and without risk, The Cambridge Consortium is not responsible for losses that occur due to correct or incorrect interpretation of **/u/fortune** *news*.

For more information, call or write:

/u/fortune *news*
73 Jason Street
Arlington, MA 02174
(617) 663-7700, (617) 646-5603

The text for this newsletter was prepared using a Fortune 32:16 for text input. The information was then transferred to an Apple Macintosh Plus with kermit and formatted for typesetting using the Aldus Pagemaker program.

UNIX™ is a trademark of Bell Laboratories, Inc. Fortune 32:16™, Fortune:Word™ and FOR:PRO™ are trademarks of Fortune Systems/SCI Technology, Inc. Multiplan™ and MS-DOS™ are trademarks of Microsoft. IBM-PC™ is a trademark of IBM. Macintosh Plus™ is a trademark of Apple Computer. Pagemaker™ is a trademark of Aldus Corporation.

©1987 by the Cambridge Consortium, Inc. 7/87

News

Continued from page 1

nars on behalf of Fortune which were presented to Fortune's internal sales force as well as to the Master Dealers.

Prior to owning his consulting firm Watts served for two years as Vice President of U.S. Sales at MAI Basic Four, Inc. where he was in charge of vertical market sales through a direct sales organization of 220 plus 30 exclusive dealers and 100 resellers. He was drafted by Basic Four after acting as a private consultant for them, following a two year stint as VP of Sales and Marketing for a $25 million computer reseller and consulting service called System House. His earlier experience includes eight years serving in various sales management positions for Wang Laboratories, including four years as Country Manager for Wang Canada. Watts is a graduate of Syracuse University where he majored in Business Administration.

Fortune hopes to capitalize on Watts' sales and consulting expertise in order to improve their distribution channel. He has been very successful in helping dealers become much more productive in the past, and he hopes to repeat that feat with Fortune. Of course in order for that to happen, two things will have to occur: 1) we'll have to buy more, and hopefully there will be an incentive to do that; and 2) there will need to be increased sales to new Fortune Users. We wish Mr. Watts luck in achieving his goals, and look forward to incentives to buy.

Software News

Fortune/SCI has announced the release of **35** new software packages. It has also announced that it is discontinuing MicroFocus COBOL and graphics related products, due to lack of demand.

In upcoming issues, we will be publishing a new edition of the index of software for Fortune/SCI equipment and a review of **Timemaker**, a time management software package designed to help professionals schedule their calendar efficiently and simplify routine office tasks. It consists of a set of office productivity tools including: an integrated Calendar, Clock, and Daily Appointment Scheduler, a Calculator and Adding Machine, interfaces that allow quick access to other applications software, and much more. This and all other software we mention are available for both the 32:16 and the Formula Series.

Fortune:Link PC for the IBM PC and compatibles is now available, supplementing the network offered by Fortune:Link DFS, a local area network software system that provides a fully distributed peer to peer network environment. Fortune:Link PC includes a network card as well as software for the PC. It adds network communications functions to the MS DOS operating system. It allows users to access files directly without having to identify the physical location of the file. To the user (and MS DOS), there is no apparent difference betwween reading (or writing) a file contained on a local disk and reading (or writing) a file on a disk of a remote system (Fortune 32:16, Fortune Formula, or IBM PC). All remote systems are accessible through a pseudo device, Drive Z, and network files appear to be members of their own hierarchical system.

Fortune:Link PC also allows users to transparently access other hardware devices (e.g. printers, plotters, modems) on other IBM PC systems within the network. These, too, are accessible through the pseudo device, Drive Z. This software also lets the user of an IBM PC control read/write access to specific files and devices within their own computer. The network administrator controls overall access within the network. While accessing FOR:PRO systems Fortune:Link PC users are also bound by its security and data integrity features (i.e. password access control).

Fortune/SCI also announced the availability of **3780 Remote Job Entry (RJE)** for the Formula. This is new enhanced version that recognizes 3780 Print Control Characters which

provides correct formatting on the 32:16. A new release is also available for 32:16 users, although it is necessary to purchase a complete package at this time (there isn't an upgrade available). It requires the installation of at least one Intelligent Communications Controller (CommB). The controller may also be used with other communication products but not concurrently.

Here are some of the other new packages that are available.

Super-Script Pharmacy Software, from WC Computer Services, maintains a database of information important to a pharmacy. Each screen is divided into four windows covering patient profile/doctor, patient prescription history, labels, and a place for making notes about the patient.

Police Package, from Howe Systems, Inc., computerizes the operations of a police department with a database system which tracks the mountains of report criteria on suspect description, arrests, missing persons, complaints, ticket processing, juvenile arrests, and vehicle maintenance.

Grafsman, by Southwind Software, Inc., is a new business presentation graphics package with an interface to word processing applications. It makes it possible to create within Fortune:Word 3.0 a wide variety of chart types and as much text on a chart as you need.

Front Desk/Message, from Spectra Data Systems, provides an easy to use system for automatically logging phone messages and tracking employees on an In-Out Board.

Arrow Office Automation (ARROW), from Arrow Laboratories, Inc., is a word processing and database package for the professional writer with a 90,000 word proofreading dictionary, a 5,000 word correcting dictionary, and multiple database features.

CTD, the Master Dealer in Michigan, has four new packages: **VoiceComm**, for use with a voice activated communication system; **CTD REtail Data System**, a point of sale system; **CTD MAFSYS**, which helps the manufacturing manager operate the shop floor more efficiently; and **CTD Shop Analyst**, for the build-to-order environment.

Placement Power, from Resource Control Systems Corp., is a software package for the personnel and recruiting industry. It stores and retrieves candidate resumes by any combination of number of important criteria. It also allows the user to search for resumes with specific comments.

Why Didn't We Think of That?

Fortune/SCI's new advertising campaign in Computer Reseller's News recently featured this ad



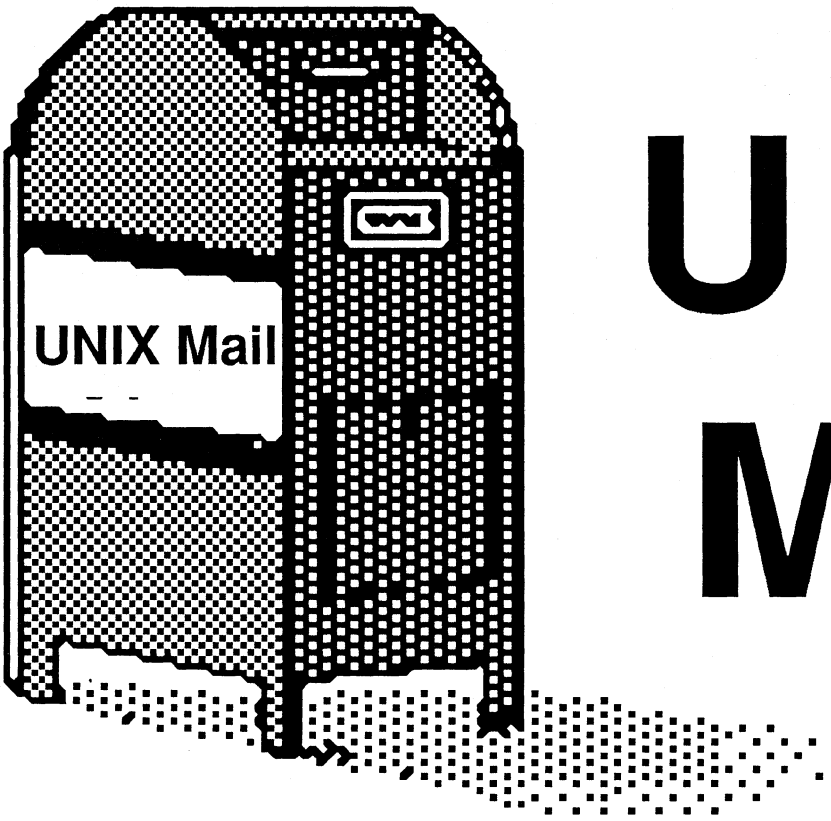
Software Discounts for Formula

In order to make the transition of purchasing a Formula smoother, Fortune/SCI has created an office automation bundle. You can purchase Fortune:Word, Fortune:Windows, and Multiplan, together, for a savings of 40% off the suggested retail price. This offer is meant to help promote the Formula for 32:16 users. This offer, dated July 10, is available for 90 days.

Bug Status Report

All SCSI based systems (mostly Formula's) with tape drive will not properly make a multi-volume tape backup set.

This will be corrected in the next release of For:Pro. Do not use multi-volume tape sets with Tape Streamer 1.3 and For:Pro 2.0 or 3.0. Instead, specify sections of the file systems which will fit onto one tape. The df and/or du -s commands may be helpful to determine this, e.g. du -s /u will provide the number of blocks that are used by the /u directories.□



UNIX Mail

How would you like to use a UNIX program that is extremely versatile? As easily as you can use it to send information to other users on your system, you can use it as a simple reminder system. The UNIX utility, **mail**, is all this and more!

Fortune/SCI has recently released a new version of FOR:PRO, the operating system for their line of computers. Some of the new capabilities and differences from the previous version of FOR:PRO are outlined in an article elsewhere in this issue. This month, in this UNIX Basics column, we discuss in more detail a new inclusion: UNIX mail.

Previously, **mail** was not included with the operating system. In order to have an electronic mail system, an owner of the Fortune had to buy the Development Utilities package. We are happy that **mail** is included in FOR:PRO 2.0, as it is a natural and useful adjunct to a multi-user UNIX system.

Easy to Use!

The main purpose of **mail** is to send information to one or more users of your Fortune computer. This can be a very convenient way of getting messages from one place to another. It should be pointed out that **mail** is a natural by-product of a multi-user system. As you know, more than one person can use your Fortune simultaneously. It is not unusual to have three, four, and many more people logged into a Fortune at the same time. Also, the same system often has dozens of "accounts" on it, one account for each person that might use the computer.

Thus, with so many people potentially using the same computer, there is often the need for you to pass some information on to another user of the computer. One example comes to mind: phone messages. It is not unusual here at **/u/fortune *news*** for me to get phone calls for other members of our staff. Let's say my phone rings and the person on the other end wants to talk to Bill. Bill, unfortunately, is not in at the moment, so I offer to take a message. If I know that Bill regularly logs in to the computer that I use, I might opt to send him the message via electronic mail.

To do so, I might type the following at the UNIX prompt:

```
$ mail bill <CR>
Bill:  Sue called and wants to talk to
you about the meeting next week - you
can reach her at 555-1589.
Mark
^D
$
```

Let's dissect this example a bit. First notice the line that starts with the dollar sign (**$**), `$ mail bill`. You will recognize the dollar sign as the UNIX prompt. The **mail** starts the UNIX mail program running and the **bill** tells **mail** who to send the message to, in this case, it is going to **bill**. (Bill has an account on our system which coincidentally is his name).

After I hit a CARRIAGE RETURN (**<CR>**), the **mail** program sits quietly and waits for me to type in a message. The next several lines in the example above, everything between `Bill:` and `^D`, is the message that I am sending to Bill. The **^D** (CTRL-D) signals the end of the message. After typing the CTRL-D, I would see the UNIX prompt again.

Thus, with a simple command and some typing, we've sent a message to Bill.

How to Read Your Messages

What do you do if someone sends you a message? Reading your mail is very easy. Just type the word **mail** to your UNIX prompt and hit **RETURN**. If you have any messages in the mailbox that the system keeps for you, they will be displayed on your screen. Remember, though, that the latest mail message will be displayed first. This is **LIFO** - *Last In, First Out.* If you don't keep this in mind, you might get confused as to the time sequence of the mail messages.

When you read your messages, your have some options as to how you can dispose of your mail. The basic things you can do is (1) save your mail, (2) delete your mail, (3) forward your mail to another user, (4) move around within your mail messages, and (5) exit from mail. We will discuss how to do each of these operations. First, let's look at the following display:

```
$ mail
From root Mon Jul 20 15:09:19 1987
Mark,
We really appreciate all the fine
work you've been doing recently...

?
```

This is what I would see if I had a mail message. The final question mark (**?**) is the prompt from the mail program. I can type in various commands at this point that control how mail will dispose of the displayed mail message. If I want to see what options I have, I can type a question mark (**?**) to the question mark (**?**). If I do this, I will see the following display:

```
q       quit
x       exit without changing mail
p       print
s[file] save (default mbox)
w[file] save without header
-       print previous
d       delete
+       next (no delete)
m user  mail to user
! cmd   execute cmd
?
```

The first two options listed control how we exit from mail. If you type a **q** to the question mark, we quit from mail with the understanding that any messages you had previously deleted, will in fact be deleted. If you type an **x** to the question mark, mail exits without deleting anything. Sometimes the **x** option is good to remember if you happen to delete a message that you later decide to save.

However, in most cases, you will want to delete a message after you read it. Otherwise, your mailbox will fill up with unwanted old messages. Deleting a message is very easy: just type a **d** and the current message will be deleted.

If you have been sent some information that you want to save, it is still not a good idea to leave it in your mailbox. It is actually a better idea to write it to another file. The **s** and the **w** options listed above perform this function. Thus, if you type just an **s**, the message you are viewing will be saved into a file called **mbox** in the current working directory.

mbox is the name of the default file in which messages are saved. As you save a message, the newly saved

UNISOFT INC.

Experts in Computer Consulting - Sales - Service

Presents

FORTUNE SYSTEMS HELP

We Provide:

- Expert Trouble-shooting
- Years of programming and accounting experience
- *Free Initial Consultation*
- Training Available for:
  - * BASIC/IDOL/BAS and Multiplan
  - * UNIX and Ext. Fortune:Word
- Hardware Repairs & Upgrades
- FORTUNE trained experts at a cost that will surprise you!
- *Trade-Ins Welcome*
- Specializing in:
  - * Hard Disk Trade-In
  - * Hard Disk Upgrades

Authorized Fortune Systems Repair Depot
Fortune Systems National Value Added Reseller (VAR)
CALL: (303) 671-8600 EMERGENCY HELP: (303) 649-8505

UNISOFT INC. 14100 East Jewell Avenue Suite 27 Aurora, Colorado 80012

message is appended to the mbox file. You can save a message to a file of your own choice by simply putting a file name after the **s**, as in:

```
? s filename
```

The current message would then be saved to the file called *filename*.

The **w** option works exactly the same way except the header line of the mail message is not saved. The header line is the line that begins, in our example, `From root...`

This is only a subtle difference between the **s** and **w** command and will not be discussed further at this point.

Another valuable feature of **mail** is the ability to forward messages to other users of the system. Suppose someone sends me a message regarding a cookout planned for the next Sunday and also invites me to feel free to invite anyone else I might like. An easy way to send this message on to other users is the **m** command, as in:

```
? m fred sally
```

This command would forward (or mail) the current message to both fred and sally.

The commands **-** and **+** are used to move backward and forward through your mail messages. If you type a **-** to the question mark prompt, the preceding message will be re-displayed. A **+** typed to the prompt, will display the next message. Please note also that simply typing the RETURN key will also display the next message.

The final function is accessed by the exclamation mark (**!**). This is a shell escape, which allows you to run UNIX commands from within the mail program. Thus, any command that you could run from your regular UNIX prompt, you can run from within mail by preceding the command with an exclamation mark.

Letting You Know That You Have Mail

The system tells you that you have mail in two separate ways. First, each time you log in to the computer, presuming you have mail, the system will print out:

```
You have mail.
```

This, as you might guess, indicates that someone has sent you some new mail since the last time you looked at your mail.

The second way the system tells you that you have mail occurs while you are logged in. (It should be noted that this entire discussion is predicated on the assumption that you are using the UNIX Bourne shell). If you have set the environment variable **MAIL**, then the system will tell you that you have new mail almost immediately after you receive it. Actually, the system takes a look at your system mail file after each UNIX command you type and if there is a change in this mail file, the system will print out on your terminal the message: `You have mail`. Setting the **MAIL** environment variable is best done by adding a line to your **.profile** file (for more information on the **.profile** file, see Volume 2 Number 5). For

TRI-TECH COMPUTER SERVICES, INC.

At Tri-Tech Computer Services, Inc. One of our goals are results that don't cost you a fortune (no pun intended). We are a service organization which offers new and used system sales as well as service to new and existing Fortune products. Our people are technical, but they communicate on a level **you** can understand.

At Tri-Tech, you will never encounter pushy salespeople, only people who will do their best to serve you.

Our offices are located in Kensington, Connecticut and Newton, Massachusetts. With clients on both coasts and in between, we support the complete line of Fortune's multi-user computer equipment. When you deal with **Tri-Tech**, you are dealing with a company that is willing to 'go the extra mile' to ensure complete satisfaction.

We consider ourselves experts with Fortune products. We offer superior service and support at reasonable rates. We have the experience, the resources and *now* a new nationwide toll free number to handle your requests.

We're taking over the No.1 spot. And we mean business. If you need Fortune Computer equipment or services, consider the company that many corporations rely on.

Tri-Tech Computer Services, Inc.

Call Toll Free 1-800-622-TTCS or write Tri-Tech Computer Services, Inc., 106 Cynthia Drive, Kensington, CT 06037.

Number one is no further than a toll free call away. (In CT call Collect)

SPECIALS

H.P. Series II, LaserJet or LaserJet 500+ (15% List with this ad).
Fortune Workstations and Genicom Printers (Used)

Kensington, CT (203)828-9784 Newton, MA (617)965-4453

Toll Free 1-800-622-TTCS (Outside Connecticut)

the following example, let's suppose that your login account name is **fred**. In this case, you would add the following line to your .profile file:

```
MAIL=/usr/spool/mail/fred
export MAIL
```

If your account name was mary, you would simply replace fred with mary. (Don't tell fred!).

Miscellany

So far in this discussion of the mail program, we have indicated that you can send a message to someone by first typing something like: **mail mary**. Then, you would type your message on the terminal and end the message with a CTRL-D. This is not the only way of getting a message to mary with mail. It is often much more efficient to first compose your letter in your favorite editor (e.g. **ed**, **vi**, or **sc**) and then direct mail to send this file to mary. This is accomplished quite easily by using UNIX's redirection (for more on redirection, see Volume 4 Number 4). Let's say you create a file called **mary.let**. You would mail this to Mary by typing:

```
$ mail mary < mary.let
```

The **mail** program, as you have so far gathered, is useful for sending messages to others on your system. The UNIX **mail** program can also be used as a simple reminder mechanism. Simply mail yourself reminders of things to do, and you will be reminded of them each time you read your mail.

The Global Menu and Mail

Even though mail is now part of FOR:PRO it is not on the Global Menu. However, this can remedied relatively easily. Perhaps the easiest remedy is to escape to the UNIX shell and use mail as has been described in this article. To escape to the shell from the Global Menu, simply type **!** to the prompt on the Main Menu of the Global Menu. You will then probably get a dollar sign prompt and then you can use **mail**. After finishing, you can get back to the Global Menu by hitting CTRL-D.

The other way is to set up an entry on the Global Menu which will run a shellscript that allows you to either read your mail or send mail. We include a shellscript that will allow you to use mail through the Global Menu (see Figure 1).

To add the mail capability to the Global Menu requires two steps. First, type the shellscript into a file on your Fortune. We suggest that you put it in the **/usr/local** directory and call it **mailscr**. If you don't have a **/usr/local** directory you can create one by following these steps: (1) Login as root; (2) Type `mkdir /usr/local`.

Second, you must run the **addglobal** command. We regularly include this command on all of the new versions of our diskettes. If **addglobal** is not on your system, please check on our software diskettes for this program. (If you currently do not have a complete set of our diskettes, we highly recommend that you purchase them. You can't beat the price and they are, honestly, filled with extremely helpful programs...not the least of which is **addglobal**).

The **addglobal** command is used to install a program on the Global Menu. It is self-explanatory. Simply run it and answer the appropriate questions. Before you do this, however, you must be logged in as root. The following is a sample addglobal session:

```
# addglobal

Would you like to (a)dd or (d)elete a
product?
a
Please enter the selection you wish to
change (eg. E5)
E4
Thankyou...Now please enter the product
name
Mail
Ok...Now please enter the programs full
pathname
/usr/local/mailscr

/m/menu/control/:global:E4:':   Mail:/
usr/local/mailscr:

Depress <RETURN> to continue or <CANCEL/
DELETE> to exit

successfully updated menu system '/m/
menu/control/' menu screen 'global' se-
lection 'E4'
#
```

This will place the **mailscr** script on your Global Menu under E4.

Conclusion

In this article we have discussed how to use the UNIX **mail** facility. Mail is very easy to use and is a convenient way of sending messages to other users on your Fortune system. In addition, we discussed how to add the **mail** capability to the Fortune Global Menu. If you have comments on this article, please write us. □

Mark Palmerino

Figure 1

```
:
prompt "Please enter r to read mail or s to send mail: "
read ans
echo
case $ans in
        r) mail;;
        s) echo "Please enter name(s) to which"
           prompt "mail is to go:"
           read names
           set `echo $names`
           echo
           prompt "Have you already composed your letter (y/n): "
           read composed
           echo
           case $composed in
                        n) echo "Type in the text of your message"
                           echo "Followed by a CTRL-D on a line by itself"
                           echo
                           mail $*;;
                        y) prompt "Please enter name of file: "
                           read filename
                           echo "Sending mail..."
                           mail $* < $filename;;
           esac
esac
```

**** See the NEW product DOS LINK ****

SOFTWARE FOR THE FORTUNE 32:16

NEW PRODUCT!!! DOS LINK ($495.00) - provides easy-to-use menus and programs that transfer data between any PC (including Laptops) running DOS and the Fortune system. Here are just a few examples of how this product can be used:

- Transfer Multiplan or Fortune:Word Documents from or to your DOS system.
- Transfer other ASCII or binary files.
- Use the PC or Laptop as a remote terminal on the Fortune system.
- Transfer THOROUGHBRED (SMC) BASIC data that has been extracted using one of our other products

Documentation and software for both the DOS and Fortune system are provided. Data may be transferred via a direct connection (9600 baud) between the two computers or via modem (at modem speed). You may also order additional hardware for your PC that may be required to accomodate the transfer.

If you are interested in purchasing a Laptop computer that can be used with this product for people on the road or at home, give us a call!! We also provide consulting services for interfacing your DOS system to the Fortune ($50/hr).

MULTIPLAN LINK - can pass data from BAS/IDOL/BASIC files to Multiplan and back. For example, it could be used to send data from any of the BAS files (Chart of Accounts, Customer Master, Inventory Master, etc). Allows selection of fields to be sent and provides logical retrieval and key range selection. **$295.00**

RECORDS PROCESSING LINK - can pass data from BAS/IDOL/BASIC files to Records Processing and back. For example, it could be used to take customer address data from the Customer Master and will automatically make the Records Processing List document for mail/merge. Allows selection of fields to be sent and provides logical retrieval and key range selection. $295.00

ASCII LINK - can pass data from BAS/IDOL/BASIC files to ASCII and back. Use to send data between machines or databases. For example, it could be used to send data from any of the BAS files (Customer Master, Inventory Master, etc). Allows selection of fields to be send and provides logical retrieval and key range selection. **$295.00**

TERMINAL/PRINTER LINK - provides capability to print data on a printer connected to a workstation. Can print BAS/IDOL/BASIC reports, Multiplan and Fortune:Word documents. BAS/IDOL reports can also be viewed before printing, converted to Fortune:Word format; accumulated for mass printing; backed up to diskette; printed on any system printer. An option to print BAS/IDOL reports on a LASER PRINTER is $50 extra. Many other options are included. **$195.00** (Fortune:Word printing on a printer connected to a workstation is $50 extra)

TELEPHONE LINK - on line telephone message, telephone directory and inter-office mailbox/menu system. Telephone message entry resembles typical pink slip. Telephone directory can search by name or company. Mailbox/message system can send and receive Fortune:Word documents or Multiplan documents. Does much more! Give us a call. Written in Business Basic. **$295.00**

CALCULATION LINK - provides programs for amortization, depreciation, loan repayment, averaging (with graph), linear correlation and breakeven analysis. Written in Business Basic. **$95.00**

KOMPACT PERSONNEL ACCOUNTING - provides for data capture of personal, wage/salary, job, education, salary/wage, appraisal training, dental, medical and life insurance data. Includes over 80 pre-defined reports/worksheets. Written in Business Basic/IDOL. **$995.00**

PRINTER LINK - provides printcaps for Okidata 84, 2410, TI 855, Anadex 6500 and Printronix. Includes shell script for changing pitch and LPI. Allows these printers to be used in Fortune:Word. **$95.00**

MODEM LINK - allows Hayes 1200/2400 modem to be used on one port for both incoming and outgoing calls. **$95.00**

WESTERN LINK - provides instruction on how to use your Fortune for sending/receiving telex messages using Western Union's Easylink. Can replace your telex! Fortune ITE or Handshake software required. **$95.00**

UNI-KOMP 9803 Rocktree Houston, Texas 77040 (713) 937-7782

System Administration: Part 3

No Longer Requires a Degree in UNIX Wizardry

This series is a must for all owners and users of the Fortune computer. Dave is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the Fortune Computer, and is president of the Houston UNIX User's Group. He contributes independently to ***/u/fortune news.***

by Dave Kloes

In the last issue, we finished our discussion of the ".profile" file which was the first file/directory listed in the / directory. If you remember, we provided the following list of standard files and directories in the / directory:

```
# cd /
# ls
.profile dev      sa       unix
b                 etc      lost+found
tmp               usr      bin  f
m                 u
```

The next "file" we have to discuss is the "b" directory. We refer to "file" here because according to UNIX documentation, a directory is just another kind of file. The "b" directory normally contains all of the software and programs for Thoroughbred Basic (SMC Basic), IDOL and the Business Accounting System (BAS) if this software is installed on your system. If you do an "ls" in the /b directory, it should look something like this:

```
# cd /b
# ls
BAS           BWORK       basic.Tpool
cutuf.sod     etc         BDATA
IDOL          basic.psd   cutug.eod
ipls          BUTIL       TD
cutrr.sod
```

The "BAS" directory contains your BAS programs. The "BDATA" directory contains your BAS data files. The "BUTIL" directory contains the Business Basic utility programs. The "BWORK" directory is a working directory for BAS and IDOL. If you have BAS Training installed on your system, the "TD" directory contains the data files for training. The "basic.Tpool" directory is used to keep track of which terminal (**T?**) you are on when using BAS, IDOL, Basic, training, etc. The file "basic.psd" is the Basic interpreter itself. The "cut" files are used for START-OF-DAY and END-OF-DAY procedures in BAS and IDOL. These are the shell programs that are executed when these functions are selected. The "etc" directory (not to be confused with "/etc") contains shell routines associated with the new 6.5.12 Thoroughbred interpreter. The "ipls" directory contains the "ipl" parameter files that the Basic interpreter uses for each user that uses the interpreter.



This month Dave talks about the /b, /bin and /dev directories. In addition, he summarizes UNIX file permissions and devices in the UNIX file structure.

We now return back to the root directory (/). The next directory that we will discuss is the **/bin** directory. The word "bin" is commonly used on UNIX systems to refer to a storage place for program files. In particular, "/bin" contains the commands normally reserved for the System Administrator's use. Notice that there are no subdirectories in this directory. We will not discuss each of the individual commands in this directory. Rather, we will discuss some of these commands as we proceed, where they are applicable to the System Administrator.

The **/dev** directory is different from

other directories on a UNIX system. It contains all of the "special" file names for devices. On a UNIX based system, each device that is used by the system has a name associated with it. Here is a partial listing with the associated device name:

```
Device                  Device Name

Floppy disk drive       fd*, rfd*
Hard disk drive         hd*, rhd*
Terminals               tty*, console
```

Each "partition" on the hard disk has a separate device name for it. For example, **/dev/hd00** is the "boot" partition, **/dev/hd01** is the "swap" partition, and **/dev/hd02** is the partition that has your system programs and data files. In addition, if you have more than one disk drive on the system, the partitions of the second disk drive device would be referred to as **/dev/hd10** and **/dev/hd12**. (**Editor's Note**: Next month we will begin publishing a series of articles on various aspects of the UNIX file system in which some of these details will be discussed in a fuller manner). Let's look at a couple of definitions:

***boot* partition** - portion of the disk that contains information about the device that is read when the system is "booted" or started up.

Memory - that portion of the system that actually stores the information that is being processed. Consists of ROM and RAM. Physically, these are "cards" with computer chips, etc. that are inside your system.

Read Only Memory (ROM) - this is the portion of memory that is always resident.

Random Access Memory (RAM) - this is the portion of memory that is used for processing user tasks.

***swap* partition** - in normal operation, users data and programs are processed in RAM. When the memory that is available on your system is used, the system "swaps" jobs between memory and the "swap" partition of the disk. The larger the "swap" space, the faster the system will run because more data can be processed. When a hard disk is first loaded, you can specify the swap space. Based on this, a portion of the disk is set aside for swapping. That's one reason why even though you may have a 20 megabyte (20 million characters of storage) disk, all 20 megabytes are not available for storing your data.

We will be discussing these concepts in more detail in a later article. The discussion at this time is in relation to the device names that are contained in the "/dev" directory. The device name for the "boot" partition of the hard disk is /dev/hd00 or /dev/rhd00. The device name for the "swap" partition of the hard disk is /dev/hd01 or /dev/rhd01. The device name for the data partition of the hard disk is /dev/hd02 or /dev/rhd02. These device files also look different than normal file or directory entries when you do a **ls -al** command. Let's take a look at a couple of them:

```
# cd /dev
# ls -al hd00 tty02
brw-r—— 1 root   1,  0 May 14 07:23 hd00
crw-w-w- 1 root 9,  0 Jul 19 03:56 tty02
```

We are the *largest* Authorized Fortune Master Dealer in the Pennsylvania and Delaware area. As part of the **P**rofessional **S**olutions **G**roup of companies with over 35 years in business automation, we have the *stability* that our customers demand

Our services include a factory trained field engineering team, a complete parts inventory including system upgrade capability and a software support department.

For all of your Fortune needs contact:

Mark Shields of PSG at
2550 Eisenhower Avenue
PO Box 762
Valley Forge, PA 19482

or you can call Mark at (215) 666-1400

The first character of the read/write permission section is "b" for hd00 and "c" for tty02. The "b" stands for a block device and the "c" stands for a character device. If data for that device is written in blocks (normally 512 or 1024 character segments) then the "b" is used. If data for that device is written a character at a time, then the "c" is used. Data is normally read or written to disk in blocks (remember hd00 is the first partition of the disk) and is entered from terminals (tty02 for example) a character a time.

For those of you that are not familiar with permissions on UNIX files, now is probably a good time to summarize them for you. (Also, for more information on file permissions, see Volume 1 Number 5 of /u/fortune news). First of all, remember that usually the first of the 10 "bits" of information above is normally a "-" for an ordinary file or a "d" for a directory. The other bits are for the read, write and execute permissions on a file. If we were to split these nine bits up into their logical sections, they would look like this:

```
r  w  -     r  -  -     -  -  -
user        group       other
```

The first three bits would be applicable to the owner or user of the file. If groups are set up on the system (on the Fortune, everyone defaults to the same group), permissions are determined by the second set of three bits. If you are not the owner or do not belong to the group, the last three bits would determine your permissions. Each of the three groups could have any combination of "rwx" (**r**ead, **w**rite, **e**xecute) permissions. *Read* permission indicates whether you can view the file. *Write* permission indicates whether you can change the file. *Execute* permission pertains only to executable files (such as programs or shell scripts) and determines whether you can run the program.

The **chmod** command is used to change permissions on a file and only the owner of the file or "root" or "manager" can do this. The command uses the following format:

```
chmod  _____  +  _____  <filename>
       ugoa   -  rwx
```

The first parameter you enter is any combination of "ugoa" depending on the set of permissions you want to change where "u" stands for "user" or "owner"; "g" stands for "group"; "o" stands for "other" or "a" to change all three. The "+" is entered if we want to add permissions or the "-" to take permissions away. Any combination of the "rwx" can be used to indicate what permissions we want to effect. If we wanted to add "write" permission for "group" and "other" in the example above, we would enter:

```
chmod go+w /dev/tty02
```

The permissions would then look like:
rw-rw—w-

If we only wanted to add "read" permission for "other" we would enter:

```
chmod o+r /dev/tty02
```

The permissions would then look like:
rw-rw-rw-

SUPPLEMENT TO ALL VERSIONS OF BAS AT A REASONABLE PRICE, TOO!

JOB COSTING

The Right Tool for All Trades

BaSiC SOFTWARE CORPORATION

For More Information Call or Write...
BaSiC Software Corp.
5201 Powhatan Street
Baltimore, MD 21207
301-448-9460

Available on:
IBM AT & compatibles
UNISYS/SPERRY
Altos, All Levels

NEW TAX LAW MAY AFFECT YOUR COMPANY

If your company has over 250 employees, you must submit a form to the Social Security Administration (IRS) by June 30, 1987 indicating how you plan to submit your 1987 W-2 information.

Uni-Komp HAS YOUR SOLUTION

W-2 LINK will AUTOMATICALLY take your existing BAS Payroll files and put them on diskette in the format required by the IRS.

We are ready to help you with this process and in filling out the IRS form. Call (713) 937-7782 now to place your order. Cost of the software package is $395.00 if ordered before September 1, 1987. Cost of the package after September 1, 1987 is $495.00.

Uni-Komp
9803 Rocktree
Houston, Texas 77040

If we wanted to take "write" permission away from "user", "group" and "other" we would enter:

```
chmod a-w /dev/tty02
```
OR
```
chmod ugo-w /dev/tty02
```

The permissions would look like:
r—r—r—

This last example would add "write" and "execute" permissions for "user" and "group":

```
chmod ug+wx /dev/tty02
```

Now our permission bits would be:
rwxrwxr—

Normally, we should not change permissions on the files and device entries in the **/dev** directory. These are usually initially set and subsequently controlled by the system. We felt the discussion on changing permissions was appropriate here since this is the first time we have displayed the results of the **ls -al** command.

Let's continue on with our **ls -al** command result above. The information just before the date shows a "9, 0" for hd00 and "1, 0" for tty02. These are called the **major** and **minor** device numbers. A "major" number describes a certain class of device. In our two examples, any device with a major number of "1" is a hard disk and any device with a major number of "9" is a terminal (Note: device classes are not the same on all UNIX systems). Each of the devices in a major category are distinguished by their minor number. For example, tty02 has a minor number of 0, tty03 had a minor number of 1, etc. Here are a few of the major classes of devices:

0 - floppy disk
1 - hard disk
4 - floppy disk (raw)
5 - hard disk (raw)
9 - terminal (first Comm A)
10 - terminal (second Comm A)

Some of the device entries have an associated "raw" device. For example, for each "fd" entry there is a "rfd" entry and for each "hd" entry there is a "rhd" entry. The "fd" and "hd" entries are block devices and the associated "raw" (rfd, rhd) entries are character devices. Some UNIX commands require that devices which are normally block devices be raw devices. For example, when using the "format" command to format a hard or floppy disk, the raw device name is required. Also notice that devices such as "Comm A" boards (these are the boards where each of the terminals is connected) have their own major number. Typically, the major number corresponds with a particular printed circuit board in your computer, e.g., 0 is the mother board, 1 is the hard disk controller. The minor numbers are often separate "parts" or devices on a board.

The device entries made in the /dev directory are made using a special UNIX (For:Pro) command. The **mknod** command is used to create these special files. Here is the syntax of the command:

```
mknod dn <b/c> maj min
```

where dn stands for the device name, maj stands for the major number and min stands for the minor number. If we were to remove the /dev/tty02 entry above (you will have to be "root" or "manager"), we could remake it by using the following command:

```
mknod /dev/tty02 c 9 0
```

Fortunately, you don't have to worry about manually making these entries on the Fortune system. If you remove the /dev/tty02 entry and execute the following commands, the entry will be made for you:

```
# cd /dev
# rm tty02
# mkdevs
```

The **mkdevs** command makes all the applicable device entries for you. In fact, each time the system is started, this command is executed by the system to insure all applicable device entries are there. This way, any new hardware that is added to the system will automatically have the device entries made for it. On most other UNIX systems, the System Administrator is responsible for making these devices entries.

There are a couple of other devices in **/dev** that we should mention at this point. The file "/dev/null" is the UNIX "garbage can". Any output that is redirected to this device goes to "never never land," not be seen again. Look at the following example:

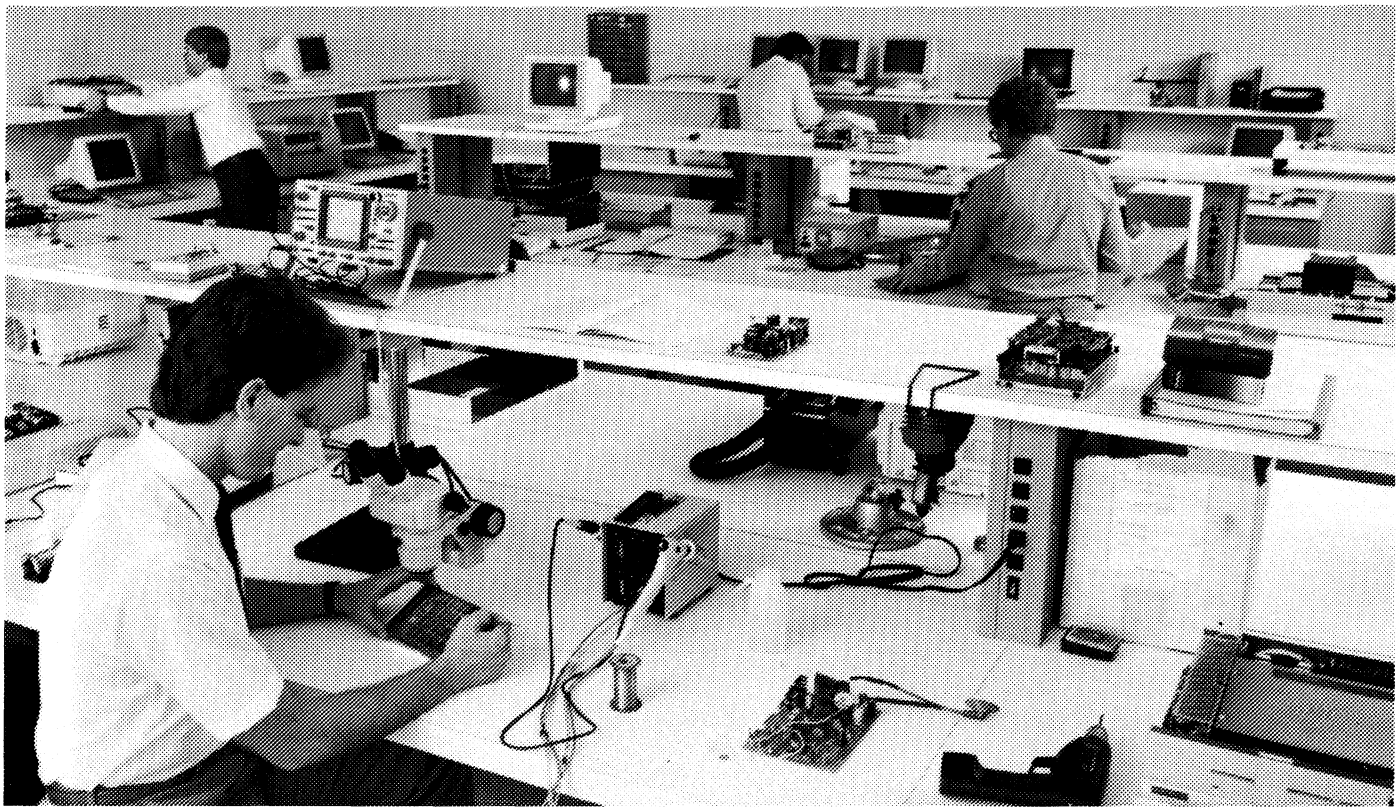
```
dd if=/dev/hd02 of=/dev/null
```

This is a command you can use to check your hard disk. The input file (**if**) is /dev/hd02 which is the data portion of your hard disk. The output file (**of**) is /dev/null which is the UNIX garbage can. Since /dev/hd02 is a block device, this command will read all of the blocks on your hard disk. Depending on the size of your hard disk, the "dd" command will return the number of blocks it has read when the command is finished. If there are any bad blocks, you will get an error message. The command could take a while to finish depending on the size of your disk (e.g. 20 minutes). We'll talk about the **dd** command in greater detail in one of our subsequent articles.

Two other files may be in your **/dev** directory. If you have installed either ITE or Fortune to Fortune Copy, files that start with "cua" and "cul" may be present. The "cu" stands for "calling unit" and the devices are used for calling out on your Fortune. Consult your ITE or FTF manuals for more information about how these files are created and how they are used.

Needless to say, we will not be talking about every file that is on your system. Obviously, if you look at your listing of the /dev directory, there are other devices we have not talked about. Hopefully, this general discussion will help you to understand the **/dev** directory and what it is used for.

In the next issue, we will begin talking about one of the most important directories on your system (at least from the System Administrator's point of view) - **/etc.** □



The Best Just Got Better

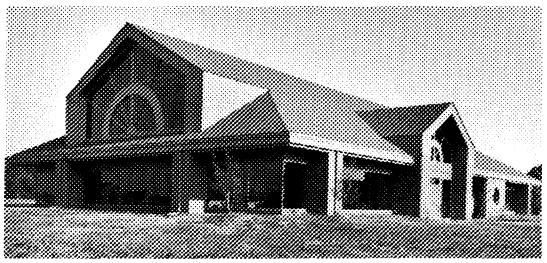
At Micro Resources, our commitment to Fortune/SCI is clear. We've just opened a brand new facility to make us even more competitive and service-oriented for Fortune Systems users.

Expanded Service
Not only have we added service personnel, our newly-equipped service department is simply the most advanced in the mid-west. That means fast turn-around for system upgrades or repairs.

Expanded Inventory
Service can only be as fast as the needed components are available. That's why we've enlarged our on-hand inventory. Volume purchases also keep our prices competitive.

Expanded Upgrades
All this growth has led to a landslide in upgrade business. That volume has provided us with a wealth of components and systems available in limited quantities at fantastic values.

Expanded Classrooms
Now Micro Resources has the space and integrated training facilities to host dozens of employees. Our training programs are tailored to your needs and your schedule.



We believe in Fortune/SCI. And we believe in supporting Fortune/SCI users with the finest service and product at a competitive price. So, if you're unable to find a resource to meet your needs, call Micro Resources. Our dedication to Fortune Systems and to complete service is simply unsurpassed.

Call us at 1-614-766-2335. In Ohio, call toll free 1-800-622-UNIX.

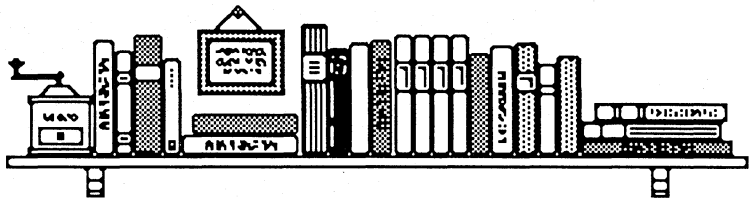
Authorized Fortune Master Dealer

Micro Resources

255 Bradenton Avenue, Dublin, Ohio 43017

BEYOND TECHNOLOGY . . . *RESULTS*

The BASIC Advisor



The BASIC Advisor is brought to us from Ray Wannall. Ray is President of BaSiC Software Corporation which is located in Baltimore, Maryland. This month Ray discusses an upcoming Concept Omega conference, BAS sales and support, and finally Ray helps a reader from the U.K.

We are gearing up for Concept Omega's "Thoroughbred Leaders Conference" formerly known as the "Thoroughbred Dealers Conference". This year it will be held at the Marriott's Hilton Head Resort in Hilton Head, SC, September 16-18. I can hear some of you saying, "But I thought you said it would be in Florida this time!" Hey, I tried! Potential participants were polled this past spring, and the choices offered were South Florida, Central Florida or Hilton Head. My contingency just came up short. Nevertheless, I expect the conference will be even more exciting than last year's affair in San Diego because fourth generation software has arrived.

For those of you who have not been bombarded with Concept Omega's surveys, promotional literature and press releases, here is a sampling of the hype:

- The opening ceremonies on Wednesday will include a synchronized slide show with music, organized around the conference theme.

- Proposed sessions include sales and marketing tips, new products for 1988 and beyond, developing applications with the fourth generation language, a technical forum, and the popular keynote speeches.

- The Thoroughbred Leaders Classic, a golf tournament for duffers and pros alike, will be held on Tuesday, September 14. Foursomes will be arranged and a variety of competitions held. Winners will be presented awards at the Friday "Winners Circle" luncheon.

- Participants are encouraged to bring their spouses. Activities are being planned for those who do not wish to attend the conference sessions. Possibilities include: an all-day tour of Savannah; a 2-3 hour tour of Hilton Head; a fashion show; an art gallery tour; wine tasting and more.

- Planned are a "Low Country" beach party on Wednesday evening and a "United Nations" banquet on Thursday evening. Each course of the latter will be a specialty from one of Thoroughbred's international dealer's countries.

- Activities on Wednesday and Thursday will also include the Software Fair and Manufacturer's Hospitality Suites.

- Recent press releases from Concept Omega have announced the early completion of the fourth generation language (in time for the conference), a new Thoroughbred computer (it can't be better than the Fortune) and the addition of 120 new Value-Added Resellers (VAR's) in a 90-day period. As of July 17, Concept Omega had processed approximately 70 registrations, which is far ahead of the 1986 pace. As usual, yours truly is included in this number. I hope to again meet many readers of /u/fortune news and encourage you to register if you have not yet done so. We want Concept Omega to know that Fortune dealers are interested in their activities: we certainly made ourselves known last year! Although walk-in registration will be accepted at the start of the conference, it is recommended that, to assure yourself of a room, you contact Concept Omega at (201) 722-9565 before September 1, 1987. The conference fee is $350.00 after August 15 (additional company representatives may attend for $300 each). The hotel room is $95.00 per night, up to 2 in a room, and $10.00 per night for each adult beyond 2 in a room. Concept Omega is prepared to assist you with travel arrangements.

I will cover the highlights of the conference in a Fall, 1987, issue of /u/fortune news, so stay tuned. Now, back to our normal stuff.

***Question**: Now that Fortune Systems has been sold, who is responsible for BAS sales and support, Fortune/SCI or Tigera?*

Answer: Who supports BAS other than Fortune dealers is a question that

than Fortune dealers is a question that has had me perplexed for years. As far as sales are concerned, BAS for Fortune will be a product of Fortune/SCI. I was elated to learn recently that **Elizabeth Muth** has remained with Fortune/SCI and that her responsibilities have been expanded to include customer support, product support, beta installations, and third party software.

By the way, has anyone been as confused as I have been about the pronunciation of Tigera? Apparently the "i" and the "e" are long, the "g" is hard and the accent is on the second syllable.

***Question**: I goofed. A few years back I expanded the length of a data element in an IDOL-defined file that came with my order processing software.only to learn later that the original length would have sufficed. But we now have data in the file which needs the additional field length, and changing all of the records at this time is more of a job than I wish to handle.*

My problem is that new records going into the file need to be defined with the shorter length. New records are entered with a BAS data entry program which is, somehow, finding the expanded length, printing the underline prompt on the screen and overwriting other information on the screen which the operator needs to see. My research has pinpointed the variable L9 as the carrier of the element length, but L9 is redefined for each field. This one is a little over my head. Can you help?

Answer: Sometimes IDOL gives us just enough tools to make us dangerous. Unless your understanding of IDOL greatly surpasses the information contained in the documentation, I strongly recommend that in the future you confine your programming efforts to new work and lay off making modifications to the existing packaged software. In the meantime, here's your fix.

Among the restricted variables you are told (without reason) not to use are J9, L9, A$ and D$. In BAS data entry programs, which were created under the IDOL system, A$ and D$ are used to hold information read from the INDEXED "*DE" file containing the data entry screen attributes. The A$, which is in the IND=0 record, contains a picture of the screen itself and is activated with the command:

```
PRINT 'CS','SB',A$,'SF'
```

or some variation thereof. The D$ is read from the IND=3 record (second field) and, when printed, may look something like this:

```
070204 C    250220B 2 070406A
A250425     070620 A  250625
```

The length of D$ will always be a multiple of 10, with each group of ten characters relating to a single data element to be entered. This means that a data entry program which is designed to take 20 fields of entry on one screen will use a D$ with a length of 200. When we display our above example as individual blocks of ten, we can more clearly view our element attributes:

Element 1: 070204 C
Element 2: 250220B 2
Element 3: 070406A A
Element 4: 250425
Element 5: 070620 A
Element 6: 250625

FORTUNE USERS:

SIMPLIFY SYSTEM-TO-SYSTEM FILE TRANSFER
TRADE UP AND SAVE MORE THAN 75%

Is your Fortune to Fortune Copy cumbersome, difficult to set up and use?

Do you want easy to understand menus so anyone can send files without shouting for help?

Are you frustrated by the complex setup and maintenace required by Fortune to Fortune Copy?

Would you like to have an easily maintainable Fortune to Fortune network with solid security?

YOU NEED HANDSHAKE BLAZER!

TRADE UP your Fortune to Fortune Copy
for HANDSHAKE BLAZER and save $$$$

HANDSHAKE BLAZER standard list is $695

Limited Time Offer
ONLY $150

This special offer good thru August 31, 1987.

For more information call:

DATA INC.
250 S. Main St. #201
Thiensville, WI 53092
(414) 242-3999
or your LOCAL FORTUNE DEALER

The first four numbers in each block dictate the cursor position for the entry field. Element 1 entry is taken at column 7, line 2 on the screen, element 2 at column 25, line 2, etc.

The next two numbers represent the maximum number of characters which may be entered for the element, i.e., your L9 length. More on this in a moment.

Position 7 is the padding indicator, position 8 is the element type, position 9 is the precision (or blank for non numeric elements) and position 10 is a date indicator if applicable. These terms are defined in your IDOL manual.

The data entry program keeps track of which field is being entered with a simple counter contained in J9. J9 is increased by one each time you complete entry for a field and reduced by one each time you press <F3> to backup. The number one is subtracted from J9 and the result is multiplied by 10 just before the INPUT statement so that values contained in D$ can be referenced:

```
7500 REM "GEN D.E."
7510 LET R9=(J9-1)*10
7520 LET
P7=NUM(D$(R9+1,2)),L7=NUM(D$(R9+3,2)),L9=NUM(D$(R9+
7520:5,2))
7540
INPUT(0,ERR=7500)@(P7,L7),X7$:(LEN=L9)
```

In this typical example, P7 has captured the column for cursor positioning, L7 the line and L9 the maximum allowed length of entry.

So how do we tell the program to reduce the value of L9 for one field only? We simply re-define D$ back at the beginning of the program. If you cannot find where D$ is read, you can assume that it was defined before the entry program was run. (To verify, throw in an ESCAPE right after statement 100 and print D$.) At a point in the program after D$ has been defined (but before data entry has begun) you can add a statement to alter the value of D$, for example:

```
0105 LET D$(5,2)="06"
```
(Change Element 1's length to 6)

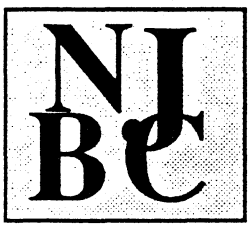
```
0105 LET D$(15,2)="15"
```
(Change Element 2's length to 15)

```
0105 LET D$(45,2)="24"
```
(Change Element 5's length to 24)

The above process will alter this data entry program only: other parts of your IDOL system will not be affected because we have not written the value of D$ back into the *DE file.

***Question**: It is a sad fact of life that it is very hard to find technically competent people here in the United Kingdom, when compared with the general level of technical know-how that can be assumed in the United States, and I am sorry that I have to trouble you from so far away!*

Before leaving my previous employment to become an independent consultant, I set up a Fortune-based BAS/IDOL/BASIC set of applications. Some of these applications were specifically designed using IDOL (release 1.2), but I had to work with some severe limitations in these early versions (ie restricted to a max of 15 characters



New Jersey Business Computers

Authorized Fortune Systems Master Dealer

Authorized Fortune Service

Authorized Fortune Training Center

NEW JERSEY BUSINESS COMPUTERS has the Largest Affiliate Dealer Network in the New York and New Jersey Metropolitan Area.

For The Most Reasonably Priced TRAINING, SERVICE and SUPPORT Available.

call

301 Route 17 North, Rutherford, New Jersey, 07070

201-438-1410

per field, even though the manual said that 30 could be used-and had been used-for the BAS designed applications).

Since then, the company approached me because they wanted to run the original IDOL-created data bases using INFORMIX. I looked at the problem, needed help and could not find it in the UK. By then, the company changed its mind and lost interest, just as I was about to order the ASCII link from UNI-KOMP in Houston, Texas.

My question is, is there an easy way for me to transfer an IDOL created file (really an SMC BASIC file) so that the end of record markers will be compatible for use by an INFORMIX database that mimics the original IDOL one?

Note: The reason for this is that difficulty was experienced when a 20th field was added to the original IDOL database. The file had been transferred and expanded to accommodate the addition of this extra field; yet although the field was accepted by the header detail, no data could be entered into the field. I won't bore you with the details. Suffice to say that INFORMIX was seen as having less restrictions so that changing specs, adding fields, and having much larger field lengths would have been much easier.

Finally, I note that a BAS/IDOL update is available (release level 1.0) and that IDOL II has been released. (I don't think this is the same as IDOL release level 2.0). I would be grateful for any information on these.

Answer: I certainly do not feel troubled by your inquiry from far away. In fact, it's rather pleasant communicating with someone other than fellow colonists.

When you refer to a restriction of 15 characters per field, I must assume that you mean per field description. The actual number of characters per field is defined in the element detail record under EL LN, field 2, which allows fields to be defined up to 99 characters. The field description, however, is a part of the key to the Global Element Dictionary (CGLBD) and must be exactly 15 characters long (regardless of what your book says). I agree that this restriction can be frustrating at times, especially when you want to say something like "QUANTITY SOLD FROM WAREHOUSE NUMBER 1, YEAR-TO-DATE", which comes out looking like "QTY SOLD W1 YTD". But I'm afraid that all versions of IDOL have this quirk.

Concerning your question about IDOL to INFORMIX transfers, I called Dave Kloes at UNI-KOMP who advised me that yes, it is currently being done. Since you are in Dave's playground, I suggest you write to him for further details.

Note: The reason your file would not accept data in your new field is because you did not tell the header record where to display the new field during file maintenance. When you add a field, header field number 4, NO OF ELMS, is automatically updated by IDOL. \U\You\u\ must increase one of the four fields in the header record called COL LEN (column length) in order to display and maintain the new field.

The BAS/IDOL update, release level 1.0, was issued by Fortune back in early 1983. It is probably useless at this time. Another update diskette was released around December of 1983. This one was BAS release 1.1 and IDOL release 1.3, and it required FOR:PRO 1.7 or above. And if you have been reading this publication with regularity, you should know that we now have the infamous 2.0 BAS applications. You can purchase IDOL II from any Thoroughbred dealer. Contact Concept Omega at PO Box 1035, 102 Old Camplain Road, Somerville, NJ 08876 for information about dealers in the United Kingdom. If you do not wish to purchase IDOL II (designed for software developers and entirely different from IDOL), I'm sure you can call a Fortune dealer for a copy of the 1.3 IDOL update. If you cannot find one, write to me again.□

Vantine Computer Services

Professional Solutions for Business Microcomputer Users

Sales Service Consulting
Software Customization Support
Software/Hardware Upgrades

Over Three and a half years supporting Fortune Users

Vantine Computer Services
70 Walnut Street, Wellesley, MA 02181
(617) 239-8084
Authorized Fortune Dealer

Glossary

Continued from page 1

diskette and press EXECUTE. When you are asked for source name, type in either gloss1, gloss2a, etc. You can leave the same name for the destination document. Repeat the process to copy each of the glossaries over. This will place them into your directory. Choose **Remove Diskette** from the **Filing Menu**, and you will be able to remove the examples diskette.

Creating Your Own Glossary

You can create your own glossary the same way you create any other document. Simply go to the **Glossary Functions Menu** and choose **Create a Glossary**. Follow the instructions, and you will end up with a blank **Fortune:Word** document. You can either begin to create your own glossary, copy entries from Fortune's glossary documents, or use the "create by example" method to generate glossary entries. Since we have discussed the "by example" method in the past, and it is clearly explained in the Glossary Manual, we won't go into it here, but it is very easy. Any user is capable of creating and using glossary by example.

Attaching a Glossary

You may have many different glossary files, each containing up to 94 entries. This provides virtually unlimited possibilities for creating glossary entries. You might create one glossary for letters, one for a report, one for memos, etc. However, because you can have many different glossary files on your system, it is necessary to "attach" the one you would like to use at any particular time. Attaching the glossary tells **Fortune:Word** to look in that glossary for your glossary entries. Only one glossary may be attached at one time. Glossaries can be attached from the Glossary Menu, from any Index Screen (with the exception of the indexes generated from within an Edit Screen), using the shortcut code agl, or from the Edit Screen using CMD LF/GL. For a complete description of these different methods, see Volume 2 Number 5 of **/u/fortune** *news*.

Making a Word Bold — Automatically

Attach **gloss3** and go into a test document (you might want to make a copy of an old document that you can play with). Position the cursor on the first letter of a word. Now press the LF/GL key (the GLOSSARY key) and then hit a lower-case c. The word that the cursor was on will be made bold. As it happens, you will probably see each of the steps that happens when you manually use the bold function. Let's look at the entry and some ways that you might modify it.

```
1    entry c  /*in Chapter 10*/
2    {
3       mode"b"
4
5       while(((char >= "A") & (char <=
"Z")) | ((char >= "a") & (char <= "z")))
6       {
7         right
8       }
9
10      mode "b"
11   }
```

The first thing that happens in this entry is the entry label, which consists of lines 1 and 2. Next, in line 3, the "**mode**" key is entered, followed by the letter "b". This turns on bold mode, just as if you had typed those two keys (**MODE b**). If you change the b in lines 3 and 10 to a "u", then this entry will automatically underline a word. Lines 5-8 are used to move the cursor right until it reaches the end of the word, which is determined when the character is not a letter, e.g. a space, punctuation mark, or number. The computer checks to see that the letter the cursor is sitting on is between A-Z (greater than or equal to A and less than or equal to Z) or between a-z. Note that the computer differentiates between upper and lower case in Glossary functions. If the character the cursor is on is a letter, the command is given to move the cursor right. Because the while command is given, this sequence repeats as long as the characters continue to be letters. Once a "non-letter" is reached, the entry continues down to line 10, which toggles the mode "b" back to off, the same way you would if you were doing the task manually. You can see that the only real intelligence the computer needs to perform this entry is to determine where the end of the word is.

We can make this entry work a little faster by adding two lines to the entry. One way to do that is to just choose **Edit a Document,** and enter gloss3 for the document name. The glossary file will appear. Go to page 3 and you will find **Entry c**. Insert the lines 2a and 10a from the listing below. The second way is to start your own glossary file and copy **Entry c** into it. To do that, create a glossary as described above. Then use the SHIFT COPY function to copy from **gloss3** to your new glossary. Make sure to copy the complete glossary including the final brackets. If you use this method, you can rename the entry, create separate versions to underline, etc.

```
1    entry c  /*in Chapter 10*/
2    {
2a      call display(false)
3       mode"b"
4
5       while(((char >= "A") & (char <=
"Z")) | ((char >= "a") & (char <= "z")))
6       {
7         right
8       }
9
10      mode "b"
10a     call display(true)
11   }
```

These lines tell the computer not to bother changing anything on the screen until all of the work is done. call display(false) is a way of telling the computer to turn off the display. Likewise call display(true) turns it back on after everything else is completed. This will make the entry work about 20% faster and also eliminate the momentary bolding of everything on the screen.

Changing a Sentence

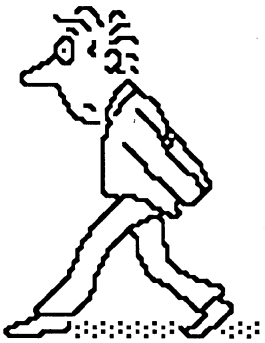
To make this entry work on a sentence instead of a word, we just have to alter the **while** statement in line 5. We want to check to see if the character is a **period**, **exclamation mark** or **question mark**. In this case we want to continue moving right while the character is not equal to a period, etc. The function *not equal* is represented by "!=". That statement would look like this:

```
5    while((char != ".") & (char != "!") & (char != "?"))
```

Have fun playing with these. Next month we'll look at a couple of other entries from the Glossary Disk.□

Josh Lobel

/u/help



Don't let your computer get you down! Write to us about your problems and we'll try to solve them.

Dear /u/fortune news,

/u/fortune news is an asset to the Fortune user community; keep up the good work. The following are just a few hints to add, if you wish, to /u/help:

1. The **grep -i** flag, as noted eons ago (Volume 3 Number 4), does not work. Instead use **grep -y** (undocumented) to accomplish the same thing, that is, ignore case:

```
$ grep -y searchstring filename
```

2. In Mark Anderson's /u/help (Volume 3 Number 9), he discusses altering **/etc/rc** to avoid the necessity of entering the date and time at the setdt screen during system start up. Using the **t** option to **setdt** in **/etc/rc** is another method to accomplish the same end. The argument following **t** (documented in the programmer's manual, setdt(8)) is the number of seconds to wait for user response. If the specified number of seconds is reached, the command times out and the system continues with startup. In this example, it is set to wait 30 seconds:

```
/etc/setdt t30 < /dev/console > /dev/console 2>&1 # e
```

3. On Text Editors (Volume 3 Number 10): **Emacs** has long been ported to the Fortune. There is a version listed in the "Software Catalog" for sale from UniPress Software, Inc., 2025 Lincoln Highway, Suite 312, Edison, NJ 08817. Telephone: (201) 985-8000. I don't know if the implementation is good or even if it is still available. The page in the catalog is dated November, 1984 and at that time was priced at $395.

Best Regards,

Anne Rosenbloom

Anne is currently a Software Support Manager for Fortune in Frankfurter, West Germany. She previously worked in the Fortune support groupp in California and as a Technical Support Manager in Monaco. We appreciate her comments. We've modified our /etc/rc startup file to wait 45 seconds on the setdt command. Thanks, Anne.

Appointment Problems?

The next question was phoned in by a dedicated user of some of our free software. Specifically, she regularly used the program called **appt**. Appt is a simple appointment calendar type program which is found on the **D.C. Grabbag Diskette.** Apparently, she recently loaded on a newer version of the For:Pro operating system (1.8) and after this the **appt** program no longer worked. The error message that she received was:

```
grep: program not found
```

The question is, of course, why did **appt** break and what can be done about it?

Expand your Fortune® without spending a fortune.

New equipment costs big bucks. The Good Fortune Trader sells fully tested burned-in equipment, refurbished to Fortune® standards at prices that won't make waves with your cash flow. Call or write today for a listing of our specials.

THE GOOD FORTUNE TRADER

60 DAY WARRANTY

a division of Computer Services of Atlanta

(404) 447-4151

5825 Live Oak Parkway, Suite 2-C, Norcross, Georgia 30093

Answer: The above error message indicates that the **appt** program is trying to call a program called **grep** which it can not find. (For more information about **grep** and **appt**, see Volume 3 Numbers 4 and 5). The **appt** program is actually a shellscript which does indeed call for the program **grep**. **Grep**, however, is not a regular part of For:Pro but rather is a part of the **Development Utilities**. Thus, most of our readers will not have **grep** on their system. We do offer, for those who are interested, a version of **grep**. This **grep** is included on our **Compressor's Delight diskette**.

The problem revolves around the installation of the new For:Pro. Our best guess is that this new installation inadvertently erased a copy of **grep** that was previously on the system. The remedy would be to recopy **grep** onto the system.

If a copy of **grep** could not be obtained easily, then you could try finding a copy of Volume 3 Number 5 of **/u/fortune *news***. In this issue we explain in detail how to modify the appt shellscript so that it does not require any special programs.

Hung Printer?

We, at **/u/fortune *news***, had a problem with one of our printers recently. Actually, I can't explain exactly how the problem started but the symptoms were as follows. The printer stopped printing and the job it was printing was still in the queue. I used **lprm**, the UNIX command to remove jobs from the print queue, to delete the current job. I thought that this would relieve the problem. However, when I reissued the print job, nothing happened. I could use **lpq** to look at the print queue and sure enough, the print job was queued. But the printer was not going. I tried to remove and resubmit the job several times, but to no avail.

It was about this time that I thought I should try and see what the system was doing. So, I issued a **ps -a** command which prints out all the current processes being executed. I noticed that even though I had removed all the jobs from the queue that there were still some processes associated with the printer. The following display is an abbreviated **ps** listing of the programs I saw listed:

```
2293 p05  0:01 /usr/lib/lpd -p 1
2295 p05  0:01 /usr/lib/lpf -lF...
```

This shows that an **lpd** and an **lpf** were running. I deleted these programs with the UNIX **kill** command (e.g. `kill 2293 2295`). After this the printer worked properly.

I am not sure if this is a problem with the operating system or if it was just a bizarre series of events that led to my printer problem. I would think, for instance, that deleting jobs from the lpr queue would take care of the any associated processes. This is obviously not the case, however, and we include this little story in the hopes that we will save some of you time and money.

The following are bugs that have been reported to Fortune/SCI:

Problem: If searching for a string within a bold and reverse video section of a Fortune:Word document, and the string is found it is not readily apparent.

Solution: Because Fortune:Word uses the bold function to highlight the string that has been found, the string will not be seen on the screen. If the message "Search for what" appears in the top right of the screen, then the string has been found and the cursor will be on the first character of that string. If the message "text not found" shows up on the bottom of the screen, then the string was not found and the cursor will be in the same location in which the search was started.

Problem: If a user, after inserting a minimum of 2 lines of text into a Fortune:Word document, presses the **PAGE** key and then decides to delete it by pressing the **DELETE** key, the up arrow key once and then **EXECUTE**, the terminal will hang.

Solution: The user will ahve to log onto the system from another terminal and from the UNIX prompt, type the command:

```
ps -ax
```

On the screen will be four columns with the headings **PID**, **TTY**, **TIME**, and **CMD** respectively.

Look under the **CMD** heading for the wped fn command (where fn is the name of the Fortune:Word document in which they were working). Now find the **PID** number for that command. Next type the command, fromthe UNIX prompt:

```
kill PID
```

where **PID** is the number you found in the previous step. Your terminal should be back in its normal state. Be aware however, that you are killing your current word processing session. This means that you will lose any unsaved work.

Formula Hangups

There are several situations in which the Formula 4000 and 8000 systems will hang. Generally, the solution to the problem requires attention by an authorized Fortune/SCI trained technician. If you are experiencing intermittent hangs, you should contact your dealer. The following configurations are known to be potentially problematic:

1. Formula 4000 and 8000 systems with 3 or more COMM 6's.
2. Formula 8000 systems with 4 meg boards and multiple COMM 6's.
3. Formula 4000, 8000 and SCSI SXT systems may hang during tape backup when appending files to those already on the tape.
4. Formula 4000, 8000 and SCSI SXT systems may hang during concurrent disk and tape operations.□

Review: For:Pro 2.0

We recently upgraded our 32:16 to **For:Pro 2.0** and have some initial reactions.

Installation

For:Pro 2.0 installs from the Global Menu, just like most other Fortune Software. Our installation went without any major problems. The files that you may have customized such as /**etc/passwd**, /**etc/group**, /**etc/devtype** will all be saved and restored after 2.0 is installed. Your termcap and printcap files are saved with the extension **.OLD** if you need to refer back to them. There were some files that are not protected and may need to be restored manually after the installation. These include /**.profile**, customized getty's, and most importantly, the Global Menu files. Although the program will attempt to refresh your Global Menu to restore all of your programs to the menu, any customizations you have performed may be lost. We suggest that you make a copy of the /**m/menu/control** directory onto a floppy disk before installing For:Pro 2.0. If you need to resurrect your Global Menu, you can do it by copying the files global.mnu and menu.file back onto your system. When ever working with system files such as these, we suggest moving the existing versions of files to another name before overwriting them.

Operation

The most substantial change in the operation of For:Pro 2.0 is a greatly improved menu system for changing device connections. It is now possible to attach modems for both incoming and outgoing calls with relative ease. Graphic representations of the devices connected help to clarify the correct connections for peripherals.

Most of the other operational changes in For:Pro 2.0 will be fairly transparent to users. One obvious fix to the floppy drive eliminates the brief floppy L.E.D. flash when a floppy is inserted. When you change the date, instead of having to answer yes that the date is correct, you can just press EXECUTE. Finally, the echo and test commands have been incorporated into the shell to speed execution of shell scripts. There may be some other changes in the user interface, but they are relatively minor.

New Features

The additions to For:Pro 2.0 have been listed in past issues. They include:

1. Incorporating Fortune:Link 1.1 and Distributed File system;
2. Support 3.5 MB Memory
3. Support four CommA-6 controllers
4. SCSI Command Set Support for QIC-36 (Tape), ST-506 (Disk) and Embedded SCSI Disks
5. For:Frog — Optimized for Disk Performance
6. Expanded Fixed Disk Bad Block Sparing
7. Enhanced Device Connection Menu
8. Enhanced Line-Printer Spooler

Performance

We haven't used For:Pro 2.0 long enough to make significant comments on the performance improvements of this new release. Fortune states that there is a 10% improvement in random disk access which should aid performance. In addition, the **For:Frog** program has been enhanced to reduce disk fragmentation.

One comment that a dealer made to us is that there may be a price to be paid for these improvements for certain users. As a result of the changes to For:Pro, the kernel has gotten bigger, and will require more memory. If your computer is operating just under a threshold in memory (e.g. you have just enough memory to run your applications without swapping) the larger kernel may put you over the edge. If this is the case, you will need to add more memory to your system before you will gain from For:Pro 2.0. In fact, you may see some degradation until the memory is added.

You should also verify that your existing software will operate properly with For:Pro 2.0. For instance, a new BASIC interpreter is needed to work with the new version, as well as a new copy of the Extended Laser Interface. Other than these comments, we know of no other reason not to upgrade to 2.0.

Final note:

In the press announcement for For:Pro 2.0, Fortune has included the following paragraph:

In the near future the operating system will be sold as an option to allow users and resellers a choice between For:Pro and UNIX System V for the Fortune 32:16 series. With the release of UNIX System V users and resellers will have an opportunity to select the operating system best suited to their application requirements. Fortune Systems will continue to support For:Pro after the addition of UNIX System V to the product line.

System V is probably the wave of the future (although it may affect the Formula line more than the 32:16 line). We cannot comment on the implications of this at this time, however, we will keep you posted as more information becomes available.

Josh Lobel

Product Releases from Fortune/SCI

The following are several product releases we received from Fortune/SCI recently regarding Databases that have been released.

Informix Products

INFORMIX-SQL based products on the Fortune Formula and 32:16 under FOR:PRO have been released. Forming a cohesive group of interrelated database products, the offerings include:

INFORMIX-SQL: a flexible relational database management system based on the proposed ANSI-standard SQL.

INFORMIX-4GL: an integrated fourth generation application development language, based on SQL.

INFORMIXR-ESQL/C: complementary tools for embedding SQL statements in "C" and accessing "C" functions from INFORMIX-SQL.

File-it!: an interactive file management system for creating and maintaining simple databases and applications, built on C-ISAM.

C-ISAM: the de facto standard Indexed Sequential Access Method (ISAM) subroutine library for the UNIX operating system, and the foundation of INFORMIX-SQL based products.

UNIFY

Unify is a high-performance relational database management system. It is a transaction-oriented database, specifically engineered to optimize performance in transaction-oriented applications. The Unify DBMS provides the tools required to create and maintain a database, as well as a wide range of tools for application development. The product consists of a menu handler, screen painter, query-by-forms, report generator, and structured query language (SQL). These tools were designed with the user in mind, making them flexible and easy to use.

Other features of Unify include a 'C' interface, variable length text, the ability to store the database over more than one device, and custom message control, making internationalization of an application as simple as changing a string file. This latest version of Unify also includes a binary data type, and text data type. These unique features give users the ability to store both digitized data, including voice and graphics, as well as large documents such as manuals, within the database.

Unify is available in a Full Development System, which gives the user the full power to design and implement a database application, and a Runtime System, which offers the sole capability of running an existing Unify application.

32:16 Version. Fortune also has available Unify version 3.2 for the Fortune 32:16. Although this is not the latest version, the product is similar to 4.0 with the exception of the binary and text data types, the variable length text, and the ability to store the database over more than one device. Version 3.2 also has a less advanced version of custom message control. This version of Unify should provide sufficient functionality for the majority of existing 32:16 customers.

PROGRESS

Fortune/SCI announces the addition of the PROGRESS DBMS, version 3.2, to the Fortune Formula product line. PROGRESS is a relational database managememnt system designed specifically for application development. It offers Fortune users the functionality usually found in only major mainframe products. Progress is called a "fourth generation" product because the functionality is more advanced than that of standard third generation languages such as Cobol, Basic, or 'C'. The product consists of five key components: a relational DBMS, a Data Dictionary, an Application Language, a Screen and Report Formatter, and a Procedure Editor. Together these components create a flexible, highly productive environment to build and utilize sophisticated applications.

The PROGRESS DBMS is a fully relational database that was designed to support high-performance transaction processing. It is known as a "crash-proof" system as it contains unique multi-user concurrency and recovery failure.

The PROGRESS Screen and Report Formatter makes report writing an easy task. It includes automatic formatting for default screens and reports, automatic data validation, plus support for the user's own validation procedures. The PROGRESS Procedure Editor provides the user with an interactive application building environment. A full-screen syntax-checking editor, an on-line HELP facility, and extensive use of function keys all combine to create a productive, user-friendly environment.

To complement the PROGRESS full Development System, Data Language also offers a Developer's Toolkit, which includes utilities to help the developer manufacture and distribute their PROGRESS application. Also available is the PROGRESS Runtime System and Query/Runtime System. The Runtime System provides the sole capability to run a PROGRESS application, while the Query/Runtime System is simply a Runtime System that includes additional utilities allowing the user to write their own queries.□



Dear Fortune Owner:

As you have probably read in */u/fortune news,* Fortune Systems SX models are significantly more powerful and faster than the PS and XP models. If you have a PS or XP model system, you should take advantage of our special promotion on SX upgrades. For a limited time, and while our supplies last, we will upgrade your system, including the new SX Cabinet at no additional charge for the cabinet. With this special offer, you may choose any combination of the following upgrades and/or expansions:

- *SX Processor with Current Release of Multi-User For:Pro*
- *60 MB Internal Streaming Tape*
- *30 MB, 45 MB, 70MB High Speed Hard Disk*
- *6-Port Intelligent Asynchronous Controller*
- *Upgrade to Current Release of all Fortune Software*
- *Fortune:Windows (Included at NO Charge)*

Additionally, we can perform these upgrades within 24 hours of receiving your system. In fact, if you ship your system overnight on a Friday for Saturday delivery, we will have it back to you on Monday morning, all ready to plug in and run; not losing the use of your system for a single work day. And, we will issue you a full 90-day warranty on the entire system.

JFM provides services to Fortune Systems owners either directly or through our authorized Affiliated Dealers. If you haven't used our services, I invite you to discover JFM; you will be glad you did.

Sincerely,

Morris Frydman
President

Talk to us, then talk to our clients

JFM
BUSINESS
SYSTEMS
The Merchandise Mart
Lobby Suite 144
Chicago, Illinois 60654
Call (312) 644-0870

Authorized Fortune Systems Master Dealer

FORTUNE SYSTEMS
an SCI Systems Company

No matter what your product or service, staying competitive requires control

Let QBC, your company's one source computer supplier, provide your business with solutions to:

Order Entry	Payroll
Accounts Payable	General Ledger
Accounts Receivable	Sales Analysis
Invoicing	Legal Time Billing
Inventory Control	Manufacturing Software
Office Automation	(MRP II Plus)

QBC is your one source for hardware, software, maintenance, customer training and data conversion.

MASTER DISTRIBUTOR FOR UPSTATE NEW YORK AND ERIE, PENNSYLVANIA

QBC

QBC Systems Inc.
375 NORTH FRENCH ROAD, AMHERST, NEW YORK 14120
(716) 691-5201 1-800-443-5770

DO YOU KNOW...

there is an organization that has a thorough knowledge of

FORTUNE BAS
MULTIPLAN IDOL

We have been providing widespread support to our Fortune clients for more than 4 years.

If you need:

- training
- support
- maximum computer performance
- peace of mind relating to processing
- customization
- application results
- minimum errors and confusion
- better ideas on how to increase the power of your Fortune

Call Steve Rosenfeld at (314) 731-3636

SUPERIOR COMPUTER SYSTEMS, INC.
917 N Hwy 67, Suite 105
P.O. Drawer T
Florissant, MO 63031
Making computers work for you

Coming soon...Rich Suslick, now of Cedar Cliff Systems, begins a series of articles about how to keep your hard disk and file system in good order...More on Fortune:Word Glossaries...Dave Kloes continues his System Adminstrator's series...A review of TimeMaker, the Desktop Utility for the Fortune...New free software disks, and more.

/u/fortune news

73 Jason Street
Arlington, MA 02174

Address Correction Requested

/u/fortune* news

The Newsletter for Users of Fortune/SCI Computers | August 1987/Volume 4 Number 8

Indexes, Bullets, and more

More on Glossaries

Last month we described some **Fortune:Word Glossary** basics and a couple of the glossary entries provided on the glossary disk provided with **Fortune:Word 3.0**. In this issue we will continue in this vein by examining some more glossary entries included on this disk. Please note that even if you don't have **Fortune:Word 3.0** program, these entries can be used with **Fortune:Word 2.0** — the only difference is that you'll have to type them in yourself.

Another comment is in order. Although many of these examples may be just perfect for your needs, they may require slight alterations. We will try to provide enough information for you to do that, and the **Glossary User's Guide** from Fortune provides much additional instruction. We encourage you to get your feet wet and experiment with glossaries.

Creating an Index

The first glossary we will discuss is on page 5-7 of the **Glossary User's Guide**, and is displayed in **Figure 1**. It is entry a in the **gloss2a** glossary found on the **Fortune:Word 3.0 Glossary Disk**.

This entry is used to copy words from your document onto the Workpage, and then return you back to your document in the same position. It could be used for creating a variety of lists, although it is probably most useful for creating word lists for indexes. Once you have copied all of the words to the Workpage, you can copy the words into a "word list document" which can be used with the **Index Generator** feature of **Fortune:Word**. (In order to do this you would use the `SHIFT COPY` feature to copy between two documents.)

The comment lines, in each figure ,describe how the entry will work.

Let's look at the entry. The comment lines explain how the entry works. Line 1 and 2 are the label and opening bracket which are generic for all entries. Line 3 creates what is called a variable, which in this case is called **title**. A **variable** is a slot where the computer can stick some information,

See **Glossary,** page 19

Figure 1

```
1  entry a /*in Chapter 5*/
2  {
3          title = word          /*read the word at the cursor location*/
4          command note          /*mark the cursor position in the document*/
5          goto "w"              /*go to the workpage*/
6          goto down             /*go to the bottom of the workpage*/
7          call feed(title)      /*insert the word saved in the variable*/
8          goto note             /*go back to the bookmark cursor position*/
9  }
```

Creating an Autodialer

A 6-line program that dials your phone!

Most users who own modems know that they can issue a command to the modem and have it dial a phone number. In most circumstances, they are dialing up another modem at a remote location. However, it is possible to use a similar technique to make your modem dial the phone for a voice conversation. We'll show you the basic technique in this article.

In order to create the autodialer, we use something like the following command: (We'll spruce it up in just a moment.)

```
echo -n "ATDT617 663-7700^M" >
  /dev/cul4
```

This is the guts of the program. We

See **Autodial,** page2

In This Issue...

All you ever wanted to know about file systems, Rich Suslick begins a new series on File System Management...**Page 3.**

The /etc directory and related files in our continuing series on System Management...**Page 11.**

ipl files and a *negative* error 102 in the BASIC Advisor...**Page 16.**

More on Glossaries...**Page 1.**

A Phone Autodialer...**Page 1.**

*/u/fortune *news* is published by The Cambridge Consortium, Inc. It is not affiliated with Fortune Systems/SCI Technology, Inc..

/u/fortune *news*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz

Technical Editor
Scott Bradner

Advertisting Manager
Mark Palmerino

/u/fortune *news* is published 12 times per year by The Cambridge Consortium, Inc. Neither **/u/fortune** *news* nor The Cambridge Consortium are affiliated with Fortune Systems/SCI Technology, Inc.

Subscriptions cost $48 for one year within the U.S.A. For foreign subscriptions, add $27 each year for added postage and handling costs.

Back issues are offered at:

Vol 1, Numbers 1 - 3	$1/each
Vol 1, Number 4	$.50/each
Vol 1, Numbers 5 - 8	$3/each
Vol 2, Numbers 1 - 7	$3/each
Vol 3, Numbers 1 - 12	$4/each
Vol 4, Numbers 1 -	$4/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $10 each. Please add $5 per order for foreign destinations.

Although every attempt is made to insure that our contents are error-free and without risk, The Cambridge Consortium is not responsible for losses that occur due to correct or incorrect interpretation of **/u/fortune** *news*.

For more information, call or write:

/u/fortune *news*
73 Jason Street
Arlington, MA 02174
(617) 663-7700, (617) 646-5603

The text for this newsletter was prepared using a Fortune 32:16 for text input. The information was then transferred to an Apple Macintosh Plus with kermit and formatted for typesetting using the Aldus Pagemaker program.

UNIX™ is a trademark of Bell Laboratories, Inc. Fortune 32:16™, Fortune:Word™ and FOR:PRO™ are trademarks of Fortune Systems/SCI Technology, Inc. Multiplan™ and MS-DOS™ are trademarks of Microsoft. IBM-PC™ is a trademark of IBM. Macintosh Plus™ is a trademark of Apple Computer. Pagemaker™ is a trademark of Aldus Corporation.

©1987 by the Cambridge Consortium, Inc. 8/87

Autodial

continued from page 1

assume you have a "Hayes compatible" modem. These modems dial the phone in response to the **ATDT** (ATtention Dial Tone) command. What we want to do in our shell script is to send something like "ATDT617 663-7700" out to the modem which will start it dialing. Our command begins with the command **echo -n**. The echo command simply spits things out to the screen. The **-n** tells it not to send a line return after it echoes. This is necessary because if a return is sent, it cuts off the modem before it has a chance to dial. First we tell echo to spit out "ATDT", the dial command. This is followed by our phone number. (It gets awfully quiet around here in the summer.) The ^M is actually a single character, a CONTROL-M, which is simply a RETURN without a line feed. This tells the modem to dial the number without hanging up instantly. When you go to create the shell script, you probably need to do something special to enter the ^M — in **UNIX**, **ed** or **vi** you will need to precede it by a CONTROL-V. In **sc** or **screen**, you use a CONTROL-C. The > sign tells the computer that we want to redirect the output from **echo** to **/dev/cul4**. **/dev/cul4** specifies the port that we have the modem connected to. In your case it might be **/dev/tty04** if you are only using your modem to call out, or it might be **/dev/tty94**. The **tty** or **cul** name is arbitrary, but it has to coincide with the port name that you use to call out on your modem. If you are unsure of the name of the port, give us a call and we'll try to help you find it out.

Now this command isn't very flexible the way we've written it, because it can only be used to call our own number. You could create lots of commands for calling all of your friends, but it's much easier to use a variable name in the command. Look at the following command line:

```
echo -n "ATDT$1^M" > /dev/cul4
```

Here instead of our phone number, we've substituted $1. This refers to the first argument that follows our shellscript name. Suppose we put the line above into a file called ph and made it executable (chmod +x ph). If we entered ph 617-663-7700 then our phone number is $1. Note that this will only work if there aren't any spaces in the phone number, because each space defines a new argument, e.g. $2, $3, etc. To be safe we should probably make our command look like this:

```
echo -n "ATDT$*^M" > /dev/cul4
```

All we have changed here is the $1 to a $*. $* refers to all of the arguments to the command. That way if there are any spaces in the phone number, it won't make any difference — the whole thing will get echoed out to the port. We have now perfected the core of the autodial command.

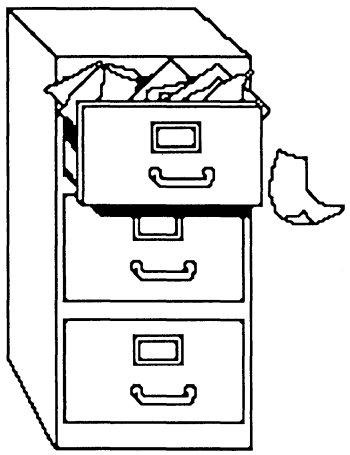
In order to really make the command work it has to be included in a somewhat more complicated shell script. Here's what that looks like:

```
1  sleep 30 > /dev/cul4 &
2  echo -n "AT^M" > /dev/cul4
3  echo -n "ATS7=10^M" > /dev/cul4
4  echo -n "ATDT$*^M" > /dev/cul4
5  sleep 15
6  echo -n "AT^M" > /dev/cul4
7  echo -n "ATZ^M" > /dev/cul4
```

When you have a modem set for both incoming and outgoing calls, it is necessary to "tie up" the port for the duration of the shell script so that it remains in the outgoing mode. This is what the **sleep** command does. The 30 means it will be busy for 30 seconds. Line 2 echoes out an AT (Attention) just to make sure the modem is ready to accept the commands. Line 3 instructs the modem to only wait 10 seconds for a carrier before hanging up the phone. Without this command the modem and speaker will stay on the line while you are talking, resulting in some unpleasant effects. Line 4 is our old friend that was described in detail above. Line 5 waits for 15 seconds. Line 6 sends out the command ATZ, which resets the modem to its original state (like turning it off and on), which resets the carrier detect time to the default which is about 30 seconds. This is the way it should be set to call another modem. That's all that we have in our dialer.

See **Autodial**, page 22

Essentials of UNIX File System Maintenance



by Richard Lee Suslick

This is the first in a series of articles written by Richard Suslick on the internals of the UNIX file system. This series of articles addresses both the hardware and software aspects of how UNIX organizes disk allocation. Later articles will address how to correctly use ***fsck****, the UNIX system's file system check routine,* ***ncheck****,* ***icheck****, and* ***fsdb****.*

Rich is president of Cedar Cliff Systems Corporation which is located in Washington DC, (Alexandria, VA). He provides UNIX training, professional software and management consulting. Mr. Suslick specializes in database and communication design issues and has had 7 years experience with Fortune Computers and 12 years experience with UNIX.

INTRODUCTION

Most UNIX users have come to appreciate the hierarchically organized tree structure of files and directories under the UNIX operating system. What many people fail to realize is that the operating system is constantly performing a series of complex operations to maintain this elegant facility.

To understand what **fsck** and any of the other file system maintenance routines do, it is first necessary to understand both the terminology and structural components being used by the operating system.

First let us look at the building blocks physically available on each disk drive. Appropriately, each of these units are actually called BLOCKS. Under Fortune's (or SCI's) version of UNIX, FOR:PRO, each block consists of 512 bytes (characters). Under newer UNIX System V releases these chunks have been increased to 1024 bytes.

Why 512 bytes? Well, mostly because of the hardware involved. The evolution of 5-1/4 inch winchester disks (your hard disk) was such that this was a convenient size of data to move in one I/O operation. The controller in the Fortune was, until recently, designed by Western Digital, and was compatible with most of the available disk drives of its time. These drives were known as ST506 compatible drives. The model number ST506 was one of the first 5-megabyte drives and the interface became a standard (with some later expansion and improvement).

What Does a Disk Drive Look Like?

This type of disk drive consists of a series of BLOCKs laid out in concentric circles on a number of platters. Picture a stack of records on a record player with a gap between each record and a set of tone arms reading from both or just one side of each of the record surfaces at once.

The major difference with this analogy is that records consist of just one spiral. Disk drives consist of concentric magnetized rings. Like the record's groove, each of these concentric rings is called a TRACK. One more thing: each of the "tone arms" (actually called read/write HEADs) are all tied together like your fingers are to the palm of your hand. Unlike your hand, the HEADS all move in and out on the platters together (move your hand flat at the wrist and you will see how heads move). Since all heads are moving in and out together and are over the same concen-

tric ring on each platter at the same time, we call the area addressed by all the HEADS over a set of TRACKS a CYLINDER.

To add to this confusion each platter's surface is divided into pie sections around each concentric circle. The number of sections a track is divided into is usually 16 or 17 on the Fortune. Finally, each of these sections, call SECTORS, consists of 512 bytes. Ah, now we see the connection between a BLOCK and a SECTOR. Under the FOR:PRO system a BLOCK is a SECTOR of the disk.

A typical 20 megabyte disk drive might be designed like:

Disk ID: C20 - 20 Mega byte
Sectors/track: 17
Heads: 8
Cylinders: 320

The term HEADS refers to the number of platter surfaces that are read/written and available on the drive.

With all this information we can now compute the number of blocks on a disk drive. For the C20 its 17 * 8 * 320 = 43520.

Why 512 bytes? Well, mostly because of the hardware...each sector on a disk consists of 512 bytes.

You might notice that 43519 blocks * 512 bytes/block = 22.28 megabytes not 20 megabytes. For a 70 megabyte drive this is actually 17 sectors/track * 8 heads * 1024 cylinders = 139263 blocks or 71.3 megabytes. This is the actual amount of space available on the drive. How much is available to the user is part of the next topic.

For our use, blocks are numbered from 0 to the end of the drive, e.g. 43519 for the 20 megabyte drive described above. Unfortunately, the hardware does not address the drive this way. The hardware must still address the disk by specifying which head, track, and sector(s) it wishes to read or write. Because of this THE SYSTEM MUST BE ABLE TO DETERMINE THE CONFIGURATION of the drive before it can access the drive randomly.

To determine the drive's configuration, all drives can be addressed to read the first block on the drive, e.g. block 0 is head 0, track 0, sector 0. This block has special value; it contains all the other information about the hardware and logical configuration of the drive. Hence, the system during the *boot* (power up) process goes out to each drive on the system and reads the configuration of the drive for future use.

Under most UNIX systems block, 0 is referred to as the BOOT BLOCK of the drive. Fortune has usually referred to it as the CONFIGURATION BLOCK. Those of you who have had the unfortunate experience of receiving the message "`Drive n Has Gone Bad`" during the boot processes have seen the result of losing or garbaging the configuration information at the begin-

Our Memory Doesn't Cost You A Fortune

Would you like to:

Speed up your computer?
Add more users?
Use the latest software, like Fortune:Windows™?

Compatibility:	The F1MB is absolutely 100% compatible with Fortune's own 1 Meg memory board.
Affordability:	USA is proud to have equaled and exceeded the quality and performance of our competition, while offering substantial savings for both the dealer and end user.
Dependability:	All boards are burned in for 72 hours and tested for 48 prior to shipment. Our One Year Repair or Replace Warranty underlines our confidency in the product.
Repairability:	The F1MB facilitates repair in the field as it is 100% RAM socketed. Now you can repair a board by simply replacing a defective chip rather than face an expensive board exchange -- reducing maintenance costs and system down time.

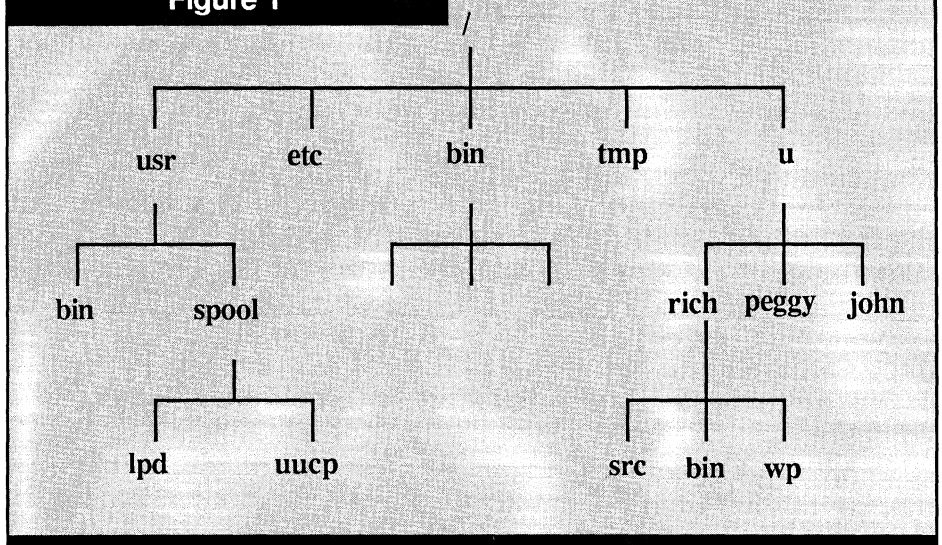
If you have a Fortune SX, XP45 or recent model XP30, you can use our board by simply plugging it in. Older systems can be upgraded to use our board with a simple and economical modification.

For Details Call: ***Upgrade Service Associates***

850 Lexington Street, Suite 15, Waltham, MA 02154 Phone: (617) 239-8147

Our Guarantee: If you're not completely satisfied within 30 days, USA will gladly refund your money.

Figure 1



A user's perspective of the File System Structure

ning of the disk. Often, this block may be restored by an experienced UNIX wizard without significant loss of data on the system.

Software Disk Addressing Under UNIX

From here on, we begin to address the software configuration of a disk drive. Under the Fortune configuration, the next 50 to 70 (first track boundary greater than 47) blocks are used for diagnostic and hardware sparing. Hardware sparing is a facility by which the system uses one of the "spared" blocks at the front of the disk as a substitute for an unusable block somewhere else on the disk.

Unusable blocks are defective areas on the drive. Usually, this is caused by defective manufacturing where the magnetic media on the platter is imperfect. Typically, most all drives have a few. The older 20 megabyte drives usually have 2 to 8. The newer 45 and 70 megabyte drives have 15 to 30. The system can only handle up to 45 bad blocks and track 0 of the drive is always guaranteed to be error-free from all manufacturers.

While the specification of which blocks are bad is usually a manual activity, FOR:PRO has the ability to provide *automatic* block sparing. Once spared, the substitute block is automatically read by the hardware/software of the system. This process will be addressed in a later article.

After this section of blocks, an optional BOOT area of the disk may be specified. This boot area contains a program loaded before the UNIX operating system that actually reads the hardware configuration of the machine and determines how and where to load the rest of the UNIX operating system (*kernel*) and how to start it up. The "kernel", as it is often called, is that part of the UNIX operating system which remains in the computers memory at all times.

The remaining sections of the disk are composed of user specified logical sizes called partitions. Each partition is viewed by the UNIX system as a *logically* separate disk. Unless reconfigured, FOR:PRO utilizes only two or three partitions. The first partition is always the configuration block, spare block area, and the boot file area, if one existed. The second partition is used as a *swap* area. This is the area the computer puts programs temporarily when the system is using more program space than there is real memory. Finally, the last partition is used for the user/system data/program space.

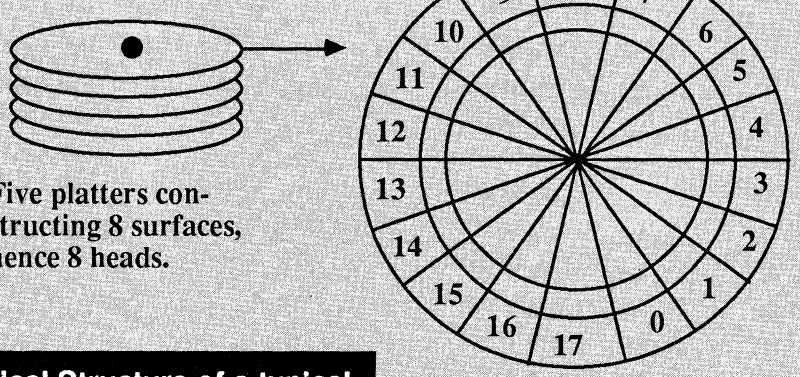
Logical Disk Software Design (Partitions)

This last partition is the portion of the disk which is used for what most users perceive as the directory/file system. Up to eight partitions may be made on a disk. There are advantages and disadvantages to splitting a disk into more partitions. These subtleties will be discussed in another article.

The specification of which blocks are bad can be performed automatically and efficiently by FOR:PRO.

Each logical partition is divided into four sections: the BOOT BLOCK, the SUPER BLOCK, the I-NODE section, and DATA BLOCKS. In reality the last two sections of the partition, the i-nodes and the data blocks, are interspersed through out the partition. For the purpose of this discussion, however, you may picture the three sections as consecutive chunks of the partition.

Figure 2



Physical Structure of a typical Fortune disk drive.

On the Fortune, the BOOT BLOCK is unused in all partitions except partition 0, where the BOOT BLOCK is the configuration block for the drive discussed above. The SUPER BLOCK contains the configuration information for this partition, e.g. size of the partition, number of i-nodes, and some information about the free space available in the partition. This block if damaged or destroyed is much harder or nearly impossible to reconstruct.

The SUPER BLOCK is read by the UNIX kernel whenever the device is mounted. For the root partition this is done automatically during the startup procedure. This SUPER BLOCK is kept in an internal buffer of UNIX and kept in machine memory (RAM) at all times. The SUPER BLOCK contains the information the kernel needs to access files in this partition's file system.

The INODE section of the partition (sometimes called the I-LIST) consists of a set of blocks whose purpose is to act as an index (hence the I in inode) for the remainder of the partition. Eight I-NODES are stored in a single block and are sequentially numbered from 0. Each INODE consists of 64 bytes. The maximum number of I-NODES in a single partition on most UNIX systems is 65535. Due to a bug in the FOR:PRO system this number has been limited to 32767 in a single partition.

An inode contains important information about a file that is essential for the proper operation of your computer.

What is an INODE? Why INODES? Some systems contain their directory information in one place. The names of all the files and controlling information are kept in a single directory area. Expansion of files, random access to the contents of a file, and appending to a file are restricted by other "flat" file systems structures. Noncontiguous files are easily handled by UNIX and not easily handled by other disk allocation structures.

Each file under UNIX is defined by a single INODE which contains the size of the file, id numbers of the owner and group, the physical location of the start and end of the file, time stamps of creation, modification, and last access. An INODE DOES NOT CONTAIN THE NAME OF THE FILE. File names are stored in a directory which is a data file with a special structure, but a data file none the less.

A file, including a directory file, is a set of blocks pointed to by an INODE. Of the INODE's 64 bytes, 13 two-byte entries make up the complete pointer list for a file. You should be asking yourself at this point, if you're still following: how can all the blocks necessary to allocate a 5 or 10 megabyte file be specified with only 13 pointers in the INODE?

We are the *largest* Authorized Fortune Master Dealer in the Pennsylvania and Delaware area. As part of the **P**rofessional **S**olutions **G**roup of companies with over 35 years in business automation, we have the *stability* that our customers demand

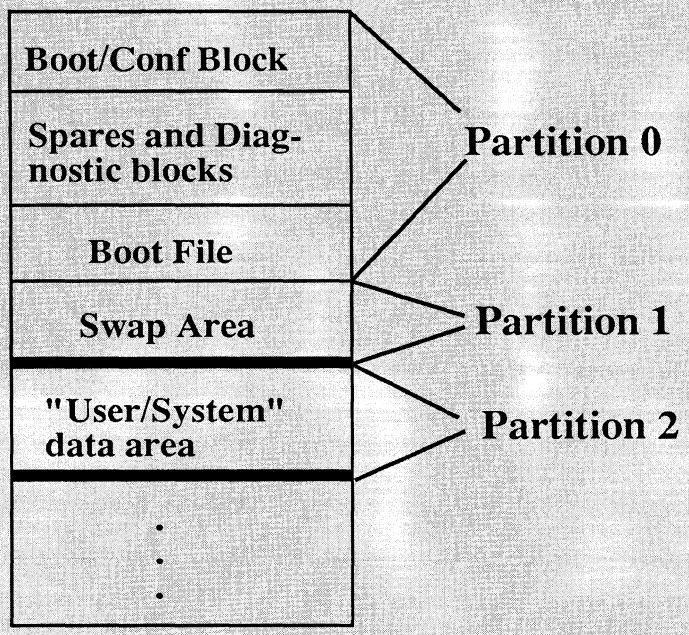
Our services include a factory trained field engineering team, a complete parts inventory including system upgrade capability and a software support department.

For all of your Fortune needs contact:

Mark Shields of PSG at
2550 Eisenhower Avenue
PO Box 762
Valley Forge, PA 19482

or you can call Mark at (215) 666-1400

Figure 3



Physical Layout of a Disk

Actually, only the first 10 block pointers point to actual disk blocks containing file data. The next 3 blocks point to data blocks that contain more pointers. The eleventh block points to a block that points to the next 128 blocks in the actual file. This block is called an indirect block. The twelfth block points to a block that points to up to 128 indirect blocks.

This kind of block is known as a double indirect block. That is, it points to a block of blocks that point to the file. Finally, if the file exceeds 16,522 blocks (10 + 128 + (128 * 128)) a triple indirect block is created which then points to 128 double indirect blocks. Thus the maximum file size is 2,113,674 blocks (a laser disk maybe?).

Obviously, the task of fetching information from a huge file is rather significant. Because of the buffering the UNIX kernel provides, however, this overhead has an amazingly small impact on using large files. The flexibility of the hierarchy and expandability gained by such a structure usually outweigh the price for such large file lookups.

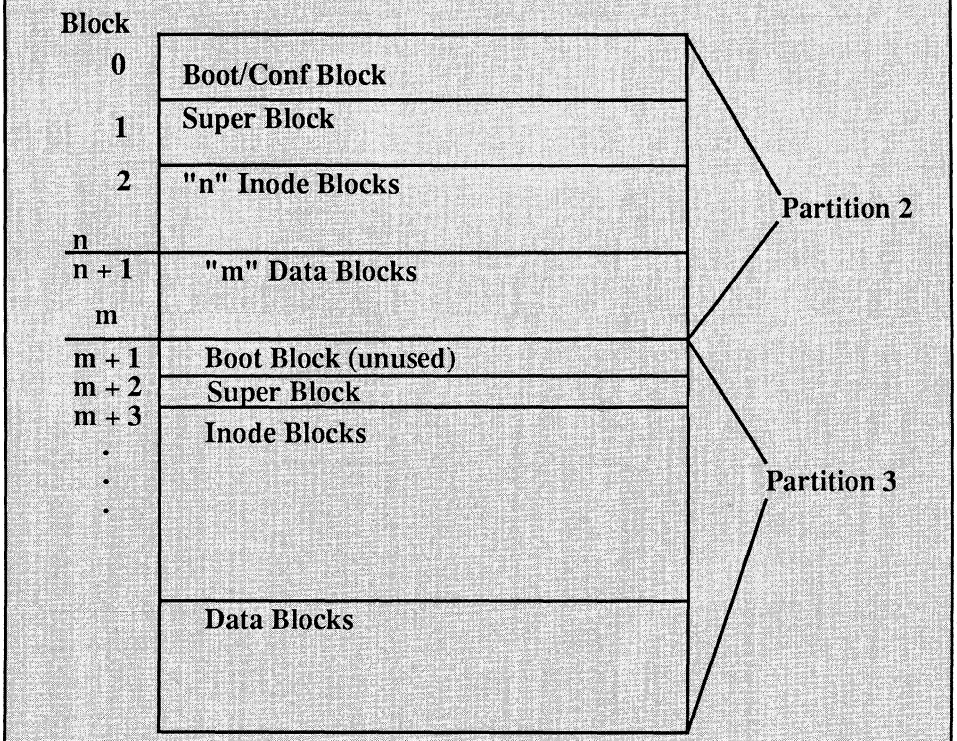
Directories Are Separate Concepts from INODE Allocations

Finally, we arrive at the concept of a *directory*. Above, you have seen how blocks of a file are stored in inodes and indirect blocks. Now we will address how directories are stored. First directories are files like any other file in the system except only the UNIX operating system may modify or create them. They may be read as a file by users but never changed except under operating system control.

Directories under FOR:PRO consist of files containing a block or a series of blocks consisting of 16 byte entries.

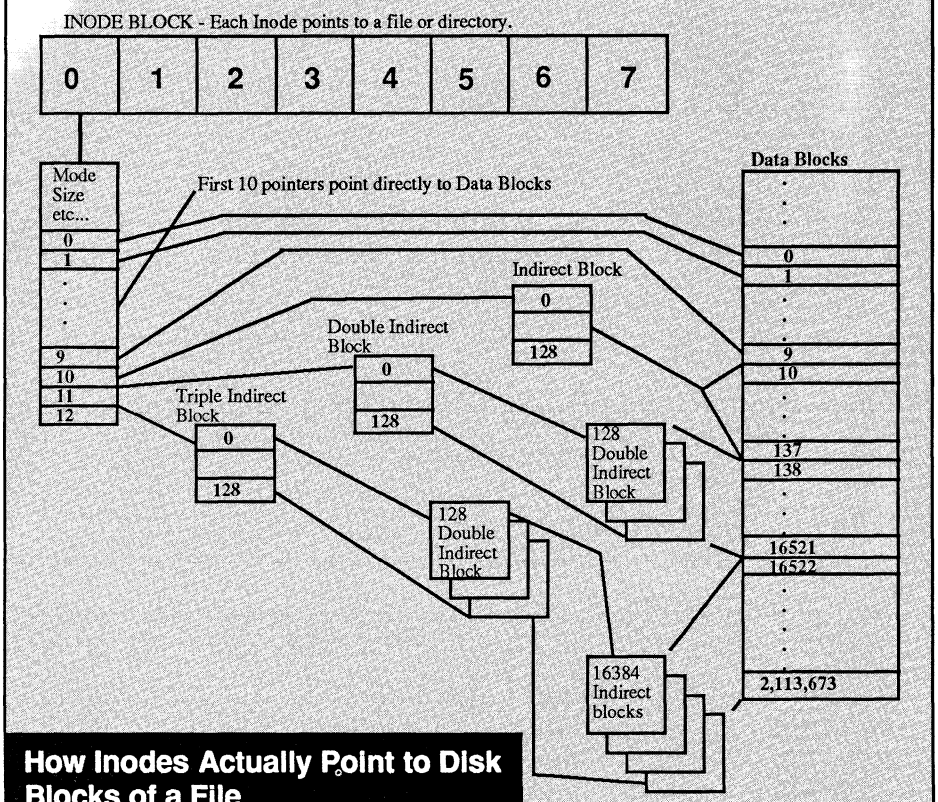
Directories under FOR:PRO consist of files containing a block or a series of blocks consisting of 16 byte entries. Each 16 byte entry consists of 14 bytes for the name and 2 bytes for a pointer to the INODE of the file. There are two special entries in every directory, '.' and '..'. These two file pointers provide pointers to the directory itself and to its parent directory, respectively. The

Figure 4



The Logical Segments of Partition

Figure 5



How Inodes Actually Point to Disk Blocks of a File

root directory, which has no parent directory, has a '..' entry that points to itself as well.

To find a file, the system searches through the directory looking for the name with that segment of the path. It then continues by reading the file pointed to by the inode of the new segment of the path, etc., until it arrives at the final entry of the directory containing the file. Thus, to find /usr/lib/uucp/xyz the systems first reads the root directory searching for the 'usr' entry. Then it uses the inode information to find the directory file 'usr' and searches for 'lib'. Then it uses the inode for the 'lib' entry to find the directory file 'uucp'. Finally, it finds the entry 'xyz' in the 'uucp' directory and locates the inode for the file 'xyz'. The inode is then read to find the permission, size, etc. as described above, and the file is ready for use by the user.

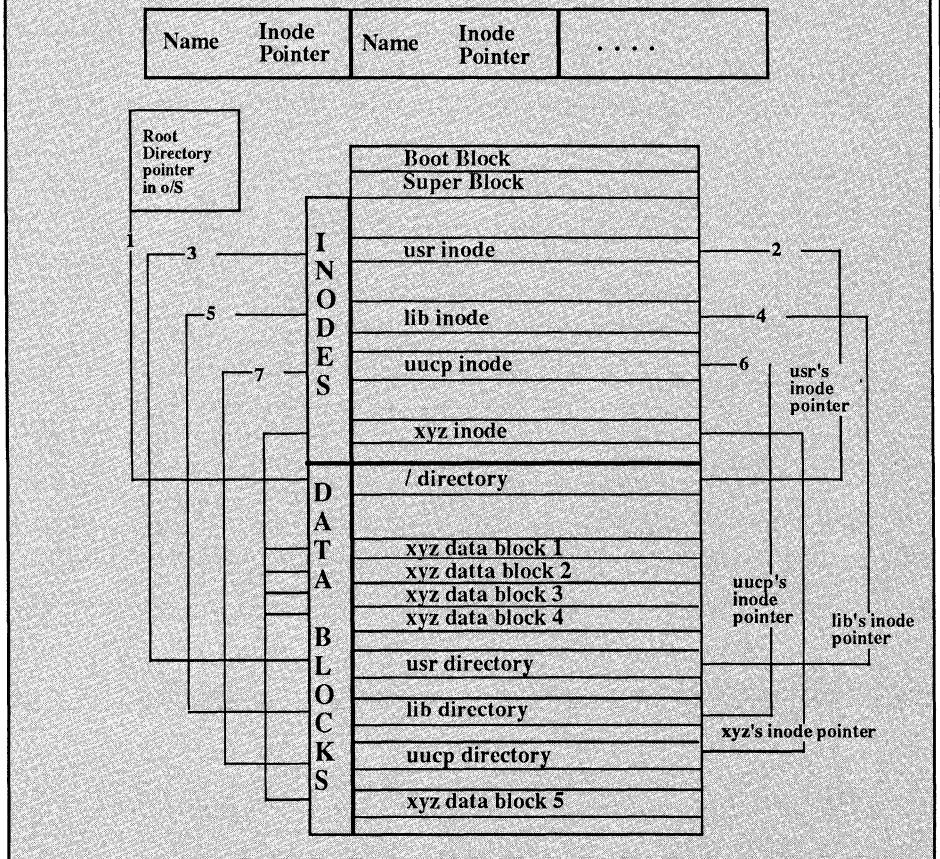
This is why we have the programs: **fsck**, **icheck**, **mkconf**, **mkfs**, **ncheck**, and some others. The file system is a rather complex labyrinth of pointers and files, eloquently interwoven.

One more point of consideration should be made. Much of the file system information is kept in operating system buffers in memory. The positive side of this form of file management is an obvious improvement in speed for file manipulations. The negative side is that if the system should stop abruptly due to a power failure or other problem, the pointers left in the files and inodes on the disk will not resemble the reality of the world.

My next article will address the routines fsck, icheck, and ncheck, as administrative devices for repairing damaged file systems.

Richard Lee Suslick
Cedar Cliff Systems Corporation
6036 Old Telegraph Road
Alexandria, VA 22310
(703) 960-2999

Figure 6



How Directories are Accessed

*** See the NEW product DOS LINK ***

SOFTWARE FOR THE FORTUNE 32:16

NEW PRODUCT!!! DOS LINK ($495.00) - provides easy-to-use menus and programs that transfer data between any PC (including Laptops) running DOS and the Fortune system. Here are just a few examples of how this product can be used:

- Transfer Multiplan or Fortune:Word Documents from or to your DOS system.
- Transfer other ASCII or binary files.
- Use the PC or Laptop as a remote terminal on the Fortune system.
- Transfer THOROUGHBRED (SMC) BASIC data that has been extracted using one of our other products

Documentation and software for both the DOS and Fortune system are provided. Data may be transferred via a direct connection (9600 baud) between the two computers or via modem (at modem speed). You may also order additional hardware for your PC that may be required to accomodate the transfer.

If you are interested in purchasing a Laptop computer that can be used with this product for people on the road or at home, give us a call!! We also provide consulting services for interfacing your DOS system to the Fortune ($50/hr).

MULTIPLAN LINK - can pass data from BAS/IDOL/BASIC files to Multiplan and back. For example, it could be used to send data from any of the BAS files (Chart of Accounts, Customer Master, Inventory Master, etc). Allows selection of fields to be sent and provides logical retrieval and key range selection. **$295.00**

RECORDS PROCESSING LINK - can pass data from BAS/IDOL/BASIC files to Records Processing and back. For example, it could be used to take customer address data from the Customer Master and will automatically make the Records Processing List document for mail/merge. Allows selection of fields to be sent and provides logical retrieval and key range selection. $295.00

ASCII LINK - can pass data from BAS/IDOL/BASIC files to ASCII and back. Use to send data between machines or databases. For example, it could be used to send data from any of the BAS files (Customer Master, Inventory Master, etc). Allows selection of fields to be send and provides logical retrieval and key range selection. **$295.00**

TERMINAL/PRINTER LINK - provides capability to print data on a printer connected to a workstation. Can print BAS/IDOL/BASIC reports, Multiplan and Fortune:Word documents. BAS/IDOL reports can also be viewed before printing, converted to Fortune:Word format; accumulated for mass printing; backed up to diskette; printed on any system printer. An option to print BAS/IDOL reports on a LASER PRINTER is $50 extra. Many other options are included. **$195.00** (Fortune:Word printing on a printer connected to a workstation is $50 extra)

TELEPHONE LINK - on line telephone message, telephone directory and inter-office mailbox/menu system. Telephone message entry resembles typical pink slip. Telephone directory can search by name or company. Mailbox/message system can send and receive Fortune:Word documents or Multiplan documents. Does much more! Give us a call. Written in Business Basic. **$295.00**

CALCULATION LINK - provides programs for amortization, depreciation, loan repayment, averaging (with graph), linear correlation and breakeven analysis. Written in Business Basic. **$95.00**

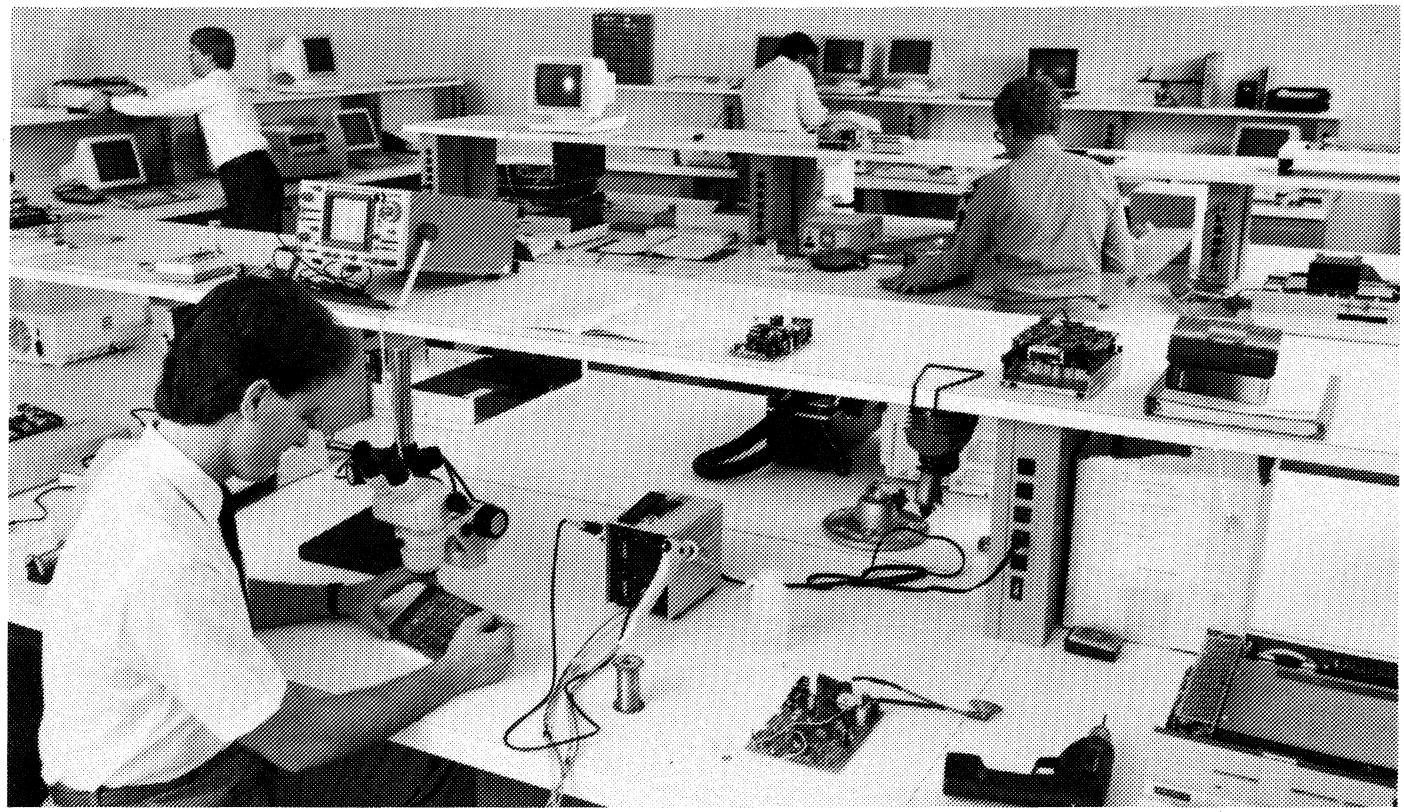
KOMPACT PERSONNEL ACCOUNTING - provides for data capture of personal, wage/salary, job, education, salary/wage, appraisal training, dental, medical and life insurance data. Includes over 80 pre-defined reports/worksheets. Written in Business Basic/IDOL. **$995.00**

PRINTER LINK - provides printcaps for Okidata 84, 2410, TI 855, Anadex 6500 and Printronix. Includes shell script for changing pitch and LPI. Allows these printers to be used in Fortune:Word. **$95.00**

MODEM LINK - allows Hayes 1200/2400 modem to be used on one port for both incoming and outgoing calls. **$95.00**

WESTERN LINK - provides instruction on how to use your Fortune for sending/receiving telex messages using Western Union's Easylink. Can replace your telex! Fortune ITE or Handshake software required. **$95.00**

UNI-KOMP 9803 Rocktree Houston, Texas 77040 (713) 937-7782



The Best Just Got Better

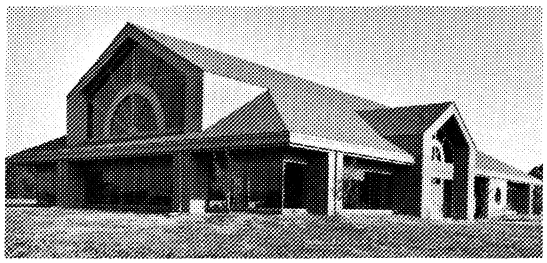
At Micro Resources, our commitment to Fortune/SCI is clear. We've just opened a brand new facility to make us even more competitive and service-oriented for Fortune Systems users.

Expanded Service
Not only have we added service personnel, our newly-equipped service department is simply the most advanced in the mid-west. That means fast turn-around for system upgrades or repairs.

Expanded Inventory
Service can only be as fast as the needed components are available. That's why we've enlarged our on-hand inventory. Volume purchases also keep our prices competitive.

Expanded Upgrades
All this growth has led to a landslide in upgrade business. That volume has provided us with a wealth of components and systems available in limited quantities at fantastic values.

Expanded Classrooms
Now Micro Resources has the space and integrated training facilities to host dozens of employees. Our training programs are tailored to your needs and your schedule.



We believe in Fortune/SCI. And we believe in supporting Fortune/SCI users with the finest service and product at a competitive price. So, if you're unable to find a resource to meet your needs, call Micro Resources. Our dedication to Fortune Systems and to complete service is simply unsurpassed.

Call us at 1-614-766-2335. In Ohio, call toll free 1-800-622-UNIX.

Authorized Fortune Master Dealer

Micro Resources

255 Bradenton Avenue, Dublin, Ohio 43017

BEYOND TECHNOLOGY . . . *RESULTS*

System Administration: Part 4

No Longer Requires a Degree in UNIX Wizardry

This series is a must for all owners and users of the Fortune computer. Dave is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the Fortune Computer, and is president of the Houston UNIX User's Group. He contributes independently to ***/u/fortune news.***

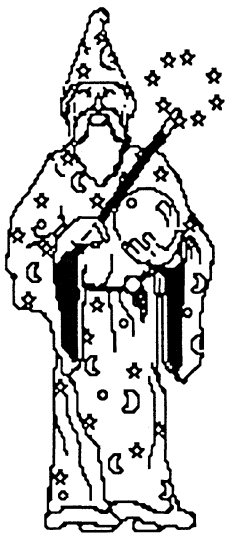
by Dave Kloes

In our last article, we said we would be discussing one of the most important directories on your system - **/etc**. It is an important directory because it contains many of the administrative files and commands used by your system. Figure 1 is a list of the files and commands that are initially loaded into this directory.

Many of these files and commands are critical. If a "critical" file or command were missing, the operation of the system would be severely inhibited. Most of the files and commands loaded by the "Cold Boot Set" are critical files. Let's divert our discussion of **/etc** for a moment to talk about the "Cold Boot Set".

The "Cold Boot Set" of diskettes are those diskettes that are used to initially load your system. For the For:Pro 2.0 operating system, there are three volumes of Cold Boot diskettes. The files and commands on these diskettes can be loaded onto your system in three different ways, depending on what you are trying to do. We will be talking about these procedures in more detail in future articles. Let's take a brief look at them here:

1. The diskettes are initially loaded onto a blank hard disk by using the "Cold Boot Procedure". This procedure is used to initially load your hard disk with the minimum files and commands required to start up the system. This procedure would be used when your system is first installed or if a hard disk is replaced.



This month Dave talks about some of the most important system files - those found in the /etc directory. Also, terminal ports and Comm A boards are discussed.

2. The Cold Boot Set can also be installed using the "**S5**" (install) selection on your global menu. This might be useful, for example, to restore critical system files or commands that have been lost. This method is also used to "upgrade" your system from a previous version of For:Pro to version 2.0. The system must be able to start up in order to use this method.

3. Indivdual files or commands may be restored by "mounting" the diskette and copying the files directly onto the hard disk. We will discuss this when we talk about the "mount" command. This method assumes that the system can be successfully "booted" (or started) and that you can login.

4. Individual files or commands may be restored by bringing the system up in "single-user" or "maintenance" mode. This method assumes that the system cannot be successfully "booted". There are at least two ways of doing this which we will discuss later.

The most important comment we need to make here is that because the "Cold Boot Set" contains your critical system files and commands, you should al-

ways have them available in case any problems arise. If you do not have these diskettes, we recommend that you make a set using the "S5" selection on the Global Menu. After you have selected "S5", select "backup" and then "Cold Boot" from the "Product Maintenance Menu". As the system administrator, you should have a backup available for each of the products listed on the "Product Maintenance Menu", in addition to the Cold Boot Set. If you don't, now is the time to back them up. Many times we have gone in to fix a problem that required re-installation of a product and found that the client did not have the original copy of the Fortune product diskette or a backup copy that was made using "S5".

Enough for now about the Cold Boot Set. The first file we will be discussing in the **/etc** directory is "devtype". This file contains most of the information pertaining to your system devices (terminals, printers, etc.). It is automatically modified whenever you use "**S2**" **39** (Define Device Connections) from the Global Menu. If this file were lost, you would have to redefine all of your terminals and printers using "S2" 39. If you execute the command "`cat /etc/devtype`", a typical entry might look like:

```
tty01   DB630   P    1    m    Diablo Printer   1
tty02   FT      T         m    Sharon           1
```

Each of these entries has the parameters that are normally set up when you use "Define Device Connections".

Defining devices in FOR:PRO 2.0 is one of the exciting new changes that was made to FOR:PRO 1.8 or 1.8.1.1.

These parameters are device name or port (tty01), device type (DB630), device class (P = printer, T = Terminal), device number (1 — In this case, the 1 stands for printer #1. If you have other printers on your system, you would see #2, etc.), baud rate (m=9600), comment (Diablo Printer), and state (1 = enabled). Looking at this file in For:Pro using the "**cat**" command to view your device status is much quicker than going through the menu system. As a sidenote, if you are logged in as "root" or "manager", you can define your devices by executing the command "/m/sysman/devconn" (For:Pro 2.0 only). This would be the same as selecting "S2", 39 through the menu system.

Defining devices in For:Pro 2.0 is one of the major changes that was made compared to For:Pro 1.8 or 1.8.1.1. If you are not familiar with how to do this through the Global Menu, details are in "Fortune 32:16 Series Set Up Guide" which you should have received with your 2.0 Cold Boot Set. Also, the "Multi-User" diskette is no longer required for For:Pro 2.0.

We don't want to assume that each of you understands the parameters that are listed above. So, for the beginning system administrator, we will elaborate on some of the terms and concepts.

MULTIREG

Multiple Regression Program

A Professional Econometric Estimation Program For The Fortune Systems 32:16 Computer.

MultiReg is a professional tool, designed to provide detailed output of Multiple Regression Analysis, on the Fortune 32:16.

MultiReg generates an exhaustive array of statistical results.

*Regression Coefficients
*Correlation Coefficients
*Standard Errors and t-values
*Actual and Predicted Values of Dependent Variable
*Values of Residuals
*Means of Variables
*Analysis of Variance Table
*F Ratio
*Variance-Covariance Matrix
*Graph of Residuals
*Autocorrelation Coefficients
*Zero-Order Correlation Matrix
*Standard Error of Estimate

Order Your Copy Today

Econometric Forecasting, Inc.

6971 N. Federal Hwy, Suite 102
Boca Raton, Florida 33487-1648
Phone: (305) 994-6373

Dealer Inquiries Welcome.

Retail Price: $495.00

MultiReg contains a Variable Transformation Menu, which allows for 12 different types of variable transformations to facilitate the estimation of non-linear reltionships.

MultiReg has the power to run literally dozens of regressions with one stroke of the Return Key. One example in the Reference Manual is the estimation of an econometric model. You simply type one command line, calling MultiReg and directing the output to the Printer. After you press the Return Key, you might as well go out to lunch, because it will take a 20MB Fortune 32:16 ten full minutes to crank the numbers, and after that it will take a Qume 945 Printer 55 minutes to print the resultes. The printed output will be 120 pages, containing detailed results on 33 stochastic equations with 128 estimated regression coefficients.

The MultiReg Package consists of two Disks, in a durable plastic library case. Disk 1 of 2 is the Training Version of MultiReg, which operates in "Interactive Mode". Disk 2 of 2 is the Standard Version of MultiReg, which operates in "Batch Mode". A clear and concise Training Manual and a comprehensive Reference Manual are provided.

Figure 1

```
# cd /etc
# ls
altconbaud      flopchange      mid             profile         shutdown
bootcp          format          mkconf          pstat           surftest
chgrp           fsck            mkdevs          rc              termcap
chlog           fstab           mkfs            rc.pass1        uconf
chown           getty           mklost+found    rc.pass2        umount
devtype         group           mknod           rc.real         update
disk            halt            mount           rdconf
disktab         init            mtab            reboot
disktab.orig    login.help      passwd          sdisktab
dskselect       lprserv         printcap        setdt
```

Each "device" on the system has a name so that For:Pro knows what you are referring to. In particular, if you look at the back of your machine, the ports are named as follows:

tty02 tty04
tty03 tty05
tty01 (printer port)

If you were looking at your system from the back and it had one 4 port "**Comm A**" board, the top left "port" is tty02. The one under that is "tty03". The port in the center at the bottom is called the "printer port" or "tty01". The piece of equipment that has the serial ports on them is called a "Comm A" board. Fortune currently has three types of Comm A boards: one that has six ports, one that has four ports, and one that has two ports. Each "port" allows you to attach one device (terminal, printer, modem, etc.) to it. If you had a six port board, the ports would be:

tty02 tty05
tty03 tty06
tty04 tty07
tty01 (printer port)

If you had another 6, 4 or 2 port Comm A board, it would go in the third slot and the devices would continue with "tty08", "tty06", or "tty04", respectively. Normally the first slot contains the "video controller" board which the console plugs into and the second slot would have your first Comm A board.

A Comm A board can be added to your system whenever you need to add more devices to your system. Let's look at a typical scenario. Your system has one 4 port Comm A board. You currently have a printer attached to tty01. You also have three terminals (in addition to the console) attached to ports 2-4 and you have a modem attached to port 5. With this configuration, all of your ports are being used. You now decide to add another terminal to your system.

The Cold Boot Set of diskettes are those diskettes that are used initially to load your system. They contain critical system files and commands.

Let's look at some of the possibilities available to you as the system administrator.

1. Let's assume that the modem is only used when your favorite consultant needs to call in to fix a problem for you and that money is a factor. One thing we could do is to put a "switch" box on port 5. A switch box allows you to use one port for two different devices and to switch between them as needed. We run a cable from port 5 to the "I/O" connector on the switch box. We then run one cable from switch box to the modem and one cable from the switch box to the new terminal. A switch on the front of the switch box allows us to select either the modem or the terminal. In this configuration, we would normally leave the switch positioned to the terminal. Whether we are using the port as a terminal or a modem, the "Define Device Connections" entry would be the same with the exception of the baud rate (speed or bits per second). A terminal will normally will be set to "19200" baud and a modem at either "1200" or "2400" depending on the modem type. The higher the "baud rate", the faster the device will operate. If our consultant needed to dial in, we would set the switch to the modem position and change the baud rate to "1200" or "2400", as appropriate. You would give up the use of the terminal while the modem was in use. After the modem work is complete, you would reposition the switch and set the baud rate back to "19200" for the terminal.

2. If that scenario is too much of a hassle, you can add another Comm A board to the system. This would allow you to add the additional terminal and would give you extra ports for future expansion.

Let's look at the price of each of these boards:

Device	Cost	Cost per port
Comm A 2 port	$345.00	$173.00
Comm A 4 port	$495.00	$124.00
Comm A 6 port	$930.00	$155.00

The 2 port board costs the least but has the highest cost per port. It also would only give us only one additional port. The cost per port of the 6 port is higher than the 4 port but would give us an additional two ports compared to the 4 port. Regardless of price and cost per port, we could select any of the three boards for our current solution.

3. The decision of what to do gets more complicated when all of the slots are full. If we are currently using all of the Comm A slots and need to add additional capacity, there are still a couple of alternatives available. We can always replace a Comm A 4 port board with a Comm A 6 port as a solution. This may be good or bad depending on whether our dealer will give us any credit for exchanging the 4 port board

for the 6 port board. There is one more alternative to keep in mind. We could go to a "consoleless" system. Although this may be a more costly alternative, it may be the only alternative if you must have additional ports. The theory of the consoleless system is that we will change the configuration of the system so that the first slot, normally used by the video controller can be used for a Comm A board. In this scenario, the "console" terminal would be removed and replaced with a regular terminal and would be plugged into "tty01". This would mean we no longer need the video controller board since we don't have a "console" to plug into it and since that is all it is used for. We would move the printer, which is normally attached to "tty01" to one of the ports on a Comm A board. Obviously, this would require that we purchase another Comm A board and another terminal to replace the console. Again, your dealer may or may not give you credit for the console and the video controller which now become useless for our system. We would also need to purchase (if they weren't already installed) 1.8 Mom Rom chips to support the consoleless system. Our cost for this alternative, assuming no trade-ins would look like this:

ASCII Terminal	$695 (old type) or $995 (new type)
Comm 6	$930
1.8 Mom Roms	$195

In this scenario we chose the Comm 6 to provide the maximum expansion capablility. The Comm 6 is always the better alternative if we anticipate additional devices. Any other alternative would mean wasting a previous board that was purchased or trying to make a deal for a trade-in.

While we are at it, let's talk about the difference between the 4/2 port boards and the 6 port board. The 4/2 port boards use the standard 25 pin RS232 connectors. The reason there are more ports on the 6 port board is that it uses smaller connectors with fewer pins. The additional cost of the Comm 6 includes adaptor cables and extenders. The extenders are required because there is not enough room to plug a cable in if the Comm A is put into the first slot (since the cabinet was not originally designed for this). If we choose to put a 4/2 port board into the first slot, the extenders are an additional $25.

That completes our discussion of Comm A boards. There are two more files in the **/etc** directory that are used for defining devices on our system - "ttys" and "ttytype". These are the normal files you will find that define devices on other UNIX systems. The "devtype" file is a Fortune innovation that makes defining devices "user friendly". The "ttys" and "ttytype" files are automatically updated whenever we "Define Device Connections". The "ttys" file is used to tell whether a device is a login or non-login device and what its baud rate is. A typical entry would look like:

```
1mtty01
0ntty02
```

The first character is a "0" or "1" which indicates a non-login or login device, respectively.

Fortune/SCI currently has three types of Comm A boards: one that has six ports, one that has four ports, and one that has two ports.

A "login" device is one that a user would login to. A terminal or a modem that someone can call into are login devices. A "non-login" device is one that a user would not login into. A printer or a modem that is used for calling out on are non-login devices. The second character indicates the baud rate. Here are some typical baud rates:

m = 9600	n = 19200
i = 1200	k = 2400

In this example, "tty01" is a login device defined at 9600 baud and "tty02" is a non-login device defined at 19200 baud.

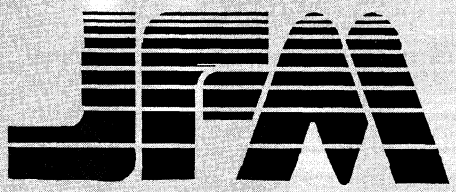
The "ttytype" file tells what type of device is attached to each port. A typical entry would look like:

```
FT      tty01
unknown tty02
```

In the **/etc** directory, there is a file called "termcap" which has the terminal capablity codes for different types of terminals that can be attached to the system. In fact, if you execute the command "more /etc/termcap", you will see that the first terminal definition is for "FT" which is the Fortune terminal. Other terminal codes are in this file as well. Each terminal has its own particular way of doing things. For example, the sequence to tell one terminal to underline something is usually different than another terminal. The "termcap" file would tell For:Pro (UNIX) what that sequence is for each terminal. The "ttytype" example above tells the system that the terminal attached to port "tty01" is a Fortune terminal (FT). Since port "tty02" was defined as a non-login device, that means the system doesn't need to know what type of device is attached and therefore says "unknown". Just as terminals are defined in the "termcap" file, printers are defined in the "/etc/printcap" file. Printer types are specified in the "devtype" file, unlike terminals which are defined in the "ttytype" file.

Next time, we will continue our discussion of the /etc directory and the "termcap" and "printcap" files. We will also be discussing the basics of three For:Pro editors - **ed**, **vi** and **screen**. (**Editor's note**: **Screen** is available on both the **D.C. Grab Bag** diskette and the new **Editors' Choice** diskette that we are announcing in this issue).□

Please write and tell us what you think of this series of articles. Is it helping you work with your Fortune equipment more effectively? What would you like to see covered in more detail. We appreciate *your* feedback. Send your responses to us at The Cambridge Consortium, Inc.



Dear Fortune Owner:

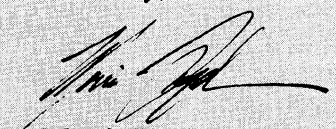
As you have probably read in */u/fortune news,* Fortune Systems SX models are significantly more powerful and faster than the PS and XP models. If you have a PS or XP model system, you should take advantage of our special promotion on SX upgrades. For a limited time, and while our supplies last, we will upgrade your system, including the new SX Cabinet at no additional charge for the cabinet. With this special offer, you may choose any combination of the following upgrades and/or expansions:

- *SX Processor with Current Release of Multi-User For:Pro*
- *60 MB Internal Streaming Tape*
- *30 MB, 45 MB, 70MB High Speed Hard Disk*
- *6-Port Intelligent Asynchronous Controller*
- *Upgrade to Current Release of all Fortune Software*
- *Fortune:Windows (Included at NO Charge)*

Additionally, we can perform these upgrades within 24 hours of receiving your system. In fact, if you ship your system overnight on a Friday for Saturday delivery, we will have it back to you on Monday morning, all ready to plug in and run; not losing the use of your system for a single work day. And, we will issue you a full 90-day warranty on the entire system.

JFM provides services to Fortune Systems owners either directly or through our authorized Affiliated Dealers. If you haven't used our services, I invite you to discover JFM; you will be glad you did.

Sincerely,



Morris Frydman
President

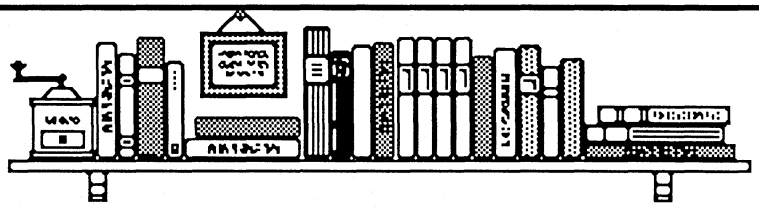
Talk to us, then talk to our clients



JFM BUSINESS SYSTEMS
The Merchandise Mart
Lobby Suite 144
Chicago, Illinois 60654
Call (312) 644-0870

Authorized Fortune Systems Master Dealer

The BASIC Advisor



The BASIC Advisor is brought to us from Ray Wannall. Ray is President of BaSiC Software Corporation which is located in Baltimore, Maryland. This month Ray discusses an upcoming Concept Omega conference, BAS sales and support, and finally Ray helps a reader from the U.K.

Question: *We have just purchased a custom accounting package (not BAS or Thoroughbred) written in Business BASIC and are getting character map overload errors when we try to enter the applications. We are currently running the new Business BASIC release 6.5.12, and the new applications are in the /b/BWORK sub-directory. What gives?*

Answer: Many custom packages on the market are shipped with their own ipl files, and some do not enable the BUTIL sub-directory as a disk device. I assume this is an exercise in vanity on the part of the manufacturer: why should their perfect software require utilities! Well, maybe their software doesn't need the utilities, but the BASIC does. Here is the way a pure dipl file should look:

```
CNF 1,9,1,19,,0
PTN 1,20000
DEV D0,1,8000,,,,,/b/BUTIL
DEV D1,1,8000,,1,,,/b/BWORK
DEV T0,7,3216,,,,,tty
DEV LP,4,3500,,,,,lp
DEV P2,4,3500,,,,,lp
DEV P3,4,3500,,,,,lp
DEV P4,4,3500,,,,,lp
DEV P5,4,3500,,,,,lp
DEV P6,4,3500,,,,,lp
IPL 1,2,T0,**PSD
```

In the first line (beginning "CNF"), the second number (9) represents the number of DEVices enabled, or number of DEV lines in the dipl file. In this pure example, DEVice D0, or Disk Zero, is enabled to the sub-directory /b/BUTIL, while DEVice D1 is enabled to /b/BWORK. Other DEVice lines define Terminals (T#) and Printers (LP and P#). To see your dipl files, log onto the Fortune 32:16 as root, and at the # sign enter: `cd /b/ipls` <CR>

The <CR> means press the RETURN key after typing in the command. Then you can type: `ls dipl*` <CR>

This will display the names of the dipl files contained in the directory. They are normally diplT0 through diplT7. You can check the contents of diplT0 with the command:

`more diplT0` <CR>

Here is an example of a diplT0 file which will cause the character map overflow error you mentioned:

```
CNF 1,10,1,19,,0
PTN 1,20000
DEV D0,1,8000,,1,,,/b/DBASE
DEV D1,1,8000,,1,,,/b/PROGS
DEV D2,1,8000,,1,,,/b/FILES
DEV T0,7,3216,,,,,tty
DEV LP,4,3500,,,,,lp
DEV P2,4,3500,,,,,lp
DEV P3,4,3500,,,,,lp
DEV P4,4,3500,,,,,lp
DEV P5,4,3500,,,,,lp
DEV P6,4,3500,,,,,lp
IPL 1,2,T0,MAINMENU
```

To correct the overflow error, we must add DEVice D3 after the DEV D2 line. We can use the "ed" editor as follows:

`ed diplT0` <CR>

PROGRAMMERS

The leading manufacturer of status display panels for the fire alarm and security industry is seeking programmers and junior programmers to develop new applications and refine existing programs.

Successful candidates will be experienced in IDOL/BAS and SMC Business Basic for Fortune and Basic 4 systems. Familiarity with UNIX and system set-up and maintenance is desirable.

Candidates interested in joining a growing team-oriented organization located in Falls Church, Virginia should write or call with resume and salary history, to:

WSA
308 Hillwood Avenue
Falls Church, Virginia 22046
703 241-0687

```
1p                                  <CR>
```

The computer lists the first line of the file: CNF 1,10,1,19,,0

```
s/10/11/p                           <CR>
```

The edited line is displayed: CNF 1,11,1,19,0. We have increased the number of DEV lines in the control line from 10 to 11. Now we need to add the new line:

```
5a                                  <CR>
DEV D3,1,8000,,,,,/b/BUTIL          <CR>
.                                   <CR>
```

The last entry was a period. What we have done here is add a line after line 5, which was the definition of DEVice D2, or Disk Two. You may examine the new file contents with the command: `1,$p` <CR>

The new file should look line this:

```
CNF 1,11,1,19,,0
PTN 1,20000
DEV D0,1,8000,,1,,,/b/DBASE
DEV D1,1,8000,,1,,,/b/PROGS
DEV D2,1,8000,,1,,,/b/FILES
DEV D3,1,8000,,1,,,/b/BUTIL
DEV T0,7,3216,,,,,tty
DEV LP,4,3500,,,,,lp
DEV P2,4,3500,,,,,lp
DEV P3,4,3500,,,,,lp
DEV P4,4,3500,,,,,lp
DEV P5,4,3500,,,,,lp
DEV P6,4,3500,,,,,lp
IPL 1,2,T0,MAINMENU
```

To exit the editing session and save the changes, type a period, a "w" and a "q", entering <CR> after each command:

This will put you back to the # prompt. Repeat the procedure for all of the other dipl files (diplT1, diplT2, etc. through diplT7). When you are finished, you may exit UNIX with CTRL-D (hold down the grey CTRL key and type the letter D). Then call the software manufacturer and complain. But do not give them the fix: let them figure it out.

Question: *Why doesn't the following BASIC command work:*

```
IF D5$=H1$ THEN LET L7=I1*A
```

When I print the values of D5$ and H1$, they are the same on the screen, but the value of L7 was not changed after the line was executed.

Answer: Try printing the LENgths of each string variable:

```
PRINT LEN(D5$)              <CR>
PRINT LEN(H1$)              <CR>
```

I'll bet you dollars to doughnuts they are different. One of the two has a trailing space or two. Adjust the length of the longer one and the value of L7 will be properly updated. Isn't it a shame we do not see spaces on the screen?

Question: *Have you received your new 6.5.12 Business*

UNISOFT INC.

Experts in Computer Consulting - Sales - Service

Presents

FORTUNE SYSTEMS HELP

We Provide:

- Expert Trouble-shooting
- Years of programming and accounting experience
- *Free Initial Consultation*
- Training Available for:
  - * BASIC/IDOL/BAS and Multiplan
  - * UNIX and Ext. Fortune:Word
- Hardware Repairs & Upgrades
- FORTUNE trained experts at a cost that will surprise you!
- *Trade-Ins Welcome*
- Specializing in:
  - * Hard Disk Trade-In
  - * Hard Disk Upgrades

Authorized Fortune Systems Repair Depot
Fortune Systems National Value Added Reseller (VAR)
CALL: (303) 671-8600 EMERGENCY HELP: (303) 649-8505

UNISOFT INC. 14100 East Jewell Avenue Suite 27 Aurora, Colorado 80012

BASIC yet? I have found a very unusual error in the new version. It seems that if you have a WAIT statement in a program and hit the ESCape key during the WAIT you get no error. However if, after the ESCape, you delete the WAIT line and RUN the program you get an error -102 and the program halts. You can solve the problem by deleting the original WAIT line and typing in RUN. I learned this while programming one morning in the early hours.

Answer: I have just received and installed Business BASIC 6.5.12, FOR:PRO 2.0 and the new 3.0 Fortune:Word. Needless to say, I have not yet had a chance to test them all. I have noticed, however, that the new BASIC is faster and more in line with the Thoroughbred BASIC found on every other computer. I see that the utilities menu has been altered to include the new terminal configurator. This will be a very exciting addition for users with non-Fortune terminals. The Wyse 50, for example, can be configured to accept 132 columns and over 40 lines per screen. The word processor, which I have had limited time to examine, appears to be a great new product. But I'll leave that to other writers in this publication.

As for your negative error 102, my advice to you is to spend the wee morning hours enjoying a good night's sleep. I don't trust any error messages at 3:00 AM. □

DO YOU KNOW...

there is an organization that has a thorough knowledge of

FORTUNE BAS
MULTIPLAN IDOL

We have been providing widespread support to our Fortune clients for more than 4 years.

If you need:

- training
- support
- maximum computer performance
- peace of mind relating to processing
- customization
- application results
- minimum errors and confusion
- better ideas on how to increase the power of your Fortune

Call Steve Rosenfeld at (314) 731-3636

SUPERIOR COMPUTER SYSTEMS, INC.
917 N Hwy 67, Suite 105
P.O. Drawer T
Florissant, MO 63031
Making computers work for you

TRI-TECH COMPUTER SERVICES, INC.
SHARE IN OUR CELEBRATION!!

Tri-Tech Computer Services, Inc. has just completed its first year of a very successful operation. **Thank you** to our patrons, from the Staff.

We are a service organization which offers new and used system sales as well as service to new and existing Fortune products. The staff at Tri-Tech has been providing services to Fortune Computer users for almost four years.

We encourage you to use the following coupons:

Limit one coupon per organization.

COUPON #1
15% Discount on next purchase of Fortune released Hardware.
(Excluding Tri-Tech Specials)
Tri-Tech Computer Services, Inc.
Expires 10/31/87

COUPON #2
20% Discount on next purchase of Fortune released Software.
(Excluding Tri-Tech Specials)
Tri-Tech Computer Services, Inc.
Expires 10/31/87

Summer Specials:

H.P. Series II, LaserJet or LaserJet 500+ (15% off List with this ad)
Micropolis 70MB Hard Disk Upgrade (45% off List with this ad)
Fortune Used Workstations (Best price around, quantity discounts)

SERVICE SERVICE SERVICE
Professional Installations

Kensington, CT (203) 828-9784
Newton, MA (617) 965-4453

Toll Free 1-800-622-TTCS, EXTENSION 333
(Outside Connecticut)

Glossary

continued from page 1

and then recall it later. In this case, **title** is made equal to **word**. **word** is a glossary feature which selects the whole word that the cursor is sitting on, regardless of where the cursor is located. As long as the cursor is on any letter in the word, this glossary entry will work. Line 4 issues the COMMAND NOTE sequence, which puts a marker in the text, so that the spot can be found later. (The COMMAND NOTE feature is very handy even if you're not using glossaries. When you press GOTO NOTE, the cursor will pop back to the same location. In otherwords, it's a very useful placemark.) Line 5 tells **Fortune:Word** to **GOTO** the Workpage of the document. Line 6 sends the cursor to the bottom of the Workpage. Line 7 inserts the contents of the variable called **title**. **feed** is a function that spits out what ever variable is between the parentheses. The word call is always necessary before any function is uses. Finally line 8 instructs **Fortune:Word** to return to the placemark that was created in line 4. That returns you to where you left off.

If you try this entry, you will see that you can see each phase of the entry working. If you want to speed it up some, you can add the lines:

`call display(false)` *insert before line 3*
and
`call display(true)` *insert before line 9*

These two commands are useful for most glossary entries. They turn the display off at the start of the glossary and on again at its completion. Since the screen isn't updated as the work is done, everything happens more quickly.

Setting up Bullets

Many of us use "bullets" to highlight specific points in our text. For instance, with glossaries you can:

- o speed up data entry,
- o improve accuracy of difficult words and/or phrases by using an abbreviation to enter them rather than type them in full each time, and
- o automate complex Fortune:Word functions.

Usually these bullets are entered with tabs and/or indents to keep the text aligned. As the bullet marker, you might either use an "o", a "–", or if you have a laser printer, the special bullet character. As an example, the typical bullet would be `indent o indent`. The glossary entry in **Figure 2** (which is from page 6-2 of the **Glossary User's Guide**, and is **entry d** in **gloss2a** on the glossary disk) assumes that you will put an indent at the start of each bulletted paragraph. The glossary entry inserts the "o" and a second `indent`. Note that you can change the o to any other character you choose between the indents.
This is how it works. **Figure 3** shows the text that we will use the glossary entry for. It assumes that you have set up your paragraphs with a single indent character.

Line 3 is a label called "`repeat`". It could be called anything, but for the

Figure 2

```
1  entry d /*in Chapter 6*/
2  {
3       [repeat]
4            goto indent
5            if(globerr) exit
6            right
7            if(char == "o")
8            {
9                goto indent
10               jump repeat
11           }
12           if(char != "o")
13           {
14               insert
15                   "o" indent
16               execute
17               jump repeat
18           }
19 }
```

sake of readability, "`repeat`" is used. Line 4 searches for the next occurrence of an indent symbol. Line 5 is an example of an **if** statement. The **globerr** keyword checks to see if an error was encountered — for instance no indent was found. If that error is encountered the entry exits. Line 6 moves the cursor one character to the right. If this next character is the bullet symbol, in our case "o" then lines 8-11 will be executed. Otherwise they will be skipped. (Note that in glossary entries you compare two things with the == (equals) or != (not equals) symbols. This contrasts with the = symbol used in **Figure 1** line 3 which is not used for comparison at all, rather it is used to assign the value of **word** to **title**.) The reason for this test at line 7 is to be sure we don't insert bullet symbols where they already exist. If there is already a bullet character there, line 9 searches for the next `indent`. That allows it to skip over the `indent` character that follows the bullet. Line 10 causes the program to jump back to the `repeat` label and continue execution from there. In this case the program jumps back to line 3. Line 12 tests the alternative possibility to line 7. Presumably if the character the cursor is on fails line 7, it will pass line 12. One or the other must be true. In this case the statements in lines 14-17 are executed. These are the lines that actually accomplish our purpose, which is to insert the bullet ("o") and `indent` character for each paragraph. Line 17 sends the program back to the repeat label. This program will continue to circle around and around until finally it doesn't find any more `indents` and it exits at line 5. As usual, you will speed the entry's performance by inserting `call display(false)` after the opening bracket, and `call display(true)` before the closing bracket.

Note that if you plan to use this entry, it will search through your entire document searching for `indents` to replace with bullets. If you have indents used for some other purpose you may have a problem. The search begins at the current cursor location in your document. If you are on page 7 of the document, it will not affect `indents` on

continued on next page

Figure 3

```
 ->o ->  speed up data entry,
 ->o ->  improve accuracy of difficult words and/or phrases by using
         an abbreviation to enter then rather than type them in full
         each time, and
 ->o ->  automate complex Fortune:Word functions.
```

Glossary

pages 1-6, so this may solve your problem. Another solution would be to use some other character other than the `indent` character to mark where your bullets will go, such as the `]` character (`SHIFT 6`). This should be some character that doesn't appear anywhere else in your document. If you do that, then the glossary entry can be greatly simplified using the **Search** and **Replace** function. The idea there is to simply replace all occurrences of `^` with `indent o indent`. Obviously if you use this approach you could do the job manually. However, if you put it all into a glossary entry, you will be able to do the whole thing with two keystrokes. The required entry is shown in **Figure 4**.

Figure 4

```
entry d
{
        REPLACE "^"
        execute indent "o" indent
        execute(2)
        REPLACE
}
```

Connecting to the Autodial Program

Elsewhere in this issue you will find a shellscript that automatically dials your phone. Since you can call any UNIX program from within **Fortune:Word**, you can also call the **ph** program we have created. To do that you would just press `COMMAND (F13) !`. When it says "*`What command`*" you would just enter `ph 617-663-7700`. Before you know it, you'll be calling our office. That's pretty easy, but suppose the phone number is already listed on your **Fortune:Word** screen? It would be nice to automatically "feed" that number to the **ph** command. That's what this glossary entry does. It is not contained on the **Fortune:Word Glossary Disk**, but it's short anyway. The entry is in **Figure 5**.

You may notice some similarities between this entry and the index entry we discussed earlier. Line 3 uses a variable to grab the phone number. Since the **word** command in Glossaries selects the word that the cursor is on, it is important that there not be any spaces in the phone number. Use dashes instead. Line 4 creates the variable **cmd** which contains the name of our shellscript. Since we called it **ph**, that's the name we use here. Any text assigned to a variable has to be enclosed in quotation marks. Line 5 uses a function called **cat** which concatenates two variables. That means that it adds both variables together into one. In our example it adds **cmd** and **phone** together and puts them into a variable called **both**. If the word the

Suppose the phone number is already listed on your **Fortune:Word** screen? It would be nice to automatically "feed" that number to the **ph** command.

/u/fortune news presents...

Two NEW Diskettes!

Our Software Library has just expanded again. See the descriptions below for the *exciting* details!

Editor's Choice Diskette

This disk contains five editors. Included are Ted, Rand (e17) and sc (also known as screen). These editors each have different strengths and are useful for editing ASCII files. Hexed is a special "sc" like editor that is built to edit binary files with ease. Also included is Dired which is an editor that is useful for working with directories in UNIX. Some of these editors support windowing and all of them are essential for the well stocked Fortune computer. Don't get caught without them.

Zip Tease Delight Diskette

This disk contains files that cross reference phone number exchanges with zip codes. We have included some shellscripts that access this data. Also included is a recent program that appeared in UNIX/World magazine: telltime. Enter an area code and the program tells you the current itme in that area. Other data files cross reference Area codes with the State and the time; Countries of the world with Phone Country Codes and the time; and States with zip codes. Never look up an area code or zip code again. Order your Zip Tease Delight Diskette today!

Limited Time Offer! Buy both of these diskettes before October 31, 1987 and pay only $15. Each diskette costs $10 separately. So order TODAY and *save* 25 percent!

cursor was on was `212-555-1212` **both** will equal "`ph 212-555-1212`". Now things get interesting.

Line 6 uses a function called **unixfun**. This function calls a UNIX command, much like the `COMMAND !` sequence does if you're doing it manually. The **unixfun** command will execute the command that is in parentheses. Since **both** equals "`ph 212-555-1212`", that command will be performed. While the phone is being dialed, you won't be able to type anything on your screen. Just wait until the modem hangs up, and everything should be restored to normal.

Figures 6 and **7** are an enhancement to **Figure 5**. **Figure 7** is described on page 10-17 of the **Glossary User's Guide**. We have altered it slightly to prevent it from erasing the entire screen. There isn't space here to explain how both of these entries work, but if you try them out, you'll probably be able to figure out what's going on.

Good Luck — and don't forget to call.□

Josh Lobel

FORTUNE SYSTEMS
an SCI Systems Company

No matter what your product or service, staying competitive requires control

Let QBC, your company's one source computer supplier, provide your business with solutions to:

Order Entry	Payroll
Accounts Payable	General Ledger
Accounts Receivable	Sales Analysis
Invoicing	Legal Time Billing
Inventory Control	Manufacturing Software
Office Automation	(MRP II Plus)

QBC is your one source for hardware, software, maintenance, customer training and data conversion.

MASTER DISTRIBUTOR FOR UPSTATE NEW YORK AND ERIE, PENNSYLVANIA

QBC

QBC Systems Inc.

375 NORTH FRENCH ROAD, AMHERST, NEW YORK 14120

(716) 691-5201 **1-800-443-5770**

Figure 5

```
entry p
{
        phone = word
        cmd = "ph "
        both = cat(cmd,phone)
        call unixfun(both)
}
```

Figure 6

```
entry p
{
        call display(false)
        phone = word
        call i
        cmd = "ph "
        both = cat(cmd,phone)
        bothback = cat(both," &")
        call unixfun(both)
        "\027"
        call display(true)
}
```

Figure 1

```
entry i /*in Chapter 10*/
{
        call display(false)
        linenumber = 5
        while(linenumber < 10)
        {
                call clrpos(linenumber,1,80)
                linenumber += 1
        }
call posmsg(5,26,"\034HD                                    \034ID")
call posmsg(6,26,"\034HD  \034ID                               \034HD  \034ID")
call posmsg(7,26,"\034HD  \034ID  \034HBGLOSSARY IN PROGRESS\034IB  \034HD  \034ID")
call posmsg(8,26,"\034HD  \034ID                               \034HD  \034ID")
call posmsg(9,26,"\034HD                                    \034ID")
}
```

Autodial

Continued from page 1

Using the dialer

The way we use our dialer is to issue a command like `ph 663-7700`. In a few seconds we hear the modem dialing the phone and hopefully a few seconds later hear it start to ring. If it rings, we pick up the phone and prepare to say hello. After a few seconds, the modem should hang up the phone. If the line is busy, there's no need to even lift your receiver. After a few seconds the computer will hang up the line.

One thing that we have conveniently overlooked is how the modem will connect to your phone system. One rather simple method which may work for you is to hold your receiver up to the speaker on the modem as it dials. If you're lucky, the phone will dial itself. If that doesn't work, you'll have to use a direct connection to the phone line. If you work in a small office, then you may already have the modem connected to the same line you ordinarily use. If you have a multi-line phone with a big fat wire, you may be able to use a multi-line connector to switch the phone to the line you want to dial out on. If you have a new fancy phone system with lots of numbers, you may have to consult your phone vendor for the best approach. Theoretically it should be possible to attach a modem to your phone so that it autodials on whatever line you are using.

Activating the dialer from Fortune:Word

In the article in this issue on **Fortune:Word** glossaries, we show how to create a glossary entry that will activate the dialer.

Souping up the program

There are several things you can do to enhance our little program. If you want to be sure that only one person uses the dialer at a time, you could create a dummy file which would be removed when the dialer completed its task. Then you would put in a **test** statement to see be sure the dummy file didn't exist before continuing to dial.

If you use a long distance phone company that requires an access code before or after the phone number, you can incorporate that information in the dial program ATDT echo statement. If you need to dial a 9 to dial out, that can be done the same way, e.g. `echo -n "ATDT9,$*^M" > /dev/cul4`. We have coupled the program to our **Progress** database of all of our clients. Since the database has the address as well as the phone number, we can have the program include the long distance dialing information if the call is out of state, and not include it if it's not. You could even check a list of exchanges that are local calls from your phone. If a grep on the exchange part of your phone number fails, then you could dial a 1 first. By making the program call itself, or by setting up another shellscript which repeatedly calls **ph**, you can automatically redial a busy number. The possibilities are endless.□

Josh Lobel

NJBC

New Jersey Business Computers

Authorized Fortune Systems Master Dealer

Authorized Fortune Service

Authorized Fortune Training Center

NEW JERSEY BUSINESS COMPUTERS has the Largest Affiliate Dealer Network in the New York and New Jersey Metropolitan Area.

For The Most Reasonably Priced TRAINING, SERVICE and SUPPORT Available.

call

301 Route 17 North, Rutherford, New Jersey, 07070

201-438-1410

INTA (U.K.) LTD.

Dear Fortune User,

Please find enclosed a complimentary copy of the **/u/fortune** *news* magazine.

This magazine, of which there are back issues available from May 1984, will keep you up-to-date with the latest Fortune/SCI product announcements, as well as providing many useful tips for users of Fortune application software.

The magazine has been written by end-users for end-users and treats the Fortune system in a way which makes it comprehensible for the majority of people.

We at **INTA** are an *official* U.K. distributor for this magazine and can make a subscription available to you including the back issues.

INTA's distribution of this **/u/fortune** *news* is a further extension of **INTA**'s support philosophy on sales, hardware maintenance, software maintenance and training.

INTA was started in March 1985 with the prime intention of offering specialised services to Fortune users. We do not maintain any other products so as a result all our engineers are very experienced with Fortune products.

Over the past two years we have grown rapidly and can rightly claim to be one of the best Fortune sales and maintenance organisations in the U.K.

We also offer a full range of upgrade services at very competitive prices. These upgrades are done with minimal disturbance to your day-to-day business.

A call to **INTA** could save you time and money, so give us a call right away.

Yours Sincerely,

R. F. Potter
Managing Director

INTA Electronics, LTD.
Berkshire House, High Street
Crowthorne, Berks RG117AQ
England.
Telephone: 0344-778969
Telex: 847102 INTA G.

Authorised Fortune/SCI Systems Dealer and Maintenance Company.

SUPPLEMENT TO ALL VERSIONS OF BAS AT A REASONABLE PRICE, TOO!

JOB COSTING

The Right Tool for All Trades

BaSiC SOFTWARE CORPORATION

For More Information Call or Write...
BaSiC Software Corp.
5201 Powhatan Street
Baltimore, MD 21207
301-448-9460

Available on:
IBM AT & compatibles
UNISYS/SPERRY
Altos, All Levels

Expand your Fortune® without spending a fortune.

New equipment costs big bucks. The Good Fortune Trader sells fully tested burned-in equipment, refurbished to Fortune® standards at prices that won't make waves with your cash flow. Call or write today for a listing of our specials.

THE GOOD FORTUNE TRADER

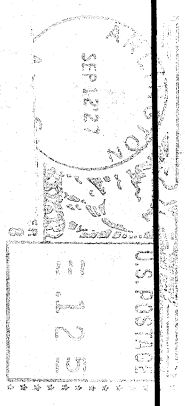
60 DAY WARRANTY
a division of Computer Services of Atlanta
(404) 447-4151
5825 Live Oak Parkway, Suite 2-C, Norcross, Georgia 30093

ForFrog 1.0 does not work with For:Pro 2.0

ForFrog 1.0, which was originally released with For:Pro 1.8, will not work properly once you have installed For:Pro 2.0. ForFrog is a program which reorganizes your hard disk into a more efficient contiguous form by reducing file fragmentation. Use of the 1.0 version with For:Pro 2.0 may damage files on your disk. DO NOT USE FOR:FROG ONCE YOU HAVE UPGRADED TO FOR:PRO 2.0.

The For:Frog program has been temporarily dropped as a Fortune Systems product. It may be replaced by a similar program in the future, but no announcement has been made as yet.

/u/fortune news
73 Jason Street
Arlington, MA 02174



/u/fortune* news

The Newsletter for Users of Fortune/SCI Computers | September 1987/Volume 4 Number 9

Essentials of UNIX File System Maintenance: Part 2

by Richard Lee Suslick

*This is the second in a series of articles written by **Richard Suslick** dealing with the intricacies of the **UNIX** file system. The last article addressed both the hardware and software aspects of how **UNIX** organizes disk allocation. This article deals with how to correctly use **fsck**, the **UNIX** system's **F**ile **S**ystem **C**hec**K** routine. The first part of the article deals with what **fsck** does, while the second part gives some hints on how best to use it to repair your file system. This repair is necessary when your system gives you the message "File check unsucessful. Please contact your dealer.".*

*Rich is president of Cedar Cliff Systems Corporation which is located near Washington DC, (Alexandria, VA). He provides **UNIX** training, professional software and management consulting. Mr. Suslick specializes in database and communication design issues and has had 5 years of experience with Fortune Computers and 12 years of experience with UNIX.*

INTRODUCTION

If you have not read the last article regarding the **UNIX** file system and are unfamiliar with the pointer system used by the **UNIX** operating system, you probably can stop reading here. In some respects, the complexity of the operating system and its device (disk) file structure make it an inhospitable topic for discussion.

I have heard it said that file systems are like children. Minor problems or mistakes can usually be fixed if caught early enough. If neglected, however, the problems compound. Once severely corrupted, they are extremely hard to fix and the damage is inevitably severe.

This article deals with how to correctly use fsck, the UNIX system's File System CheckK routine.

The reasons why file systems get corrupted are many. Usually it is due to a hardware failure, power failure, or improper operation (shutdown) of the machine. Under **UNIX**, the fragility of the file system is compounded by the desire to *cache* frequently accessed chunks of data within the machine's memory. Translated, this means that, in order to minimize the number of times the disk must be physically read, the machine uses its memory to keep changes *as if it was putting the data on the disk.* But the changes will not actually be written onto the hard disk

See **File Management**, page 10

News From Fortune/SCI

Fortune/SCI Signs Three New Dealers

Fortune/SCI announced that it has added three new dealers to its master dealer network. According to Robert A. Davis, the three dealers have committed to total purchases of more than $15.5 million worth of products over the next three years.

The three new master dealers are: **Erin Data Systems and Services**, of Melville NY, for New York City and Long Island; **IST Corporation**, of Redwood City CA, for the greater San Francisco Bay Area; and **PSG Com-**

See **News**, page 2

In This Issue...

News from Fortune/SCI -- New Dealers, New Products...**Page 1**

How to cope with fsck -- or how to repair a "damaged" disk...**Page 1**

The secrets of time and place -- of zipcodes and timezones...**Page 4**

Accounting Questions, IDOL problems -- Ray tackles some BASIC issues...**Page 16**

System Administration -- Dave Kloes discusses the termcap and printcap files, and the **ed** editor ...**Page 18**

*/u/fortune *news* is published by The Cambridge Consortium, Inc. It is not affiliated with Fortune Systems/SCI Technology, Inc..

/u/fortune *news*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz

Technical Editor
Scott Bradner

Advertisting Manager
Mark Palmerino

/u/fortune *news* is published 12 times per year by The Cambridge Consortium, Inc. Neither **/u/fortune** *news* nor The Cambridge Consortium are affiliated with Fortune Systems/SCI Technology, Inc.

Subscriptions cost $48 for one year within the U.S.A. For foreign subscriptions, add $27 each year for added postage and handling costs.

Back issues are offered at:

Vol 1, Numbers 1 - 3	$1/each
Vol 1, Number 4	$.50/each
Vol 1, Numbers 5 - 8	$3/each
Vol 2, Numbers 1 - 7	$3/each
Vol 3, Numbers 1 - 12	$4/each
Vol 4, Numbers 1 -	$4/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $10 each. Please add $5 per order for foreign destinations.

Although every attempt is made to insure that our contents are error-free and without risk, The Cambridge Consortium is not responsible for losses that occur due to correct or incorrect interpretation of **/u/fortune** *news*.

For more information, call or write:

/u/fortune *news*
73 Jason Street
Arlington, MA 02174
(617) 663-7700, (617) 646-5603

The text for this newsletter was prepared using a Fortune 32:16 for text input. The information was then transferred to an Apple Macintosh Plus with kermit and formatted for typesetting using the Aldus Pagemaker program.

UNIX™ is a trademark of Bell Laboratories, Inc. Fortune 32:16™, Fortune:Word™ and FOR:PRO™ are trademarks of Fortune Systems/SCI Technology, Inc. Multiplan™ and MS-DOS™ are trademarks of Microsoft. IBM-PC™ is a trademark of IBM. Macintosh Plus™ is a trademark of Apple Computer. Pagemaker™ is a trademark of Aldus Corporation.

©1987 by the Cambridge Consortium, Inc. 9/87

News

continued from page 1

puters, of Arlington VA, for the legislative, judicial and executive branches of the United States government, Washington, D.C., seven counties in the State of Virginia and Maryland's Montgomery and Prince Georges counties.

AIS signed as Master Dealer

Advanced Industry Systems, Inc., (AIS), has recently signed a master dealer agreement with Fortune/SCI to cover 46 counties in Florida as well as Alabama's Baldwin and Mobile counties.

AIS' president Joel Ingram projects $10 million in sales of Fortune products by AIS over the next three years through AIS' direct office and its excellent affiliate dealer locations.

Specializing in medical, dental and legal systems, AIS now provides these vertical markets with Fortune's full line of multi-tasking desktop products, all of which use the UNIX operating system from AT&T.

"We chose to sign a master dealer agreement with Fortune/SCI for three reasons," says Ingram. "First, because Fortune is a forerunner in the UNIX industry and UNIX is fast becoming the standard operating system in the multi-user computer industry. Second, because Fortune's total family of products is backed by quality and performance. And third, because we feel that Fortune's marketing strategies and planned future growth will bring about their dominant position in the marketplace." Adds Ingram, "All of these factors fit into AIS' game plan to take full advantage of the growing interest and needs of multi-user UNIX-based systems users in the rapidly growing territory covered by AIS' offices and its affiliate dealers.

Ingram's reaction to the recent acquisition of Fortune's computer and marketing operations by SCI Systems, Inc., represents the overall attitude expressed by the majority of Fortune dealers and VARs toward the acquisition. "The SCI purchase is simply icing on the cake for everyone involved," says Ingram, "and it'll bring about an even better future for us all."

Concerning service, Ingram says, "AIS wants to make certain that all Fortune/SCI end-users in western and northern Florida are cared for properly. This is in keeping with AIS' great care and concern for all of our customers, whose ongoing satisfaction helps strengthen the excellent reputation AIS is noted for."

The following are several Product Announcement we received from Fortune/SCI. These Announcements are naturally promotional in nature, but we include them so that you can be kept up-to-date on new products.

HiLine Successful Retailing

You get what you pay for. And not paying attention to your business' daily transactions could cost you.

HiLine is an integrated point of sale (POS), inventory control, sales analysis and back office accounting system. It makes sure that your retail business gets the attention it needs without you being buried under a pile of paperwork or tied to your computer. HiLine lets you calculate the most profitable inventory levels, pricing strategies, customer services, billing techniques, sales, and advertisements quickly and efficiently with just a few keystrokes. Utilizing its superior POS procedures, i.e. bar coding, magnetic stripping, and on-line positive credit checking, gains you customers and profits through high transaction counts and fast checkout times.

For further information on "HiLine" contact Mark Keelan at:

CompuAmerica, Inc.
3250 Ocean Park Boulevard
Suite 200
Santa Monica, CA 90405
(213) 450-8600

The Contractor

Don't let unexpected costs and miscalculated estimates cut into your profits!!

The Contractor, by Maxwell Systems, Inc. builds sound profits by keeping a lid on fluctuating costs. Specifically designed for the construction industry, The Contractor is a multi-user, multitasking accounting and management system. There are many modules available on the system that increase your power to determine the effects of changes in variable job costs.

Complete reporting capabilities to provide you with current and accurate job status. Government, union and labor reporting, sub-contractor information with retentions and automatic updating to the General Ledger are all included in this comprehensive system.

Built around a powerful data-base management system, The Contractor includes a powerful report writer and query language that office workers can use with ease. For further information on The Contractor, contact:

Julie Reeder or Janice Babin
Maxwell Systems, Inc.
2838 DeKalb Pike
Norristown, PA 19401
800 888-4357

The Success System

How do hotel managers, travel agents, group managers, and recreation/leisure industries remain competitive? They become part of the Success Story!

The success software packages are unique and are designed for a variety of markets. With three products available to date, these packages can provide solutions for hotel properties, lodging and resort management, and the leisure and recreation industries such as Marina Management, Cruise Ship Operations, Yacht Charters, Dude Ranch Property Management, and Campground Reservations.

Lodging Data Systems (LDS) is committed to the hospitality and recreation/leisure markets and are constantly developing new products that will serve a variety of industries within this marketplace. Written in INFORMIX, the software is "user-friendly," flexible and easy to learn. LDS as been providing solutions to the hospitality market for over five years and have installations nationwide.

For more information, contact:

John Haberkorn
Lodging Data Systems, Ltd.
P.O. Box 4434
Englewood, CO 80155
303 796-1302

Fortran 77 and Pascal Programming Languages

Fortune/SCI announces the availability of the newest version of the FORTRAN 77 programming language (version 2.0) and the Pascal programming language (version 2.0) for both the Formula and 32:16. This is the first release of FORTRAN 77 and Pascal on the Formula and includes a compiler target switch feature which provides the capability to cross-compile between the 32:16 and Formula.

FORTRAN is a high-level, problem-oriented programming language, designed particularly for scientific and statistical environments where heavy calculations are necessary within applications. Fortune/SCI's FORTRAN conforms to the FORTRAN 77 standard, and includes preprocessing facilities which allow compilation of programs written in **RATFOR** and **EFL**. Other features include profiling support and array bounds checking.

Pascal is a general purpose programming lanuage designed to encourage well designed, structured programs. It is often the language chosen for teaching environments as it promotes good programming techniques such as program modularity and data structure design. Fortune/SCI's Pascal compiler conforms to the specifications of Jensen and Wirth "Standard Pascal" and International Standards Organization (ISO) Pascal. The compiler also includes the University of California at Berkeley's extensions to standard Pascal.

Performance and Limitations

To improve performance, both FORTRAN 77 and Pascal offer basic optimization features which help the developer create fast and efficient code. Specific to the Formula, support of the Motorola 68881 Floating Point Coprocessor (IEEE 754 floating point) adds another performance feature, as the execution speed of arithmetic functions is greatly increased. All outstanding test discrepancies from previous versions of these products on the 32:16 have been addressed by Fortune/SCI in these new products.

FORTRAN 77 (version 2.0) and Pascal (version 2.0) are compatible with the following products on the 32:16: C compiler (version 1.7 and 2.0), Pascal Compiler (version 1.3), and the FORTRAN Compiler (version 1.3). Please note that Language Development Tools (version 2.0) is required with these products.

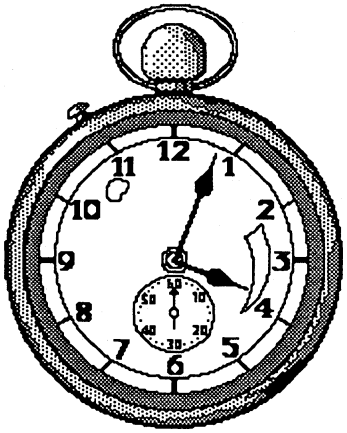
Prerequisites

32:16 system with 512 K of memory or Formula

FOR:PRO 2.0 for the 32:16 or FOR:PRO 3.0 for the Formula

Language Development Tools 2.0 (If an earlier release of Language Development Tools is installed, you must install Release 2.0 before installing FORTRAN 77 or Pascal compiler).

Of Time and Location





In this article, we use one of our newest Software Diskettes as the jumping off point to illustrate a range of UNIX functions. As a consequence, this article will (1) describe what is on the **Zip Tease Diskette**, (2) discuss some simple UNIX commands that can be used with this data, (3) illustrate some programs that work with the Zip Tease data, (4) illustrate how to speed up these programs and (5) provide you with some useful utilities for your business.

What is on the Zip Tease Diskette

There are two sorts of things on the Zip Tease diskette. First, there are data files. The bulk of these files simply cross reference mail zipcodes with telephone area codes and exchanges. There is one file for each area code. In fact, there is so much information in these files that they cannot all fit on one diskette. To solve this problem, we have compressed many of the data files so that they will fit on one diskette.

For example, all the exchanges and zipcodes for area code 201 is contained in the file called **201.dat**. Below is a listing of the first few lines in that file:

```
201 204 07920
201 204 07924
201 218 08807
201 218 08835
201 218 08869
201 218 08876
201 221 07920
```

You will quickly see that the structure of this file is area code, exchange, and zip code. Later in the article, we will show you how to use a simple UNIX command to list out all the zip codes that are associated with an area code and an exchange. In addition, we will write some shellscripts that will tell you which state a particular zip code is in. In fact, we will solve this problem in three different ways in order to show you how fast or slow some UNIX programs can be.

There are several other data files on this diskette as well. A brief description of these files follows:

area-st.dat - This file cross references area codes with state abbreviations. For example, NJ is linked with 201.

overseas.dat - This file cross references countries with their telephone country code. In addition, it gives a relative time from the time at noon in Greenwich (England). Thus, the information in this file could be used to tell you the country code for some country you might wish to call as well as the time in that country. One of shellscripts we talk about in this article accomplishes this task.

st-area.dat - This file cross references States in the U.S. and other associated properties with area code and a relative time from the time at noon in Greenwich.

statezip.dat - This file cross references States in the U.S. with the low and high zipcodes for that State. This file could be useful in ascertaining in which State you would find a zipcode.

The second type of thing that the Zip Tease Diskette has are programs which work with some of these data files. The programs that we will be

discussing in this article were programmed by us and represent some quick but useful ways of using the data. These programs are included for your benefit. In addition, we also include a program called **telltime** which was adapted from an article in **Unix/World**. This program tells you the time in a certain area of the United States based on the area code you call. We find this a useful program because we do a fair amount of long distance colling. When I make a phone call I sometimes want to know, "Is the time in the location that I am calling 1, or 2 or 3 hours behind me?" Telltime is the program that answers that question.

Want to Find a Zip Code?

In our first foray that uses this data, we explain how to find all the zip codes associated with a particular area code and exchange. This could come in handy if you know someone's phone number, but not their zip code.

Recall that we have data files that cross reference the zip code with an area code and an exchange. Also recall that these files are named by their area codes. Thus, the file called **201.dat** contains all the information for the 201 area code. If we want to find all the zip codes associated with the 646 exchange in the 201 area code, we would want to look through the **201.dat** file for all the lines that had a 646 in the exchange spot.

This turns out to be a very simple thing to do with UNIX. We can use the **grep** command to search through the file for the exchange. (See Volume 3 Number 4 for an article on how to use **grep**). The command would look like this:

```
$ grep " 646 " 201.dat
```

Grep simply looks through the **201.dat** file and prints all the lines that have a 646 in them. Actually, we are looking for the pattern SPACE646SPACE (notice the double quotes on the grep command which are each a space out from the 646). We look for SPACE646SPACE because this will exactly pick out the exchanges.

Let's make a shellscript that makes this process simpler. The shellscript is a one-liner as follows:

```
zcat $1.dat | grep " $2 "
```

Now, if we put this one line into a file (we'll call this file **fz** for Find Zip code) and make it executable (`chmod +x fz`), we can then use it in the following manner:

```
fz 201 646
```

Ok. We've thrown a couple of curves but they are easy to understand. Let's first start with **zcat**. To explain **zcat** we need to back up a little bit. The data files that contain the zip codes etc. represent a lot of information. In fact, more information than will fit on one floppy. In order to fit that information on one floppy we have compressed it. (For more on **compress**, see Volume 2 Number 7 and our **Compressor's Delight Diskette**).

Zcat is simply a program that gets at the compressed data. In principle, it is like the **cat** command (which is a command that simply types out the contents of a file onto your screen) but **zcat** is designed to work with compressed files.

What is **$1** and **$2**? If you've read through the shellscript series of articles, you'll recognize these as arguments. Arguments are placeholders of sorts. When we type `fz 201 646`, the 201 gets put in the place of **$1** and 646 gets put in the place of **$2**.

Thus, our **fz** shellscript first uncompresses the file associated with the area code we have requested and this information is sent to the **grep** program which pulls out the lines that match the exchange that we have requested. One final note, **zcat** does *not* leave an uncompressed version of the file on your disk so you do not have to worry about running out of space.

So, now you are all set to use our shell script to find zip codes, right? Well, maybe not. This is because not all of our readers have **grep.** The UNIX program **grep** is normally a part of **The Development Utilities, which can be bought from your local Fortune/SCI dealer..** There is no need to worry, however, because we have included on the Zip Tease Diskette a freely distributable program called **bm** which performs many of the functions of **grep** - and faster than **grep** at that. So, the version of **fz** on the Zip Tease Diskette looks like this:

```
zcat $1.dat | bm " $2 "
```

The Zip is in What State?

In our next program we will figure out what state a zip code is in. We will also discuss issues of speed in shellscripts.

Figure 1. zts Shellscript

```
:
# display the state given a zip code
#
zipcode=$1
echo "Searching for " $zipcode
IFS='
'               # set IFS to just newline
for i in `cat /h1/zip/statezip.new`
do
        IFS=' '         #set IFS to space
        set $i
        lozip=`expr $2 \* 100`
        hizip=`expr $3 \* 100`
        gtelz=`expr $zipcode '>=' $lozip`
        ltehz=`expr $zipcode '<=' $hizip`
        case $gtelz$ltehz in
                11) echo $zipcode "is in " $4
                        exit;;
        esac
done
```

** *See the NEW product DOS LINK* **

SOFTWARE FOR THE FORTUNE 32:16

NEW PRODUCT!!! DOS LINK ($495.00) - provides easy-to-use menus and programs that transfer data between any PC (including Laptops) running DOS and the Fortune system. Here are just a few examples of how this product can be used:

- Transfer Multiplan or Fortune:Word Documents from or to your DOS system.
- Transfer other ASCII or binary files.
- Use the PC or Laptop as a remote terminal on the Fortune system.
- Transfer THOROUGHBRED (SMC) BASIC data that has been extracted using one of our other products

Documentation and software for both the DOS and Fortune system are provided. Data may be transferred via a direct connection (9600 baud) between the two computers or via modem (at modem speed). You may also order additional hardware for your PC that may be required to accomodate the transfer.

If you are interested in purchasing a Laptop computer that can be used with this product for people on the road or at home, give us a call!! We also provide consulting services for interfacing your DOS system to the Fortune ($50/hr).

MULTIPLAN LINK - can pass data from BAS/IDOL/BASIC files to Multiplan and back. For example, it could be used to send data from any of the BAS files (Chart of Accounts, Customer Master, Inventory Master, etc). Allows selection of fields to be sent and provides logical retrieval and key range selection. **$295.00**

RECORDS PROCESSING LINK - can pass data from BAS/IDOL/BASIC files to Records Processing and back. For example, it could be used to take customer address data from the Customer Master and will automatically make the Records Processing List document for mail/merge. Allows selection of fields to be sent and provides logical retrieval and key range selection. $295.00

ASCII LINK - can pass data from BAS/IDOL/BASIC files to ASCII and back. Use to send data between machines or databases. For example, it could be used to send data from any of the BAS files (Customer Master, Inventory Master, etc). Allows selection of fields to be send and provides logical retrieval and key range selection. **$295.00**

TERMINAL/PRINTER LINK - provides capability to print data on a printer connected to a workstation. Can print BAS/IDOL/BASIC reports, Multiplan and Fortune:Word documents. BAS/IDOL reports can also be viewed before printing, converted to Fortune:Word format; accumulated for mass printing; backed up to diskette; printed on any system printer. An option to print BAS/IDOL reports on a LASER PRINTER is $50 extra. Many other options are included. **$195.00** (Fortune:Word printing on a printer connected to a workstation is $50 extra)

TELEPHONE LINK - on line telephone message, telephone directory and inter-office mailbox/menu system. Telephone message entry resembles typical pink slip. Telephone directory can search by name or company. Mailbox/message system can send and receive Fortune:Word documents or Multiplan documents. Does much more! Give us a call. Written in Business Basic. **$295.00**

CALCULATION LINK - provides programs for amortization, depreciation, loan repayment, averaging (with graph), linear correlation and breakeven analysis. Written in Business Basic. **$95.00**

KOMPACT PERSONNEL ACCOUNTING - provides for data capture of personal, wage/salary, job, education, salary/wage, appraisal training, dental, medical and life insurance data. Includes over 80 pre-defined reports/worksheets. Written in Business Basic/IDOL. **$995.00**

PRINTER LINK - provides printcaps for Okidata 84, 2410, TI 855, Anadex 6500 and Printronix. Includes shell script for changing pitch and LPI. Allows these printers to be used in Fortune:Word. **$95.00**

MODEM LINK - allows Hayes 1200/2400 modem to be used on one port for both incoming and outgoing calls. **$95.00**

WESTERN LINK - provides instruction on how to use your Fortune for sending/receiving telex messages using Western Union's Easylink. Can replace your telex! Fortune ITE or Handshake software required. **$95.00**

UNI-KOMP 9803 Rocktree Houston, Texas 77040 (713) 937-7782

Figure 2. nzts Shellscript

```
:
# display the state given a zip code
#
zipcode=$1
echo "Searching for " $zipcode
IFS='
'               # set IFS to just newline
for i in `cat /h1/zip/statezip.new`
do
        IFS=' '        #set IFS to space
        set $i
        lozip=`expr $2 \* 100`
        if test $zipcode -ge $lozip #-a $zipcode -le $hizip
        then
                echo $zipcode "is in " $4
                exit
        fi
done
```

Figure 3. czts.c C Program

```
#include <stdio.h>
main(argc, argv)
        int argc;
        char *argv[];
{
        int zipcode;
        char zipstring[20];
        char line[100];
        FILE *fp, *fopen();
        char st_abbr[20], state_name[40];
        int lozip, hizip;

        if ( (fp = fopen("statezip.new","r")) == NULL)
        {
                printf("Can't open statezip.new file\n");
                exit(1);
        }

        strcpy(zipstring,argv[1]);
        sscanf(zipstring,"%d",&zipcode);
        printf("Input zipcode is %s\n", zipstring);

        while( fgets(line, 100, fp) != NULL )
        {
           sscanf(line,"%s %d %d %s", st_abbr,&lozip,&hizip,state_name);
           lozip = lozip * 100;
           if (zipcode >= lozip)
           {
                printf("%s is in %s\n", zipstring, state_name);
                exit(0);
           }
        }
}
```

One of the data files on the Zip Tease Diskette looks like the following:

```
ST Low Hi   State
   Zip Zip

AK 995 999 Alaska
AL 350 369 Alabama
AZ 850 865 Arizona
AR 716 729 Arkansas
CA 900 966 California
CO 800 816 Colorado
```

This file gives the low and high zip code for each of the United States. Thus, given a zip code, we should be able to determine which State that zip is in by seeing if the zip code is greater than or equal to the low zip code and less than or equal to the high zip code. Notice, however, that the low and high zips are 3 digits while a zip code is normally five digits. (This will all change with 9 digit zips). For example, 99611 is a zip code in Alaska. It does not fall between 995 and 999 but it does fall between 99500 and 99900. Thus, we need to multiply these high and low zips by 100.

The shellscript, which we call **zts** for Zip To State, is shown in Figure 1. The basic idea in this script is to read each line in the **statezip.dat** file. We then multiply the low and high zips by 100. This is accomplished by the following lines:

```
lozip=`expr $2 \* 100`
hizip=`expr $3 \* 100`
```

Next, we use the expr command again to find out if the input zip code is between the lozip and the hizip. The line:

```
gtelz=`expr $zipcode `>=' $lozip`
```

will make gtelz equal 1, if the zip code is greater than or equal to the lozip, and will make gtelz equal 0 if not. Likewise, the line:

```
ltehz=`expr $zipcode `<=' $hizip`
```

will make ltehz equal 1, if the zip code is less than or equal to the hizip, and will make ltehz equal 0 if not.

Thus, when both gtelz and ltehz equal 1 on a single line, then we have found the State in which the zip code is. The case statement performs this decision function.

Making zts Faster

This particular way of finding a zip code happens to be very slow. The main reason for this slowness comes from the fact that we must perform 4 **expr** commands for each line we read in. We can't get into the details of why this is so, but each time we run an **expr**, we incur costs in terms of the load on the system.

So, we wanted to make **zts** faster. We realized that we wouldn't have to see if the zip code was both larger than the low zip code AND smaller than the high zip code if the list was sorted by low zip code. We sorted the **statezip.dat** file by low zip code and called this new file **statezip.new**.

Our new **zts** program, now called **nzts**, is shown in Figure 2. You will notice that whereas we had three **expr** commands in zts, we have only one which multiplies the low zip code by 100. Then, we use the **test** command to see if our zip code is greater than or equal to the low zip.

Let's time our two programs to see what kind of increase in speed we have been able to obtain. The **time** command in UNIX will perform the timing for us and the output from it is shown below:

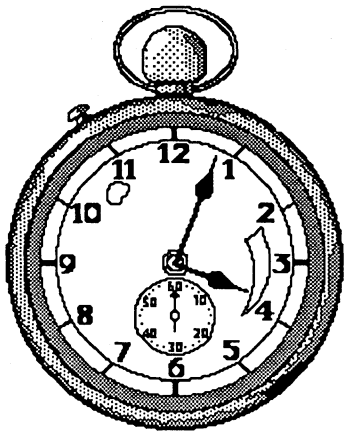
```
$ time zts 02174
Searching for  02174
02174 is in  Massachusetts

real      1:21.0
user        33.4
sys         41.4

$ time nzts 02174
Searching for  02174
02174 is in  Massachusetts

real        36.0
user        15.6
sys         18.0
```

Thus, we see that **nzts** is more than two times faster than **zts**. **Nzts** took 36 seconds to complete while **zts** took 81 seconds (1 minute and 21 seconds). So, we were successful in speeding up



this process.

However, 36 seconds is still a long time for a program of this sort. Is there anything else we can do? Yes. We could write a C program to perform this function. We did just that and the C program is shown in Figure 3.

We don't propose to explain C in this article but we did want you to see how much faster a C program is when compared to a shell script. The following is the time it takes for our new program, which we call **czts**, to find our test zip code:

```
$ time czts 02174
Searching for  02174
02174 is in  Massachusetts

real          1.0
user           .3
sys            .1
```

This program is *36* times faster than our best shell script.

What Else?

We hope we have both whetted your appetite for our new **Zip Tease Diskette** as well as emboldened you to try your hand at writing some shell scripts.

UNISOFT INC.

Experts in Computer Consulting - Sales - Service

Presents

FORTUNE SYSTEMS HELP

We Provide:

- Expert Trouble-shooting
- Years of programming and accounting experience
- *Free Initial Consultation*
- Training Available for:
  - * BASIC/IDOL/BAS and Multiplan
  - * UNIX and Ext. Fortune:Word
- Hardware Repairs & Upgrades
- FORTUNE trained experts at a cost that will surprise you!
- *Trade-Ins Welcome*
- Specializing in:
  - * Hard Disk Trade-In
  - * Hard Disk Upgrades

Authorized Fortune Systems Repair Depot
Fortune Systems National Value Added Reseller (VAR)
CALL: (303) 671-8600 EMERGENCY HELP: (303) 649-8505

UNISOFT INC. 14100 East Jewell Avenue Suite 27 Aurora, Colorado 80012

We have only been able to touch on some of the programs on the Zip Tease Diskette and on some of the uses of the data contained on it. In addition to the **fz** and **zts** scripts there is another called **otim**. **Otim** gives the Overseas Time and Telephone Country Code for an input Country. For example, if I type:

```
otim Australia
```

Otim will print out:

```
The Country Code for Austra-
lia is 61
It is now Sep 24 and 7:42 am.
```

This program is a nice complement to the **telltime** program which gives the current time in parts of the United States based on an area code. Thus, if I type:

```
telltime 415
```

I will see:

```
Time is 13:44 in San Francisco
California (415).
```

These programs are very useful for those of us who make long distance calls to various locations and don't really know what time it is in that particular locale.



Conclusion

In this article, we have illustrated some of the shellscripts included on the Zip Tease Diskette. These included zts, nzts, czts, telltime, fz and otim. These shellscripts are relatively simple in nature but do provide some useful tools for working with the data included on the Zip Tease Diskette.

We have also touched upon the idea of refining shell scripts to make them faster. We accomplished this first by removing some commands from the script that required heavy use of system resources. The main method of removing these commands was that of rearranging the data file so that fewer operations had to be performed. The second way we speeded up the shellscript was to actually write a C program. The C program was also relatively simple but it gave us a routine that was 36 times faster than our best shellscript.

Hopefully, we have given you some ideas for your own scripts. Send them into us, if you think they will be useful to other readers.□

Mark Palmerino

MULTIREG

Multiple Regression Program

A Professional Econometric Estimation Program For The Fortune Systems 32:16 Computer.

MultiReg is a professional tool, designed to provide detailed output of Multiple Regression Analysis, on the Fortune 32:16.

MultiReg generates an exhaustive array of statistical results.

*Regression Coefficients
*Correlation Coefficients
*Standard Errors and t-values
*Actual and Predicted Values of Dependent Variable
*Values of Residuals
*Means of Variables
*Analysis of Variance Table
*F Ratio
*Variance-Covariance Matrix
*Graph of Residuals
*Autocorrelation Coefficients
*Zero-Order Correlation Matrix
*Standard Error of Estimate

Order Your Copy Today

Econometric Forecasting, Inc.

6971 N. Federal Hwy, Suite 102
Boca Raton, Florida 33487-1648
Phone: (305) 994-6373

Dealer Inquiries Welcome.

Retail Price: $495.00

MultiReg contains a Variable Transformation Menu, which allows for 12 different types of variable transformations to facilitate the estimation of non-linear reltionships.

MultiReg has the power to run literally dozens of regressions with one stroke of the Return Key. One example in the Reference Manual is the estimation of an econometric model. You simply type one command line, calling MultiReg and directing the output to the Printer. After you press the Return Key, you might as well go out to lunch, because it will take a 20MB Fortune 32:16 ten full minutes to crank the numbers, and after that it will take a Qume 945 Printer 55 minutes to print the resultes. The printed output will be 120 pages, containing detailed results on 33 stochastic equations with 128 estimated regression coefficients.

The MultiReg Package consists of two Disks, in a durable plastic library case. Disk 1 of 2 is the Training Version of MultiReg, which operates in "Interactive Mode". Disk 2 of 2 is the Standard Version of MultiReg, which operates in "Batch Mode". A clear and concise Training Manual and a comprehensive Reference Manual are provided.

File Management

Continued from page 1

until they are specifically directed there by the computer at some convenient time. This process is somewhat similar to the status of your documents when you are using Fortune:Word — until you hit CANCEL/DEL and EXECUTE, your documents are in memory and are therefore in danger of being lost.

Memory caching is the most frequent cause of file system corruption. In any event, no matter what the cause, immediate action on the part of the administrator is required. I have all too often seen the catastrophic results of canceling out of the file system check provided automatically by the Fortune Operating system.

File System Check (fsck)

Prior to Version 7 UNIX (eg. before **FOR:PRO**) there was no real **fsck** program. The programs **icheck**, **dcheck**, **ncheck**, **fcheck**, and **fsdb** were used to find, isolate, and hopefully, fix corrupted file systems. Many hours were wasted by gray and white wizards painfully trying to recover from a system failure. The current **fsck** is an improved version of these programs which is a little smarter about how to correct problems while they are still minor.

The **lost+found** directory is an empty directory specially constructed to hold lost files that fsck finds. DO NOT MAKE THIS DIRECTORY BY HAND

There are a couple of preliminaries that should *always* be followed on hard disk file systems. First, **fsck**, as part of its automatic recovery ability, should have a **lost+found** directory. The **lost+found** directory is an empty directory specially constructed to hold lost files that fsck finds. DO NOT MAKE THIS DIRECTORY BY HAND. There is a shell script usually used to construct this directory with empty slots pre-allocated in it. To create this structure, change your directory to the top most (root) directory of the structure and execute "`sh /etc/mklost+found`".

Second, the **lost+found** must be located in the top most directory of the file system (eg. /) and it must have pre-allocated empty directory slots. All too often, people have used **mkdir** to construct a deceptively useless **lost+found** that cannot be used by **fsck**. If in doubt, remove the old one and create a new one as described above.

Proper Use of fsck

Improper use of **fsck** is the most common cause of continued file system corruption. There are a few simple rules about using **fsck** that if followed will reap excellent returns.

a) *Never use **fsck** on an active file system.* If the file system is being changed, **fsck** is sure to fail and further corrupt the file system. (More on this later).

b) Except for the root file system, always run **fsck** on an unmounted file system. For a badly corrupted root directory use the stand-alone maintenance (Disk 1 of the Single User FOR:PRO set) to boot the machine.

c) If you don't know how to use **fsdb** (or don't have **fsdb**) you will almost always have to answer yes (`fl`) to **fsck**'s questions. Answering no will almost always require some manual action on the administrator's part to finish correcting the file system.

d) When possible, run **fsck** on the root file system in single user or maintenance mode (More on this later).

e) NEVER cancel out of a **fsck** without manually completing a **fsck** before using the file system.

f) Unless the file system is unmounted, if you get the message:

"******* File System Was Modified ******"
or
"******* BOOT UNIX (NO SYNC!) ******"

IMMEDIATELY REBOOT THE MACHINE WITHOUT TYPING ANYTHING!

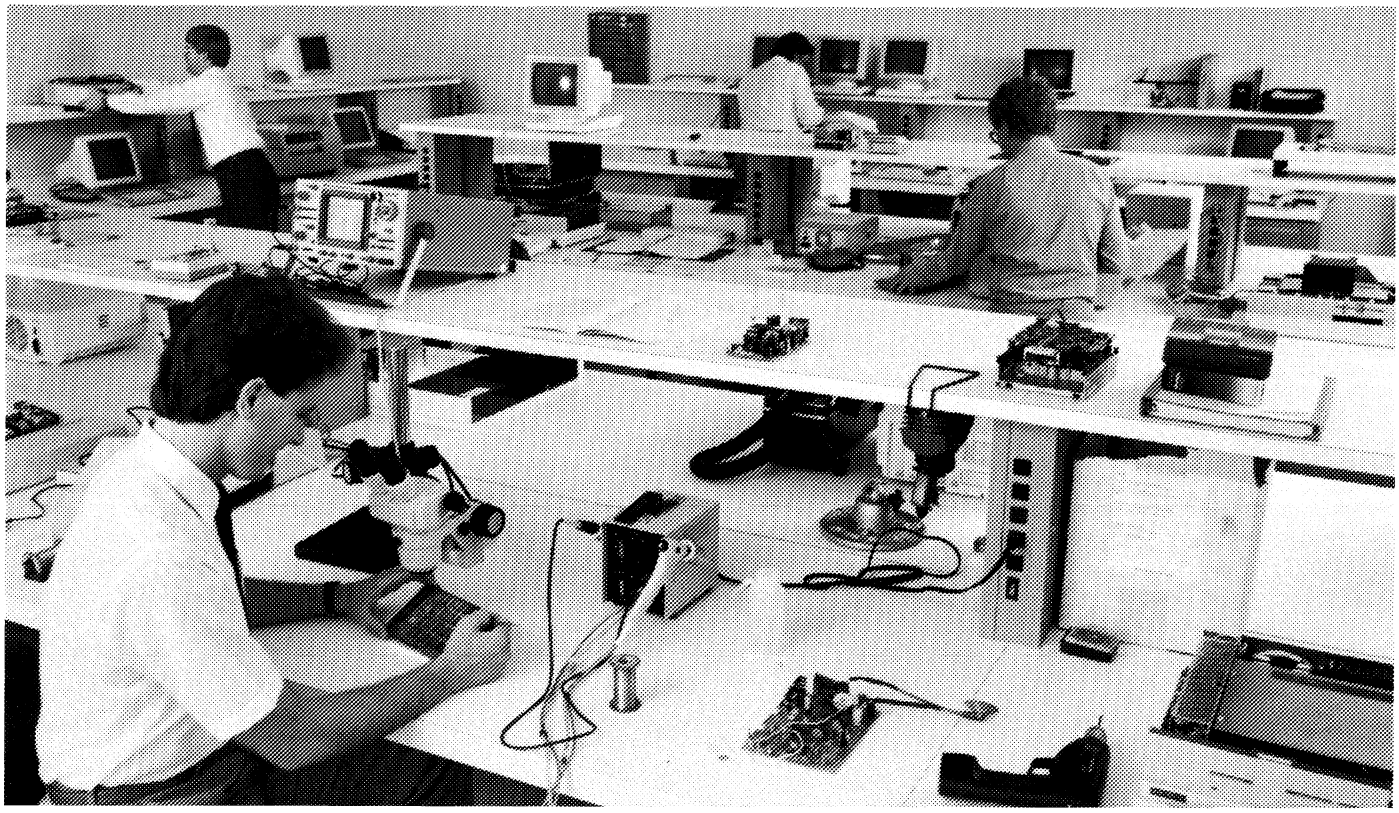
There are reasons for all of these rules. Following them strictly will give you the best results, unless you know exactly what you're doing.

Rules a, b, d, and f are all related to how UNIX caches disk blocks in memory. Obviously, using **fsck** on a file system that is in use and changing will cause **fsck** to perceive errors where they do not exist since **fsck** expects to see a "stable" picture of the file system.

There are reasons for all of these rules. Following them strictly will give you the best results, unless you know exactly what you're doing.

By unmounting a file system, you guarantee that the file system is not in use. Further, UNMOUNTED FILE SYSTEMS are never cached by the operating system. That is, UNIX does not keep in memory any information about an unmounted file system. If **fsck** corrects a problem with the file system and it is unmounted, the corrections are made on the disk and all is well.

Mounted file systems, such as the root file system (/), cause problems. As discussed in the first article, UNIX keeps a copy of the superblock (and others as well) in memory for fast access (cache). While **fsck** is busily out there fixing up the pointers of inodes, directories, free lists, etc. on the disk, a copy of the bad and possibly corrupted information sits inside the UNIX memory buffers just waiting to be rewritten back to the disk UNDOING or, even worse, FURTHER CORRUPTING all the work that **fsck** has just done.



The Best Just Got Better

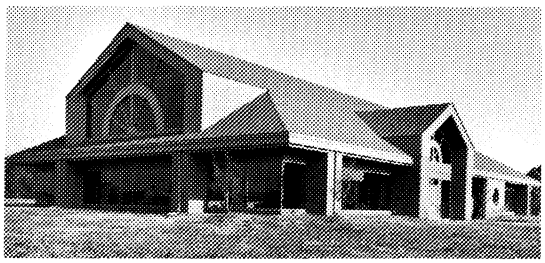
At Micro Resources, our commitment to Fortune/SCI is clear. We've just opened a brand new facility to make us even more competitive and service-oriented for Fortune Systems users.

Expanded Service
Not only have we added service personnel, our newly-equipped service department is simply the most advanced in the mid-west. That means fast turn-around for system upgrades or repairs.

Expanded Inventory
Service can only be as fast as the needed components are available. That's why we've enlarged our on-hand inventory. Volume purchases also keep our prices competitive.

Expanded Upgrades
All this growth has led to a landslide in upgrade business. That volume has provided us with a wealth of components and systems available in limited quantities at fantastic values.

Expanded Classrooms
Now Micro Resources has the space and integrated training facilities to host dozens of employees. Our training programs are tailored to your needs and your schedule.



We believe in Fortune/SCI. And we believe in supporting Fortune/SCI users with the finest service and product at a competitive price. So, if you're unable to find a resource to meet your needs, call Micro Resources. Our dedication to Fortune Systems and to complete service is simply unsurpassed.

Call us at 1-614-766-2335. In Ohio, call toll free 1-800-622-UNIX.

Authorized Fortune Master Dealer

Micro Resources

255 Bradenton Avenue, Dublin, Ohio 43017

BEYOND TECHNOLOGY . . . *RESULTS*

The messages in step (f) above are a warning about mounted file systems that have been changed. It says, "Hey I've just fixed your disk of this corrupted stuff. Don't let the UNIX kernel write that bad stuff it has back out onto the disk. REBOOT!"

Finally, if you can, boot from a floppy for badly corrupted root file systems. Booting from a floppy is easy if you have FOR:PRO 1.8 or a later release. Just insert the Single User FOR:PRO disk 1 floppy in the drive and reset the machine. When the reconfiguration menu appears, press `f5` and wait. At the maintenance screen type "`sh`" and `RETURN`. At this point your root file system is the floppy disk and your hard disk is unmounted. (The reverse of the usual situation.) Once you are in the shell perform your **fsck** on hd02 (or whatever). If you haven't followed this, then you shouldn't be doing it! Get some help from you dealer, or from someone familiar with UNIX.

The Six Phases of fsck

Thus far, I have only attempted to help users properly use **fsck**. Understanding what **fsck** is doing will also assist in this task. It is almost mandatory that you understand the structure and components of the file system discussed in my last article. It will help if you review it before continuing.

Fsck works on trees. The best way to save a tree with a sick or dead branch is to prune it. For this reason, correcting corrupted file systems almost always results in lost data. This is the reason why smart people make reasonably frequent backups.

Fsck does its work by scanning and rescanning certain components of the file system. Each phase compares components and checks that the logical representation from each of these components correctly match.

Phase 1 checks the blocks and sizes. Obviously, to get blocks and sizes, the system must read the inode list to determine the sizes and locate the blocks used by each file. Inodes are checked for validity which includes inode type, zero link counts, inode sizes, and the infamous duplicate block. Any physical disk block can only be allocated to a file once so duplicates cause interesting questions.

Phase 1 usually provides you with the ability to clear an inode. This zeros out the bad information in the node. Any file (or, worse yet, directory) that might have been pointed to is gone. Duplicate blocks mean that two files point to the same block on the disk. **Fsck** rescans to find the original inode along with the duplicate for correction in Phase 2.

Phase 2 checks the path names. Remember that directories are also files with allocated inodes under the UNIX file system. Files in error and removed in Phase 1 must have their directory entries cleaned up. Phase 2 cleans up error conditions caused by improper inode mode status, out of range inode pointers (inodes that point to Kuala Lumpur), and directories that point to bad inodes as described above.

Phase 3 checks for connectivity. Phase 2 concerned itself with removing directories that don't point to valid files. Phase 3 concerns itself with reconnecting files that aren't pointed to by the tree structure. If you are "lucky" and a chunk of a large branch got pruned, all the smaller branches will be grafted back onto the tree in, yes you guessed it, **lost+found**. Unfortunately, the name portion of the lost file will be separated from the file itself so a pseudo name (number) is assigned to the file as it is placed in **lost+found**. Matching a files contents to where and what it was is sometimes a lost cause.

Phase 4 checks the reference counts. Translated, this phase scans the link count of each entry found (and not removed) in Phases 2 and 3. Sometimes, a file that was not pointed to under the directory structure but still has an inode can be relinked back to the file system under **lost+found** as in Phase 3.

Phase 5 checks the free list. This phase scans the free-block list maintained by the file system and resolves the missing or unallocated blocks added or removed in the prior phases. When an inconsistency is detected, **fsck** provides the option to rebuild it.

Phase 6 salvages the free list. If specified in Phase 5, the system reconstructs a free block list from the newly constructed file system in the last 5 phases.

Finally, if the file system was modified and mounted, then the messages in (f) above are issued and you should reset the machine and reboot.

Review

As you can see, the file system is a rather fragile entity. While **fsck** does a pretty good job of cleaning up, or rather pruning, there are some pathological conditions which produce circular or perverse environments that **fsck** just cannot correct or handle. Fortunately these happen very infrequently, and can be corrected using **fsdb**.

Understanding the structure of the file system is most of the learning battle. Proper procedures and use of **fsck**, along with regular backups, will almost always provide a good, healthy solution to what would otherwise be a catastrophic experience.

There is a more detailed discussion of each of the error messages and prompts provided in the **Fortune Hardware Reference Guide** (1/87 jrg). Copies of this document may be obtained from Fortune or your local dealer.

My next article will address the routines **icheck**, **ncheck**, and **fsdb** and bad block sparing as administrative devices for repairing damaged file systems.

If you actually want to use **fsck** to clear up problems on your disk, read on.

Using fsck

The most common questions I receive about **fsck** are "how should I use it?" and "to what should I answer yes or no?". The simplest way to use **fsck** is let the system take the first crack at it. Most UNIX systems, including Fortune, provide **fsck** with a number of switch options with the invocation. Here is what they are:

AUTOMATIC MODE:

```
fsck -p [filesystem  ...]
```

MANUAL MODE:

```
fsck [-n] [-sX] [-SX]
[-tfilename] [-y] [filesystem ...
```

where:

1. Anything above in brackets is optional.
2. In the first form, automatic mode using `-p`, the system will attempt to fix minor system inconsistencies for: unreferenced inodes; incorrect link counts; missing blocks in free list; blocks in free list that are in files; incorrect counts in superblock. All other inconsistencies create abnormal return status and exit. No other switches may be used with the `-p` option.
3. In the manual mode:

   `-n` means assume a no response to all questions and do not modify the file system.

   `-sX` means always reconstruct the free list and rewrite the superblock of the file system. If it is known in advance that the file system is incorrect this option speeds up the fsck process by eliminating one of the passes. `X` is the blocks per cylinder used when the system was created and should (on Fortunes) be left off unless some special timing actions are desired and you know what you are doing.

   `-SX` This option like `-s` above reconstructs the free list but only if other discrepancies were discovered. For a known, inconsistent file system I always use this option (`-S`) since it speeds up the file system check.

   `-t` When performing a file check on very large file systems (larger than 70 megabytes) you will need to use this option to get **fsck** to prompt for a scratch area to use in place of memory. THE AREA SPECIFIED MUST NOT BE IN THE FILE SYSTEM BEING CHECKED OR POSSIBLY MORE DAMAGE WILL OCCUR. Usually I use /dev/hd01 (the swap area) while in single user mode.

   `-y` Idiot mode. This switch causes **fsck** to assume a yes response to all questions. You either have to totally trust **fsck** if you use this, or you may lose a great deal of your system.

On Fortune systems, after a crash, the start up script (**/etc/rc**) will run a **fsck -p** on all the file systems listed in **/etc/**

We are the *largest* Authorized Fortune Master Dealer in the Pennsylvania and Delaware area. As part of the **P**rofessional **S**olutions **G**roup of companies with over 35 years in business automation, we have the *stability* that our customers demand

Our services include a factory trained field engineering team, a complete parts inventory including system upgrade capability and a software support department.

For all of your Fortune needs contact:

Mark Shields of PSG at
2550 Eisenhower Avenue
PO Box 762
Valley Forge, PA 19482

or you can call Mark at (215) 666-1400

fstab. For those of you with multiple drives or multiple partitions, they should be listed in this file. If this fails, then you get the annoying message "FILE SYSTEM CHECK FAILED. CONTACT YOUR DEALER".

So here you are, you haven't read the first part of this article and are now looking for the short way out. The easiest way is to bring the system up into maintenance mode. Reboot and, after the system counts up to 9 and prints out the memory and system configuration, immediately hit CANCEL/DEL. You have about 3 seconds before the system will proceed to ask you for the time and date. If you failed, reset and try again.

In maintenance mode, you either will be prompted for the root password or will be logged in as root if no password is needed. Now is when the fun begins.

First, if you get any of these errors, unless you have good technical assistance, give up. They usually indicate a hardware problem with the disk or controller.

CAN NOT SEEK BLOCK X (CONTINUE?)
CAN NOT READ BLOCK X (CONTINUE?)
CAN NOT WRITE BLOCK X (CONTINUE?)

Here are the most likely errors and what to do:

PHASE 1:

UNKNOWN FILE TYPE I=nnn (clear)

Answer yes. This will zero the inode and its contents will be lost. In phase 2, you will get the UNALLOCATED error message for any directory pointing to this inode.

BBB BAD I=nnnn (clear)

Answer yes. Here an inode pointed to a block outside the legal range of blocks allocated to the partition being examined. Almost always, this will be followed by the message EXCESSIVE BAD BLKS. If this happens, you can continue but it is likely that much of the file system will be lost and the damage is catastrophic. If you answer yes, you will also get the BAD/DUP message in Phase 2 and 4.

BBB DUP I=nnnn (clear)

Answer yes. Like the BAD blocks, above this indicates that the inode pointed to a block already allocated to an inode. This will always cause phase 1b (rescan) to occur. Almost always, you will also get the message EXCESSIVE DUP BLKS. Again, if you continue, much damage has occurred and the repair may leave you with much of the file system lost.

EXCESSIVE XXX BLKS (continue)

XXX is either DUP or BAD as discussed above. If you get these and continue, YOU MUST RERUN FSCK AGAIN TO FINISH REPAIRING THE SYSTEM after this **fsck** is completed.

DUP TABLE OVERFLOW

When this occurs, it's almost always a give up situation. Without **fsdb** and some careful work, the file system is really trash. It is unlikely that continuing will provide a reasonable solution.

PARTIALLY ALLOCATED INODE I=NNNN (CLEAR)

Answer yes. This clears the inode which is in a limbo state. Most often, this is a file just in the process of being created during a crash and will not cause a problem.

PHASE 2

ROOT INODE UNALLOCATED. TERMINATING. or
ROOT INODE NOT DIRECTORY (FIX)

These almost never happen. When it does it usually means the configuration block on the drive has been damaged or is misaligned for the file system. Check the partition sizes in the conf block by using **rdconf** on the drive involved.

DUPS/BAD IN ROOT INODE (CONTINUE)

In Phase 1, some duplicate or bad inodes were found in the root inode. Fixing these will almost always result in a major loss of data and many other errors. This kind of error usually justifies using **fsdb** or a cold boot.

I OUT OF RANGE I=NNNN NAME=FILENAME

Directory FILENAME was found with inode NNNN which is not within the legal inode range of the partition. Remove the directory by answering yes.

UNALLOCATED I=NN OWNER=OOO MODE=M SIZE=SSS MTIME=TT NAME=F

This is unusual and indicates the directory F has not been allocated properly. It is usually caused by a system crash while creating or just having created a directory. Answer Yes to clear and remove the directory.

DUP/BAD I=NN OWNER=O MODE=M SIZE=SS MTIME=TT DIR=F (REMOVE)
DUP/BAD I=NN OWNER=O MODE=M SIZE=SS MTIME=TT FILE=F (REMOVE)

These indicate that in Phase 1 bad or duplicate blocks were found. As in Phase 1, answering yes will clear the inode and remove the directory/file. By answering no, phase 4 may be able to reconnect the file/dir into the **lost+found** directory if, **lost+found** has been properly made and has sufficient empty slots. *NOTE THAT IF YOU ANSWER NO, YOU MUST RERUN FSCK AGAIN AFTER COMPLETING THIS FSCK TO BE SURE THE FILE/DIR HAS BEEN PROPERLY RECONNECTED.*

PHASE 3

UNREF DIR I=NN OWNER=OO MODE=M SIZE=SS MTIME=T (RECONNECT)

Answer yes and the directory will end up in lost+found, if lost+found has been properly created and has sufficient empty slots.

PHASE 4

UNREF FILE/DIR I=NN OWNER=OO

MODE=M SIZE=SS ...(RECONNECT)

Again, like in phase 3, answer yes to try to put the file in lost+found.

LINK COUNT FILE/DIR I=I OWNER=O MODE=M SIZE=SS ... (ADJUST)

Always answer yes. This will correct the link count.

BAD/DUP FILE I=NN OWNER=OO MODE=M ... (CLEAR)

Well, if you get this in Phase 4, it's too late to save the file or directory. This will occur if in Phase 1 you received error messages and in Phase 3 and 4 you failed to answer yes or the system was unable to save the file/directory in **lost+found**. Too bad, answer yes to clear the inode. Expect data loss.

FREE INODE COUNT WRONG IN SUPERBLOCK (FIX)

Always answer yes. This fixes the superblock with the actual inode count.

PHASE 5

EXCESSIVE BAD/DUP BLOCKS ... (CONTINUE)

Try to continue. Usually this is a warning that lots of data loss occurred during the **fsck**. ***You must rerun fsck again if you get these messages after fsck is completed.***

FREE LIST/FREE COUNT/BAD
FREE LIST/SALVAGE FREE LIST

There are about eight of these types of messages concerning the free list being in error. Basically, any repairs made to the file system will almost always result in an incorrect free list on the system. Always allow the system in PHASE 6 to rebuild the free list, if asked, by answering yes.

As discussed above, many of the recovery actions taken by **fsck** will require running **fsck** again. As mentioned earlier if you get the messages:

```
****BOOT UNIX (NO SYNC!) ****

****FILE SYSTEM WAS MODIFIED ****
```

on the ROOT file system (usually hd02 on Fortunes) or on a mounted file system, you must reset your machine WITHOUT A SHUTDOWN immediately after **fsck** is done with a rerun of **fsck**. FAILURE TO DO SO WILL RISK FURTHER DAMAGE.

To be safe it is always good to rerun **fsck** whenever you have had errors corrected to be sure that the corrections have been successfully completed.

Richard Lee Suslick, Cedar Cliff Systems Corporation, 6036 Old Telegraph Road, Alexandria, VA 22310, (703) 960-2999

*(Editor's Note: The need to reboot is based on the fact the **fsck** corrects the filesystem on the disk, but not the image of it contained in the computer memory. If you don't reboot, the bad version from memory may get rewritten to the disk. However, in most cases, Fortune's **fsck** will avoid this problem by writing the superblock from the disk back into memory. You can verify that this has happened by doing a **sync** and then rerunning **fsck.** The second time it should come up clean. If it doesn't, follow the reboot instructions above.)*



Tri-Tech Computer Services, Inc.

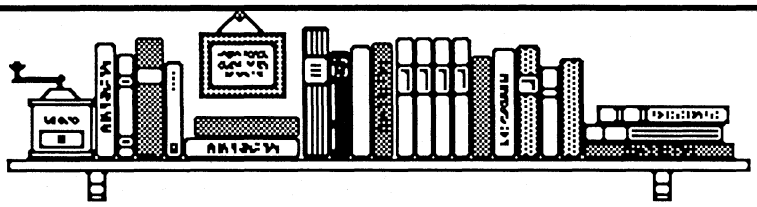
Let's Play Ball.
Our Team Provides The Winning Solutions.

New & Used System Sales * Office Supplies *** Word Processing *** Upgrades**
Laser Printers * Spread Sheets *** Consulting + Support *** Communications *** Accounting**

Software Specials:
Fortune:Word 3.0 Upgrade
Handshake I.T.E. Handshake Blazer Communications
Call Now!!!

Connecticut: (203) 828-9784 **1-800-622-TTCS** **Massachusetts: (617) 965-4453**

The BASIC Advisor



The BASIC Advisor is brought to us from Ray Wannall. Ray is President of BaSiC Software Corporation which is located in Baltimore, Maryland. This month Ray discusses an upcoming Concept Omega conference, BAS sales and support, and finally Ray helps a reader from the U.K.

Question: *I am thinking about adding business accounting software to my 32:16 and have heard that I can purchase either the Fortune BAS applications or the Thoroughbred versions. What do you recommend? Also, do I need IDOL?*

Answer: Is this urgent, or can you wait a few months? Right now the new, fourth generation applications are in development, so anything you purchase today in the BAS or Thoroughbred line will be obsolete within the year. In case it is urgent, I will give you my opinion. (Keep in mind that opinions are like noses: just about everybody has one.)

If you wish to purchase any of the Thoroughbred accounting packages you will find it necessary to locate a dealer of those products who will support Fortune Systems. This type of a dealer is rare, because Fortune has its own dealer network and markets similar packages under the name of BAS (Business Accounting Software). Fortune BAS products are the offspring of the original Data Technology Industries BAS packages, which were also the ancestors of the Thoroughbred line. Fortune BAS and Thoroughbred, therefore, are cousins, and they do bear certain family resemblances. But they are quite different because of the way each has been marketed over the years. Fortune Systems tried to create inexpensive, user-friendly modules which could be installed and operated with little or no additional support needed. (It almost worked). Concept Omega, however, stuck to the original plan of providing a network of dealers with basic packages which could be modified and "verticalized" as needed for each end user. End users of Thoroughbred products are, in general, much larger (and richer) than users of Fortune BAS packages. Thoroughbred users don't seem to mind paying tens of thousands of dollars for four or five packages. Fortune users choke at the thought of buying a package whose price tag exceeds three digits.

Although the standard Thoroughbred packages now on the market are relatively inexpensive, you cannot expect to purchase one and use it without extensive and costly dealer support. Also, Thoroughbred modules are not always independent: in order to use Purchase Orders, for example, you must also buy Order Processing. Concept Omega assures me that once the fourth generation Business BASIC is completed the business applications will be overhauled and made more user friendly. This may happen within the next year, but I am not holding my breath. In the meantime, I have to recommend you stick with Fortune BAS.

I have said it before, and I will say it again: IDOL is a tool for programmers. If your company has a programming guru on staff, you may wish to purchase IDOL, or even IDOL II, for creating custom reports, files and menus. But please don't expect IDOL to work miracles for a non-technical staff.

Question: *I am certainly glad to see your articles are starting to cover IDOL programming. Many years ago, when we bought our Fortune 32:16, we decided to purchase IDOL in lieu of any business applications. All of our software has since been custom written under IDOL. Suddenly, I have run into a problem.*

I have a large file of data which has been accumulating over the years, and I want to generate a report which shows only those records which were entered in a given month. Try as I may, I cannot seem to get the logical retrieval to pick up a single month by using the "less than" (<) and "greater than" (>) functions. The field in question is contained in E$(16,8) and is in the format MM/DD/YY. How do I do this?

Answer: There are several ways to accomplish the task, but in the interest of showing off some of the features of the IDOL report generator, I am going to take a roundabout approach.

First, let's construct an IDOL file layout which complies with your E$(16,8) format:

VAR NAME	SEQ	ELEMENT NAME	EL LG	P I	P R	TYPE	POS IN RECORD	POS IN E$ (IDOL)	DATE MASK
AN$	1	CUST NUMBER	6	C	-	A/N	1-6	E$(1,6)	
BN$	2	CUST NAME	9	-	-	A/N	8-9	E$(7,9)	
CN$	3	DATE ENTERED	8	-	-	A/N	19-8	E$(16,8)	
AN	4	PURCH THIS YR	11	-	2	N	27-11	E$(25,11)	
BN	5	PURCH LAST YR	11	-	2	N	39-11	E$(36,11)	

When you initially designed the file, you did not use any of the Date Indicator options offered by IDOL. (If you had, the length of element 3 would have been either six or five). There are four date formatting options provided:

A = MMDDYY
B = YYMMDD
C = Julian date, YYXXX

D = DDMMYY (European)

I try to use the B (YYMMDD) format rather than A (MMDDYY) because program sorts can be conducted without reformatting the date. If you look at the dates April 20, 1986, and August 15, 1987, you will see what I am talking about. When displayed in the MMDDYY format, 081587 lists chronologically before 042086. But in the YYMMDD format, 860420 comes before 870815, as well it should. (I do not use the Julian format because I despise having to write the extra code required to translate it for human beings. And I am not fooled by the "European" tag on DDMMYY: that will always be military to me).

Since you did not use the YYMMDD format in the definition of the element, we can create a new field which will reformat the date in field three. When the report generator asks you DEFINE NEW FIELDS (Y/N), answer in the affirmative and give the new field a length of 6 ('CR' for precision). The new field definition should be:

```
E$(22,2)+E$(16,2)+E$(19,2)
```

Field number six will now be your date in the YYMMDD format. This works because E$(16,8) is your date as MM/DD/YY, where MM is in E$(16,2), DD is E$(19,2) and YY is E$(22,2). Now you can define logical retrieval for, let's say, a July, 1987\ report by telling the computer:

```
#6>"870630" AND #6<"870801"
```

The DEFINE NEW FIELDS feature can be used in a variety of other ways. One of the best is to use it to show the results of arithmetic operations involving other fields in the file. In our example, field 4 is a number representing the customer's total amount of purchases this year while field 5 shows the total amount of purchases last year. Suppose we want to run a report which displays the variance from last year to this year? We can define a new field with a length of 11, precision 2, with the following definition:

```
#4-#5
```

and call the field "VARIANCE". After the record is read, the dollar value contained in field five is subtracted from the dollar value in field four and the result is placed in field six. Field six may be printed on the IDOL report and even totaled and subtotaled if desired. Neat, huh? You can similarly create an inventory value report by multiplying the ACCOUNTING COST (field 8) by the QTY TOT ONHAND (field 5) in your inventory master file and displaying it in a new field, #39. In this case the new field definition would be:

```
#5*#8
```

Play around with this feature and give me a call if you need anything else.□

FORTUNE SYSTEMS
an SCI Systems Company

No matter what your product or service, staying competitive requires control

Let QBC, your company's one source computer supplier, provide your business with solutions to:

Order Entry	Payroll
Accounts Payable	General Ledger
Accounts Receivable	Sales Analysis
Invoicing	Legal Time Billing
Inventory Control	Manufacturing Software
Office Automation	(MRP II Plus)

QBC is your one source for hardware, software, maintenance, customer training and data conversion.

MASTER DISTRIBUTOR FOR UPSTATE NEW YORK AND ERIE, PENNSYLVANIA

QBC

QBC Systems Inc.
375 NORTH FRENCH ROAD, AMHERST, NEW YORK 14120
(716) 691-5201 1-800-443-5770

Expand your Fortune® without spending a fortune.

New equipment costs big bucks. The Good Fortune Trader sells fully tested burned-in equipment, refurbished to Fortune® standards at prices that won't make waves with your cash flow. Call or write today for a listing of our specials.

THE GOOD FORTUNE TRADER

60 DAY WARRANTY
a division of Computer Services of Atlanta
(404) 447-4151
5825 Live Oak Parkway, Suite 2-C, Norcross, Georgia 30093

System Administration: Part 4

No Longer Requires a Degree in UNIX Wizardry

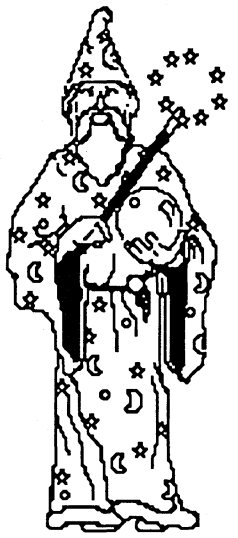
This series is a must for all owners and users of the Fortune computer. Dave is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the Fortune Computer, and is president of the Houston UNIX User's Group. He contributes independently to ***/u/fortune news.***

by Dave Kloes

Last time, we ended with a brief discussion of the "termcap" and "printcap" files in the **/etc** directory. Many of the files in the **/etc** directory, including the "printcap" and "termcap" files, are files that can be "edited". In this session, we will continue our discussion of these two files and will also give you some of the basics of the **ed** editor. Knowledge of at least one UNIX (FOR:PRO) editor is essential to the proper administration of your system.

Before we look at the Fortune terminal entry in the termcap file, there are a couple of things we need to point out:

1. For those of you that have THOROUGHBRED BASIC (SMC BASIC) on your system, the **/etc/termcap** has no relationship to whether a terminal functions properly when using BASIC, IDOL or the BAS software. Under the older versions of BASIC (prior to the 6.5.12 version), terminals functioned based on codes used in the "ipl" files (located in **/b/ipls** directory). Only the Fortune terminal and a couple of others (such as Televideo and Wyse) could be used. One of the main enhancements provided with the 6.5.12 release is the ability to use a program called the "TERMINAL CONFIGURATOR" to use many other terminals other than a Fortune terminal for your BASIC/IDOL/BAS applications. We point this out so that you will not spend countless hours trying to get a terminal to work with that software by modifying the termcap file.



This month Dave talks about the termcap and the printcap files. He also discusses the ed editor and explains the basic commands for editing a file.

2. No matter how good your termcap definition is for a terminal other than a Fortune terminal, it will not work properly when using Fortune:Word. Fortune has embedded some terminal specific codes in the Fortune:Word software that prevent this. The bottom line is that you can forget about using terminals other than a Fortune terminal in Fortune:Word. We do not know of anyone that has gotten other terminals to work 100%.

3. If your termcap entry is correct, you should be able to use Mutiplan on other terminals.

4. It may take a lot of work, but some terminals will function adequately with Fortune's menu system. In addition, you can enter the command `.term` at the bottom of the Global Menu and select terminal types that are in the **termcap** file. Don't expect anything near perfection. There is also a terminal type called "hard" that is a generic terminal type that can be used for any terminal in the menu system.

5. The termcap entry may affect the performance of other third party soft-

ware that requires correct terminal definition (Informix, for example).

6. A correct termcap entry is required in order for the **vi** and **screen** editors to function properly.

As we stated in our last article, the termcap entry for a terminal tells UNIX about the terminal type we are using. Take a look at the first line of your Fortune termcap entry (FT):

```
FT|fos|fortune|Fortune system:\
```

The first line gives us several names that we can use to tell UNIX which termcap entry we want to use. These are the names that can be used when "Defining Device Connections" in the menu system. When referring to the Fortune terminal, we can use FT, fos, fortune or Fortune system. Notice that this line is indented to the left. So, when we look at the termcap file (i.e. more /**etc/termcap**), we can see all of the terminals that already have definitions.

In addition to specifying a terminal type in "Define Device Connections", we can change the terminal type at the UNIX prompt. Let's say, for example, that the port our terminal is connected to is defined as an "FT" terminal but that we have temporarily connected a Wyse terminal that is emulating a Televideo 920. We want to use the **vi** editor, so the system must know that we are using the Televideo 920 terminal. First of all, we need to know whether there is an entry in the termcap file for this terminal so we enter:

```
# grep 920 /etc/termcap
v1|tvi912|912|920|tvi920|old
televideo:if=/usr/lib/tab-
set/std:
v2|912b|912c|920b|920c|tvi|new
televideo:
```

This **grep** command retrieves all lines with "920" in them. We got lucky, the termcap file already has a definition for this terminal. We could have also done a **grep** for other patterns to see if the terminal was defined. For example, we could have looked for "tele". Now that we know the termcap name for the terminal, we can tell UNIX about it:

```
# TERM=tvi920; export TERM
```

This resets the "TERM" variable in UNIX to "tvi920". The "export" portion tells UNIX to keep this variable available for other UNIX functions. (See previous articles on "**Introduction to Shell Programming**" for a more complete description of UNIX variables). Now we can use our terminal for **vi**, Multiplan, etc. This method of telling UNIX what terminal we are on works especially well when we are calling into other UNIX systems that may not have our terminal defined. We often transmit the "FT" termcap entry to other UNIX systems and add it to their termcap file so that we can use our Fortune terminals on them.

The other lines in the FT entry tell about other characteristics of the Fortune terminal. Notice that they are all two-digit codes. If we were to look in the

NJBC

New Jersey Business Computers

Authorized Fortune Systems Master Dealer

Authorized Fortune Service

Authorized Fortune Training Center

NEW JERSEY BUSINESS COMPUTERS has the Largest Affiliate Dealer Network in the New York and New Jersey Metropolitan Area.

For The Most Reasonably Priced TRAINING, SERVICE and SUPPORT Available.

call

301 Route 17 North, Rutherford, New Jersey, 07070

201-438-1410

Fortune Programmer's Manual, Section 5 under "termcap", we can find what the two-digit codes mean. Here are some samples:

is = Terminal initialization string
li = Number of lines on screen or page
co = Number of columns in a line
am = Terminal has automatic margins
bs = Terminal can backspace with ^H
bw = Backspace wraps from column 0 to last column
k0-k9 = Sent by function keys F1-F6
us = Start underscore mode
ue = End underscore mode
ug = Number of blank characters left by 'us' or 'ue'

Each of the sequences that follow the two character code are the sequences for that particular terminal. Even though the termcap entries look like Greek at first glance, writing or understanding a termcap entry is simply a matter of sitting down with our terminal manual ("Using Fortune Terminals" in this case) and our manual that describes what each of the codes means. We won't go through all of these codes, but let's take a look at the two underscore sequences 'us' and 'ue' as an example. The sequence after the = sign for 'us' says that if we depress the 'CTRL' and backslash keys at the same time followed by the H and P keys, the terminal will underscore everything from that point on. To terminate or end the sequence, the 'ue' sequence says to depress the 'CTRL' and backslash keys at the same time, followed by the I and P keys.

The **/etc/printcap** file defines printers the same way that the **/etc/termcap** file defines terminals. By looking at the printcap file, we can tell which printers can be used on the system. The printer must be defined in the printcap file before it can be used correctly in Fortune:Word, Multiplan, etc.. There are two entries called "PLAIN1" and "PLAIN2" that are generic entries for those printers that do not have printcap entries. There is also a generic "dumb" entry in the termcap file for terminals that do not have descriptions. The problem is that most of the special functions such as boldface and underline will not work with the generic descriptions.

Our intent is not to make you termcap or printcap writers. For those of you that are so inclined, we hope this takes a bit of the mystique away from these two files. Most of you that choose to use other terminals or printers on the Fortune system should find that the most common terminals and printers are already defined.

Since many of the files we will be talking about on the system are files that must be edited, it is appropriate to show you how to use at least one of the UNIX editors. We chose the **ed** editor for a number of reasons:

1. It is the only editor that is available on all UNIX/Xenix systems. The **vi** editor is available only if you purchased the "Development Utilities Software". For those of you that are interested in using the **vi** editor without having to purchase the Development Utilities, there is a user friendly version called **screen** that is available through **The Cambridge Consortium**. (**Editor's note: Screen** can be found on either the **D.C. Grab Bag** diskette or the new **Editors' Choice** diskette).

2. If you use editors such as **vi** and **screen**, they are not always on other UNIX systems. If you log into these systems and do not know how to use the **ed** editor, you're lost.

3. Since **vi** and **screen** are termcap dependent, what can you do if you log into a system that does not have your terminal defined? The proper functioning of the **ed** editor is not dependent on a termcap entry.

4. **DO NOT USE FORTUNE:WORD TO EDIT UNIX SYSTEM FILES!** Fortune:Word embeds its own unique characters which will result in trashed system files.

The **ed** editor is a line editor. That means all editing is relative to the line you are positioned on. You cannot use your cursor (arrow keys), keypad key (keys on the right-hand side of your Fortune keyboard) or function keys in the editor.

We will not discuss all of the commands that are available in the **ed**

Vantine Computer Services

Professional Solutions for Business Microcomputer Users

Sales Service Consulting
Software Customization Support
Software/Hardware Upgrades

Over Three and a half years supporting Fortune Users

Vantine Computer Services
70 Walnut Street, Wellesley, MA 02181
(617) 239-8084
Authorized Fortune Dealer

editor. If you learn the minimum commands and concepts that follow, you will be able to use the **ed** editor without any problems. Try the exercises as we discuss them and be sure you are in your user directory.

To invoke the editor, the syntax is:

```
ed <filename>
```

BASIC EDITOR COMMANDS

```
a = append (add) text
s = substitute text
1,$p - print all lines
5p - print line 5
i = insert text
1d - delete line 1
w = write file
3,4d - delete lines 3&4
q = quit
4,$d - delete line 4 to end
. = end append/insert mode
1,2p - print lines 1 & 2
p = print current line
$p - print last line
```

EXAMPLES OF COMMANDS:

1. To add text:

```
ed Mary
?Mary
a
Mary had a little lamb
.
1,$p
Mary had a little lamb
w
23
q
```

In this example, we start up the editor by entering "ed Mary". The editor responds with "?Mary". The question mark is the editor's way of telling us this is a newly created file. After this line is displayed, the cursor will be sitting directly under the "?". At that point, even though we do not get a prompt (like the # in UNIX), we are in the "command mode" and the editor is waiting for us to enter a command. We enter the command "a" on a line by itself which means "add" or "append" text after the current line. Remember the editor is a line editor, so all commands are relative to the line that we are on. Since there are no lines in this newly created file, all text entered after the "a" command is entered into the file until the editor sees a "." on a line by itself. In our example, we enter "Mary had a little lamb". Take our word for it, a common error is to forget the "." to end the command mode.

**$** in the "ed" editor refers to the last line and "p" means print so the "1,$p" means print lines 1 to the end. The "p" to display lines, and the "d" to delete lines have the same syntax. You can see how they work by looking at the samples listed above. Since we only have one line, the editor shows us the line that is in our file. The "w" means to write or save the file. None of the changes we make to a file are saved to disk until we do a "w". When we do the "w", the editor responds with the number of characters that are in the file. In this case, counting the carriage return at the end of the line, there are 23 characters.

The "q" command means to quit the editor. If you try to quit the editor before you do a "w", (which you may want to do for example, if you have made changes that you do not want to save) the editor will respond with a "?". Anytime the editor gives us a "?", it means we have done something wrong. If we do a second "q" without a "w", the editor assumes that this is what we want to do. And since it has questioned us once, it will exit without making the changes.

2. To insert text:

```
ed Mary
23
1p
Mary had a little lamb
i
This is a dumb poem.
.
1,$p
This is a dumb poem.
Mary had a little lamb
w
44
q
```

Now we want to go in and modify our "Mary" file. We again fire up the editor using "ed Mary". Instead of the "?Mary", we now get "23". When we edit an existing file, the editor gives us the character count of the file. A "p" command by itself will print the current line. The command ".=p" will give us the line number of the current line. In our example, we enter "1p" which prints line 1. Then we enter the "i" command which means to insert the text that follows BEFORE the current line. This is different than the "a" command which means to add text AFTER the current line. We enter the line "This is a dumb poem." followed by a "." on a line by itself which ends the insert mode. Notice that the editor does not see the period at the end of the line we inserted as ending the command mode. We then display all of the lines in the file by entering "1,$p", write the file and quit the editor.

3. To substitute text:

```
 ed Mary
44
1,$p
 This is a dumb poem.
 Mary had a little lamb
 s/little/big/
 p
 Mary had a big lamb
 w
 41
 q
```

In order to change text on an existing line, we can either delete the line and reenter it or we can "substitute" text by using the "s" command. In this example, we again edit our "Mary" file and print all the lines in the file. Whenever we fire up the editor, we are automatically positioned on the last line of the file. If in doubt, use the "p" command to print the current line so that you know what line you are on. Here we want to substitute the word "little" for the word "big", so we enter "s/little/big/". The editor looks for the pattern between the first two slashes and substitutes the pattern between the second two slashes for it. A common error is to forget the last "/" which will result in a "?" from the editor. We then print the line, write the file, and quit. As a sidenote, you can substitute all occurrences of a pattern within a line or globally substitute all occurrences of the pattern in the file with a "g/s/little/big/" or a "s/little/big/g". Look at your FOR:POR manual under the "ed" command to learn more about this.

Here's a problem for you. How would we change the following line to remove the "/":

```
Please send the memo to room 43/A.
```

This creates a problem because the "/ " is used as a delimiter whenever we use the "/" with the "s" command. We would get an error if we did this:

```
s/3/A/3 A/
?
```

Our intent was to substitute the pattern "3/A" with "3A" so as to remove the "/". The editor becomes confused because it sees the "/" as the delimiter for our substitution. Whenever we use special characters (metacharacters) and want the editor to ignore a character as an editor functions, we must precede the character with a backslash. While it may look confusing, the backslash simply indicates that the character should be treated literally as that character and not as an editor function:

```
s/3\/A/3A/
p
Please send the memo to room 43A.
```

By putting the backslash before the second slash, we are telling the editor to treat the second slash as a slash and not as a separator between the patterns to be changed. Although the "p" command can be entered on a line by itself to print the current line, we can also include multiple commands in the same line. The same result as the above could be obtained by putting the "p" at the end of the same line:

```
s/3\/A/3A/p
Please send the memo to room 43A.
```

Here's one last example where we remove a word:

```
s/room //p
Please send the memo to 43A.
```

This removes the word "room" and the space that follows it and substitutes nothing in its place.

4. To search for a pattern (word) in a line:

```
ed Mary
41
/lamb/
 Mary had a big lamb
q
```

We can search for a line that contains a pattern or word by putting the pattern we are searching for between slashes on a line by itself. In this example, the "/lamb/" will position us on the first line that the editor finds with this pattern. This is handy when we are not sure of the line number of a line we want to edit.

5. Deleting text:

```
ed Mary
41
1,$p
This is a dumb poem.
Mary had a little lamb
1d
1,$p
 Mary had a little lamb
a
Whose fleece was white as snow
And everywhere that Mary went
Her lamb was sure to go
 .
 w
129
q
```

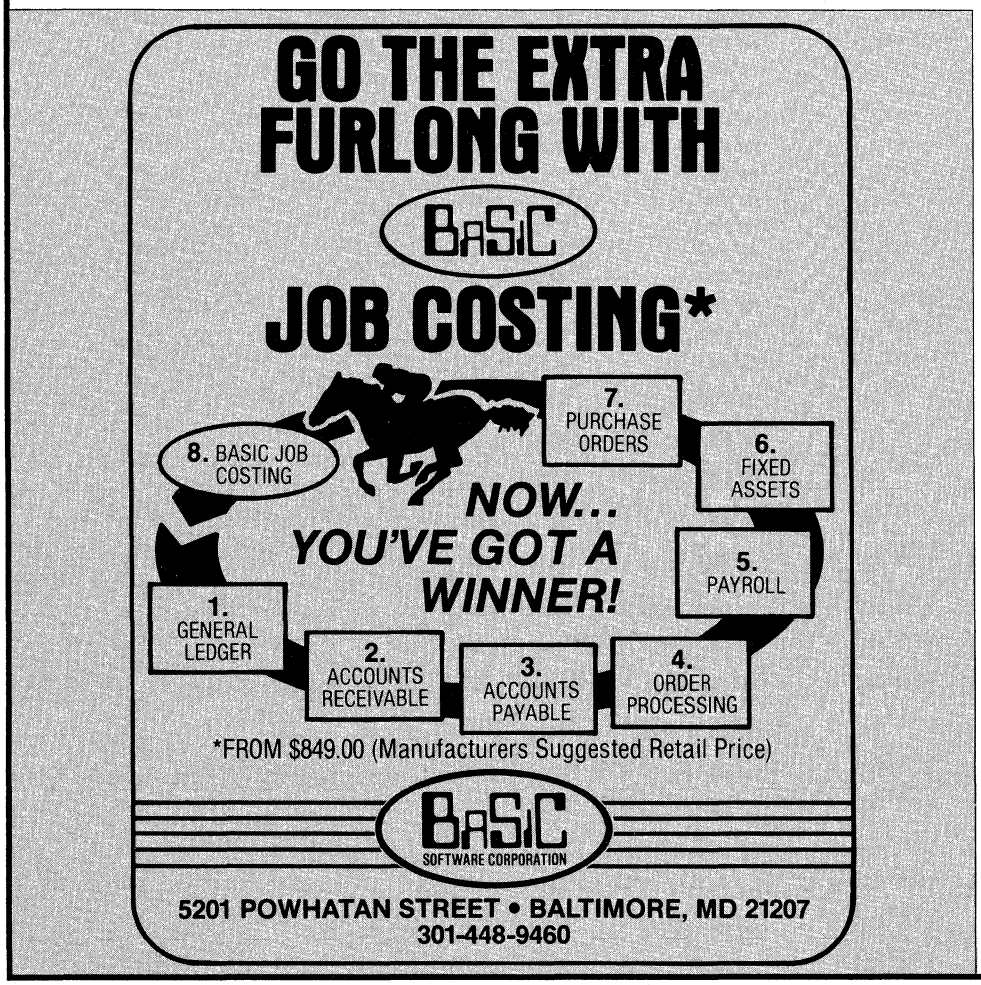
Here we edit the file "Mary" and print the contents of the file (1,$p). We can delete a line that we are not positioned on by entering the line number followed by the "d". A "d" on a line by itself deletes the current line. We then print the contents of the file; add three more lines; write the file; and exit the editor.

For those of you that are novice UNIX users, remember that we can view the contents of a file by using the "cat" or "more" ("more" lets us look at a large file a screen at a time) commands:

```
# cat Mary
This is a dumb poem.
Mary had a little lamb
Whose fleece was white as snow
And everywhere that Mary went
Her lamb was sure to go
```

Our intent is not to use the "ed" editor to write dumb poems. As we proceed, we will talk about many files, including those in the /etc directory, that we can edit. We can also use the "ed" editor to write or modify shell programs. There are many editor commands that we have not discussed. You can get by on most editing jobs with those mentioned above.

Next time we will continue our discussion of other files and commands in the **/etc** directory.□





TRADE - UPGRADE - EXPAND
Your FORTUNE SYSTEM

Processors:
- *Fortune 32:16 SX*
- *Fortune Formula 4000*
- *Fortune Formula 8000*
- *Fortune Multiplier (Run multi-user For:Pro & Fortune applications on a PC)*

Hard Disks:
- *30MB*
- *45MB*
- *70MB*
- *85MB*
- *145MB*

Tape Drives:
- *20MB External*
- *60MB External*
- *60MB Internal*

Memory:
- *256KB*
- *512KB*
- *1024KB*
- *4096KB*

Controllers:
- *Comm-A 4 Ports*
- *Comm-A 6 Ports*
- *Bit-Mapped Graphics Controller*
- *Fortune:Link Controller*

BONUS - BONUS - BONUS

All Fortune Licensed Software Upgrades and/or Replacements
At No Additional Charge with Processor Upgrade/Trade!

We can perform these trades and/or upgrades within 24 hours of receiving your system. In fact, if you ship your system overnight on a Friday for Saturday delivery, we will have it back to you on Monday morning, all ready to plug in and run; not losing the use of your system for a single work day. And, we will issue you a full 90-day warranty on the entire system.

JFM provides services to Fortune Systems owners nationwide either direct or through our authorized Affiliated Dealers. If you haven't used our services, I invite you to discover JFM; you will be glad you did.

Sincerely,

Morris Frydman
President

Talk to us, then talk to our clients

JFM
BUSINESS
SYSTEMS
The Merchandise Mart
Lobby Suite 144
Chicago, Illinois 60654
Call (312) 644-0870

Authorized Fortune/SCI Systems Master Dealer

PROGRAMMERS

The leading manufacturer of status display panels for the fire alarm and security industry is seeking programmers and junior programmers to develop new applications and refine existing programs.

Successful candidates will be experienced in IDOL/BAS and SMC Business Basic for Fortune and Basic 4 systems. Familiarity with UNIX and system set-up and maintenance is desirable.

Candidates interested in joining a growing team-oriented organization located in Falls Church, Virginia should write or call with resume and salary history, to:

WSA
308 Hillwood Avenue
Falls Church, Virginia 22046
703 241-0687

DO YOU KNOW...

there is an organization that has a thorough knowledge of

FORTUNE BAS
MULTIPLAN IDOL

We have been providing widespread support to our Fortune clients for more than 4 years.

If you need:

- training
- support
- maximum computer performance
- peace of mind relating to processing
- customization
- application results
- minimum errors and confusion
- better ideas on how to increase the power of your Fortune

Call Steve Rosenfeld at (314) 731-3636

SUPERIOR COMPUTER SYSTEMS, INC.
P.O. Drawer T
Florissant, MO 63031
Making computers work for you

Coming soon. . . A revised listing of **Third Party Software** for the **Fortune/SCI Line**; A complete index to **/u/fortune *news***; Articles on Managing Glossaries with glossary entries; Questions from our readers; the latest on the blending of the Fortune and SCI lines; a review of **Timemaker**, a scheduling/todo/directory business program, and more. Stay tuned.

/u/fortune news

73 Jason Street
Arlington, MA 02174

/u/fortune* news

The Newsletter for Users of Fortune/SCI Computers | October 1987/Volume 4 Number 10

Fortune:Word Hints

Using the Index Generator

We suspect that few people use all of the features of **Fortune:Word**. We know that it's hard to do your work and learn the ins and outs of this powerful word processor. However, we thought that we might induce you to learn about the **Index Generator** if we highlight some of its strengths.

The **Index Generator** is used to create an index of your document. You supply it with a list of words or phases that you want it to search for, and it produces a document with the page number for each keyword. Obviously if you only type short letters, this won't be of much use to you. But if you type longer documents, the Index function may be helpful for many reasons. First, you may actually want to supply an Index with the document. Second, it may help you to edit the document more efficiently if you know where in the document certain phrases are used.

For instance, if you have a long document and you want to edit the section containing the word "filibuster", it would be easiest to go directly to the page that contained that section, using the **GOTO** command. (Another way to go there would be to use the **SEARCH** function.) If you found that filibuster was on page 22, hitting the GOTO key followed by 22 EXECUTE will take you right there.

Using the Index Generator

The **Index Generator** needs two documents in order to work — your main document and a word list document. The former is just your document, there isn't anything special about it. (Well, I suppose that's just our opinion.) The second is simply a document that contains a list of words or phrases. Each word or phrase appears on its own line. A sample word list follows:

It may help you to edit the document more efficiently if you know where in the document certain phrases are used

```
search
Index Generator
glossary entry
entry a
archive
/u/fortune news
Section
```

Note that the words are not in alphabetical order. The **Index Generator** runs more slowly if the word list is already sorted. In a moment we will present a glossary entry which will create the word list for you automatically, but you can see that you could also type it in manually.

Once you have the word list, you go to the **Document Processing Tools**

See **Hints**, Page 22

Coming Attractions

****

Next month we publish an interview with Ron Borelli who is Vice-President of SCI's computer division.

****

Richard Suslick will continue with another installment of his UNIX file system series.

****

In December, we will publish a comprehensive index of **/u/fortune *news*.**

****

In January and February we will publish the current list of software that is available for the Fortune/SCI computer.

****

And don't miss Ray Wannall's BASIC Advisor or Dave Kloes' continuing series on System Administration.

In This Issue...

Editorial...Page 2.

System Administration - Dave Kloes discusses the vi editor...**Page3**.

/u/help - answers to questions about our rolodex program, the rem shellscript, problems with tape drives, and more...**Page 8**.

BASIC Advisor - Ray Wannall discusses his recent trip to a Concept Omega conference...**Page 14**.

Development Utilities - what they are and how to use some of them...**Page 16.**

Fortune:Word Hints - how to use the index generator...**Page 1**.

*/u/fortune *news* is published by The Cambridge Consortium, Inc. It is not affiliated with Fortune Systems/SCI Technology, Inc..

/u/fortune *news*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz

Technical Editor
Scott Bradner

Advertisting Manager
Mark Palmerino

/u/fortune *news* is published 12 times per year by The Cambridge Consortium, Inc. Neither **/u/fortune** *news* nor The Cambridge Consortium are affiliated with Fortune Systems/SCI Technology, Inc.

Subscriptions cost $48 for one year within the U.S.A. For foreign subscriptions, add $27 each year for added postage and handling costs.

Back issues are offered at:

Vol 1, Numbers 1 - 3	$1/each
Vol 1, Number 4	$.50/each
Vol 1, Numbers 5 - 8	$3/each
Vol 2, Numbers 1 - 7	$3/each
Vol 3, Numbers 1 - 12	$4/each
Vol 4, Numbers 1 -	$4/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $10 each. Please add $5 per order for foreign destinations.

Although every attempt is made to insure that our contents are error-free and without risk, The Cambridge Consortium is not responsible for losses that occur due to correct or incorrect interpretation of **/u/fortune** *news*.

For more information, call or write:

/u/fortune *news*
73 Jason Street
Arlington, MA 02174
(617) 663-7700, (617) 646-5603

The text for this newsletter was prepared using a Fortune 32:16 for text input. The information was then transferred to an Apple Macintosh Plus with kermit and formatted for typesetting using the Aldus Pagemaker program.

UNIX™ is a trademark of Bell Laboratories, Inc. Fortune 32:16™, Fortune:Word™ and FOR:PRO™ are trademarks of Fortune Systems/SCI Technology, Inc. Multiplan™ and MS-DOS™ are trademarks of Microsoft. IBM-PC™ is a trademark of IBM. Macintosh Plus™ is a trademark of Apple Computer. Pagemaker™ is a trademark of Aldus Corporation.

©1987 by the Cambridge Consortium, Inc. 10/87

From the Editors...

Of Past, Present and Future

It seems difficult to believe that we've been publishing **/u/fortune news** for almost four years. It's also incomprehensible that Fortune computers (as well as IBM Pc's) have only been around for about five years. Given that last fact, some may feel that we at The Cambridge Consortium chose the wrong horse in that race. We don't think so. We still use our **Fortune 32:16** everyday to do our record keeping with **Progress** and our word processing with **Fortune:Word** and **vi**. We use several terminals and each one has **Fortune:Windows** working on it.

Of course some things have changed over the years. We now have several disk drives attached to the system, 2 megabytes of memory, and almost no bugs. Our newsletter which began as a typewritten and xeroxed publication has grown into a 24-page periodical produced on an Apple Macintosh. We like to think that those changes are positive.

Recently we wrote to most of you and offered you a chance to resubscribe to **/u/fortune news** for from one to three years. Apparently many of you feel that you made the right choice also, because 25-30 percent of our readers have resubscribed. We want to thank those of you that made that choice—the capital will allow us to continue to expand. We were also heartened by the high response rate from our readers. We had expected a 5-10 percent response, which would have been well above normal. Your confidence and generosity has moved us.

So now what? We told you we would expand and we will. In December we'll be publishing an index of all of **/u/fortune news**. Early next year we'll publish an index of software available for the **Fortune/SCI** line of computers. We'll be writing articles about UNIX System V. We'll be writing about new developments with **Fortune:Word** (We hear there are some exciting additions due out in the coming months.) We'll tell you about spreadsheets, accounting, modems, databases, new products, and tricks of the trade. We'll continue to assemble more disks of free software. We want to keep you and your Fortune productive.

We are also aware that some of you feel that some of the articles are too technical in nature. We try to keep the newsletter balanced. As we expand, there will hopefully be more information for everybody. In **/u/fortune news**, we also try to fill in enough of the details so that even if we are talking about complicated things, someone who is not familiar with the subject matter will be able to use the information if they spend a short time to absorb it.

Of course the newsletter is for you. If you have comments or suggestions about our editorial content, please don't hesitate to contacts us. If you have questions that need answers, we'll try and help.

In addition to enhancing the newsletter, we hope to enlarge our pool of readers. As **SCI** brings its manufacturing weight and capital resources behind the product, we expect that more and more people will be using **Fortune/SCI** equipment and more people will be reading **/u/fortune news**. We thank all of you for being among that special group.

System Administration: Part 6

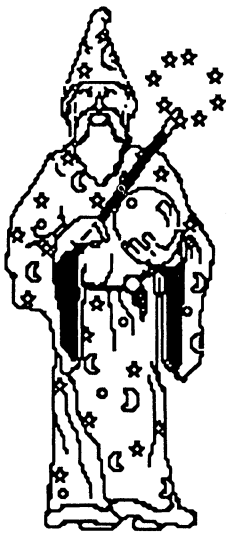
No Longer Requires a Degree in UNIX Wizardry

This series is a must for all owners and users of the Fortune computer. Dave is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the Fortune Computer, and is president of the Houston UNIX User's Group. He contributes independently to ***/u/fortune news.***

by Dave Kloes

For those of you who missed Part 5, now is a good time to go back and take a look at it. We ended the article with a discussion of the "ed" editor. This editor is on all Fortune systems and can be used to edit system files. In this article, we will show you the basics of the "vi" editor (for those of you lucky enough to have it on your systems - "vi" is only available with "Development Utilities"). Then, with either of these two editors at our disposal, we will discuss how to use the editors to modify system files in the /etc directory.

One of the main differences between the "ed" and "vi" editors is that while "ed" is a line-oriented editor (all edit commands are relative to the line we are positioned on), the "vi" editor is a screen editor (all edit commands are relative to the character the cursor is positioned on). Whole chapters and books have been written about the "vi" editor so our intent is not to teach all there is about "vi". Rather, we are going to discuss a few of the commands and concepts. By learning these few commands, you should have enough knowledge to edit files in For:Pro using "vi". We will try to shorten your learning curve by drawing some parallels between commands in "ed" and commands in "vi".



This month Dave talks about the vi editor. He explains the basic commands that are necessary for editing system files - like those in the /etc directory.

Here are the basic commands for the "vi" editor:

`a` is used to append text after the letter the cursor is on.

`i` is used to insert text before the letter the cursor is on.

`x` is used to delete character that the cursor is on.

`dd` is used to deletethe line the cursor is on.

The **arrow keys** are used to position the cursor.

`:quit!` is used to quit or exit the editor without making changes.

`:x` is used to quit and save changes.

`$` is used to position the cursor to end of line cursor is on.

`/pattern` is used to search for "pattern".

`<ESC> key` is used to end command mode.

There are many more commands for "vi" but these should be enough to get

us started. The syntax for starting to edit a file using "vi" is:

```
vi <filename or pathname>
```

Let's go through the same exercises we used for the "ed" editor using the "vi" editor:

```
# vi Mary
```

Screen will clear and we will see:

```
[]
~
~
~
~
"Mary" [New file]
```

In these examples, where appropriate, we will use [] to indicate the cursor. Also the screens will be shortened to conserve space. In the above example, the "tilde" (~) character would be repeated on successive lines filling the whole screen.
Notice that since we are creating a new file, the name of the file and [New file] are displayed at the bottom of the screen. Also, the cursor is positioned in the upper left hand corner and "null" lines are indicated with the tilde symbol. The difference between a null line and a blank line is that a null line does not exist in the file, while a blank line (which would be indicated by a blank line with no tilde) is an actual blank line that is in our file.

1. To add text:

At this point we are in the command mode and "vi" is waiting for us to enter a command. If we want to enter text at this point we can either use an "a" to append text or an "i" to insert text since it is an empty file. Unlike the "ed" editor, we will not see the command character on the screen when it is entered. In this example, to enter the line:

```
Mary had a little lamb
```

we would depress the "a" key (nothing will be seen) and then start typing the line to add it to the file.

In the "ed" editor, we ended the "a" or "i" mode by putting a period on a line by itself. In the "vi" editor, we end the command mode by depressing the <ESC> key. So after the word "lamb", we would depress this key to end the command mode. Again, nothing is visible on the screen and the cursor would be over the letter "b" of the word "lamb".

Since this is the only line of text that we want to enter at this point, we would enter ":x" to quit the editor and save the file. When the ":" is depressed, our cursor goes to the last line on the screen. We would then depress the "x" and <RETURN> keys to exit the editor and save the file. If we wanted to quit the editor without saving the changes, we would enter ":quit!".
When we enter the ":x", the following line is displayed when we quit the editor:

```
"Mary" [New file] 1 line, 23 characters
#
```

In "ed", the editor indicated errors by

Our Memory Doesn't Cost You A Fortune

Would you like to:

Speed up your computer?
Add more users?
Use the latest software, like Fortune:Windows™?

Compatibility:	The F1MB is absolutely 100% compatible with Fortune's own 1 Meg memory board.
Affordability:	USA is proud to have equaled and exceeded the quality and performance of our competition, while offering substantial savings for both the dealer and end user.
Dependability:	All boards are burned in for 72 hours and tested for 48 prior to shipment. Our One Year Repair or Replace Warranty underlines our confidency in the product.
Repairability:	The F1MB facilitates repair in the field as it is 100% RAM socketed. Now you can repair a board by simply replacing a defective chip rather than face an expensive board exchange -- reducing maintenance costs and system down time.

If you have a Fortune SX, XP45 or recent model XP30, you can use our board by simply plugging it in. Older systems can be upgraded to use our board with a simple and economical modification.

For Details Call:

Upgrade Service Associates

850 Lexington Street, Suite 15, Waltham, MA 02154 Phone: (617) 239-8147

Our Guarantee: If you're not completely satisfied within 30 days, USA will gladly refund your money.

displaying a "?". In the "vi" editor, the message "ERROR" will be displayed at the bottom of the screen. Also, once the <ESC> key is depressed, we are once again in the command mode. If we depress the <ESC> key while in the command mode, "vi" will beep us to let us know we are already in the command mode. As a sidenote, do not use any of the function keys, blue keys (except <ESC>), keypad (right hand side of keyboard) keys, the <CANCEL/DEL> key or the INSERT, PREV SCRN, DELETE, NEXT SCRN keys.

One of the main differences between the "ed" and "vi" editors is that while "ed" is a line-oriented editor (all edit commands are relative to the line we are positioned on), the "vi" editor is a screen editor (all edit commands are relative to the character the cursor is positioned on).

2. To insert text:

Let's say we want to add the line "This is a dumb poem" as the first line of the file. Again we would fire up "vi":

```
# vi Mary
```

Screen clears

```
[M]ary had a little lamb
~
~
~
~
~
"Mary" 1 line, 23 characters
```

Now, since this is an existing file, our text is displayed, and the line on the bottom of the screen repeats our file name and tells us the number of lines and characters in the file. Our cursor is in the upper left hand corner of the screen (in this case, over the "M" of the word "Mary"). By the way, there is a size limitation to the files that "vi" can handle. If you get an "out of memory" error, you will need to subdivide the file using the **split** command (available only if Development Utilities is installed), edit the separate pieces and put the pieces back together again to form the original file with the changes.

Back to our example. In order to insert text at this point, we would depress the "i" key. Again you will see no changes on your screen. Remember that the "a" and "i" commands are relative to the character that the cursor is positioned on. In this case, since the cursor is over the "M" in "Mary", we are telling "vi" that we want to insert text before the word "Mary". After the "i" is depressed, we can type in our "This is a dumb poem" line. Since we also want to insert the line feed at the end of this line, don't forget to depress the <RETURN> key before depressing <ESC> to exit the insert mode.

After depressing the <ESC> key and getting back into the command mode, we can enter our ":x" to quit the editor and save our changes.

For those of you using "vi" for the first time, you will no doubt be frustrated until you have learned the basic commands and concepts. This is one of those tools that you learn by doing. In our seminars, most students pick up on these after practicing on their own. After using "ed", you will need little motivation to learn how to use this screen editor.

3. To substitute text:

Here's where "vi" really has it over the "ed" editor. Let's say we once again want to replace the word "little" with the word "big". First we invoke "vi":

```
# vi Mary
```

Vantine Computer Services

Professional Solutions for Business Microcomputer Users

Sales Service Consulting
Software Customization Support
Software/Hardware Upgrades

Over Three and a half years supporting Fortune Users

Vantine Computer Services
70 Walnut Street, Wellesley, MA 02181
(617) 239-8084
Authorized Fortune Dealer

```
[T]his is a dumb poem.
 Mary had a little lamb
~
~
~
"Mary" 2 lines, 44 characters
```

Our cursor is positioned over the "T" in "This". One advantage of "vi" over "ed" is that we can use our arrow keys to position the cursor. In this case, we want the cursor positioned over the "l" in the word "little". So we use our down arrow key (or <RETURN> key) to go to the second line and our right arrow key (or space bar) to position the cursor over the "l". We can delete the word "little" by depressing the "x" key six times. Remember that we can do this because we are in the command mode and depressing the "x" in the command mode says to delete the character the cursor is on. One shortcut we can use is to precede the "x" with a number to tell "vi" how many characters to delete. So if we enter "6x", the word "little" will also be deleted.

Now we want to enter the word "big" in its place, so we depress the "i" key to enter the insert mode and then type the word "big". Again, we depress the <ESC> key to exit the insert mode and then do our ":x" to quit and save.

We want to mention one more command here that is not listed above. Like the "ed" editor, there is a "w" (write or save) command and a "q" (quit) command. So we could exit the editor by doing a ":w" and then a ":q" or a ":wq". Essentially, the ":x" does the equivalent of a write and a quit at the same time. If we are entering a lot of text and we want to save the changes but do not want to quit the edit session (very important to those of you who have frequent power failures), we can enter a ":w" to save the changes we have made up to that point. If you make changes and try to ":q" without doing a ":w", "vi" will give you the following message:

No write since last change (:quit! overrides)

Basically, if you have made changes and try to quit without doing a save (:wq or :x or :w and :q), "vi" forces you to enter ":quit!" to make sure you know what you are doing and that the changes will be lost.

4. To search for a pattern (word) in a line:

This command is almost like the "ed" editor. To search for a word or pattern, we simply enter "/" followed by the word or pattern. Unlike the "ed" editor, a "/" after the word or pattern is not required. The "vi" editor will position on the first occurrence of the word or pattern. For example, instead of using the cursor to position ourselves on the word "little" as shown above, we could have entered:

```
/little
```

and the cursor would be positioned on the "l" of the word "little".

5. To delete a line:

To delete a line, we first position the

NJBC

New Jersey Business Computers

Authorized Fortune Systems Master Dealer

Authorized Fortune Service

Authorized Fortune Training Center

NEW JERSEY BUSINESS COMPUTERS has the Largest Affiliate Dealer Network In the New York and New Jersey Metropolitan Area.

For The Most Reasonably Priced TRAINING, SERVICE and SUPPORT Available.

call

301 Route 17 North, Rutherford, New Jersey, 07070

201-438-1410

5. To delete a line:

To delete a line, we first position the cursor anywhere on the line we want to delete. While in the command mode, we then enter "dd" to delete that line. We can precede the "dd" with a number to tell "vi" to delete that number of lines. For example, to delete five lines, including the one the cursor is on, we would enter "5dd". In the above example, if we were positioned on the "T" of the word "This" and did a "dd", that line would be deleted.

6. To add a new line:

There are a couple of ways to add a new line. The first method would be to position ourselves on the line above the line we want to insert. We would need to be at the end of the current line so we could use either the right arrow key or space bar to move there. ("vi" will beep at you if you try to go past the end of a line). We could also enter a $ in the command mode to position to the end of a line. We would then depress "a" to enter the append mode and the <RETURN> key. This would open a line below the current line and we could then start entering our new text. Just to show you the versatility of the "vi" editor, there are two commands available for opening a new line either above or below the current line. If we are in the command mode, depressing the "o" (little o) key opens a line below the current line and automatically puts us in the insert mode. Depressing the "O" (capital O) key opens a line above the current line and automatically puts us in the insert mode.

Here are some of the more common errors when using "vi":

1. Forgetting to depress the <ESC> key to end the "a" or "i" mode. Until this is done, whatever we enter becomes part of our text.

2. Depressing function keys, etc. on the keyboard.

3. Setting the CAPS LOCK key and forgetting to unset it. "vi" uses many keys and control sequences while in the command mode. There is a big difference between a small letter and its capital equivalent when in the command mode. The little "o" and capital "O" are two examples mentioned above.

4. Not using ":quit!" to quit without making changes.

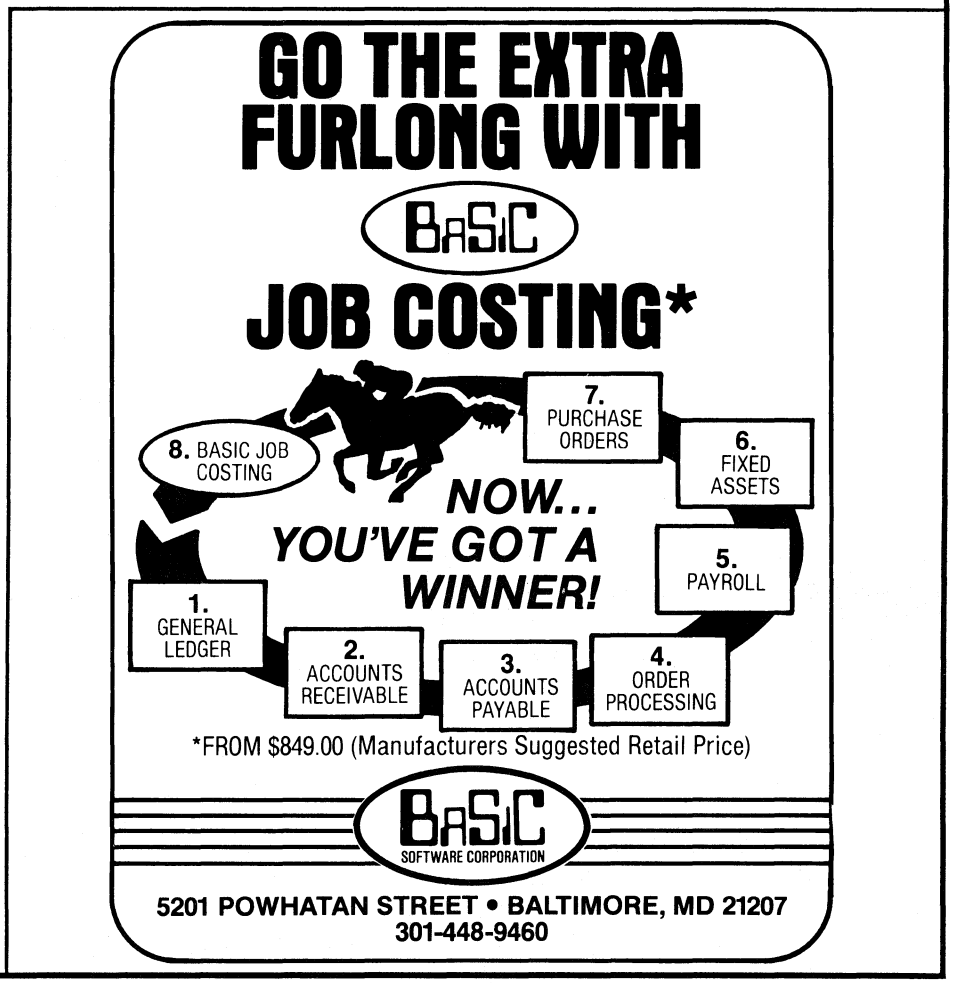
5. Using arrow keys while in the insert or append mode. If you do this, the control sequence is embedded in your text.

Another useful command when using "vi" is the "u" command. If you make a change or error that you want to "undo", you can "undo" the last thing you did by depressing the "u" command. You cannot undo something if you have entered something else in between.

You can execute FOR:PRO commands from within "vi" by preceding the command with a ":!" while in the command mode. For example, you can enter ":!date" to get the current system date and time while in the "vi" editor. You can temporarily exit to FOR:PRO and do normal shell commands at the prompt by entering ":shell"<RETURN> while in "vi". When you do a "CTRL-d" or "exit", you will be returned to "vi".

If you practice these few simple commands, you will know enough to be able to use "vi" to edit system files. Again, there are over 200 "vi" commands. So if you have the desire to learn more about this editor, refer to the "Programmers Manual". There are also "vi" reference cards available from quite a few sources. Sometimes, these reference cards are included in UNIX/Xenix books you can purchase at your local bookstore.

Next time we'll look at some of the system files we can edit in the /etc directory. It is important that you master either the "ed", "vi" or "screen" editors before attempting to edit system files that affect the daily operation of your system. □



Don't worry about your UNIX Problems

Read this month's /u/help



***Question**: I have been using the Electronic Rolodex for some time now. The other day the system crashed while I was using the rolodex, and now when I try to use it I get the message "Someone else is using the rolodex, sorry."*

Answer: This is a simple problem to fix. When the rolodex program starts up, it looks for the rolodex data in the HOME directory and reads it into memory. It also places a "lock" on the rolodex to prevent several people from using the data simultaneously. This "lock" is actually an empty file called **.rolodexdata.lock**. If the rolodex doesn't exit normally, the lock file is left in the HOME directory, and you get a message like the one above. Or you might get the following more lengthy message:

Your rolodex is already in use!

Either someone else is using and possibly modifying your rolodex, or else you forgot to exit a previous use of your rolodex, or possibly the system crashed while you were using your rolodex the last time.

You can find out who is using your rolodex by typing

```
ls -l ~/.rolodexdata.lock
```

The third column of the output will be a user's name. If the name is not yours, then you should find out what that person is doing with your rolodex and ask him/her to let you know when they are finished.

If the name is your own, then it is probably safe to remove the lock file. (It is possible that someone else is logged in as you, however...)

Type

```
rm ~/.rolodexdata.lock
```

and then you can use your rolodex.

These instructions will almost solve your problem. The one modification you need to make is to substitute your HOME directory for the "~", which is the symbol for the HOME directory in the c-shell. (Most of us use the Bourne shell.) The key part of the instructions is to remove the lock file. Also note that due to restrictions on the length of filenames, you may only see the name **.rolodexdata.l** because the "ock" in lock has been truncated.

If you are not sure what your HOME directory is (it's usually /u/joe, if your name is joe), then you can type `echo $HOME <RETURN>`. The shell will then respond with the HOME directory.

***Question**: I recently upgraded to Fortune:Word 3.0 and, for the most part, am quite pleased. One feature that comforts me is the "autosave" function which automatically copies my working file to the hard disk on a regular basis. Unfortunately, I seem to always forget to invoke it, which greatly limits its usefulness. Is there a way that I can automatically start the "autosave" function each time I use Fortune:Word?*

Answer: As a matter of fact, I have the same problem myself. The easiest way to fix this is to change your *prototype* document to incorporate the Autosave feature. This is done by simply editing the prototype document **0000**. Once in the document, hit the **STOP** key, and the number of keystrokes you want as the default interval. The default is 1024. Press **EXECUTE** to confirm your choice. Then **CANCEL/DEL** out of the prototype as usual. From that time on, any new documents you create in that library will have the Autosave feature turned on. Please note that *each* library has its own prototype document and therefore each prototype document in each library must be modified.

***Question**: I use your reminder shellscript, rem, which was described in Volume 2, Number 1. This program works with the **at** and **cron** programs. It used to work fine until I started using Fortune:Windows. Now when I create a reminder, I don't get the message until I go back to the window that was used to create the reminder. I've missed my last appointment because of that, and I've now resorted to writing you for a solution.*

Answer: Well your problem is that Fortune:Windows actually sets up extra "terminals" or devices, such as **ttp0a**. The **tty** command used in the **rem** command finds this pseudo device instead of the actual terminal

What An Opportunity



An opportunity like this doesn't happen very often.

Micro Resources is in a fix. The region's leading Fortune/SCI supplier's stock room is jammed with top-notch Fortune/SCI inventory — with more on the way.

So we're dropping prices to levels you probably thought you'd never see. Grab your wish list. This is a brief opportunity to upgrade your system you simply can't afford to pass up.

If you need more memory, additional terminals or faster motherboards, now's definitely the time to buy. But don't wait too long — **quantities are limited.**

Increasing your Fortune/SCI system's productivity isn't very difficult. And Micro Resources just made it a lot easier. Call Chuck Kennedy, 766-2335, today — before it's too late.

Micro Resources

255 Bradenton Ave., Dublin, Ohio 43017 614 • 766 • 2335

number, and so the echo is directed to the window rather than the terminal.

To fix this, we want to find a method to get the terminal number, rather than the window device. In order to do this we created a new **ENVIRONMENT** variable called **TERMINAL**. You can create the TERMINAL variable in your **.profile** by including the line: `TERMINAL='tty'; export TERMINAL`.

This simply uses the **tty** command to get the terminal number before Fortune:Windows is started. Then you substitute the **$TERMINAL** variable in the **rem** command instead of the **'tty'** command. (As a by-product, this will speed up rem because it will not have to do the **tty** command each time it is run). Both the old and new versions of rem are included as figures with this column.

The following suggestions come from Mike Dyberg at Accutrend System Services in Puyallup, WA.

System Notebook

We recommend that you keep a system notebook that contains problems that you have encountered, ideas you have, procedures, and tips that you have found. This will be an invaluable reference.

As you change or add personnel, this book will save you from having the same error created by your new people or at least provide them with information on how to fix it. It will take some time and effort to make the book work, but the payback will be in time saved and real dollars.

Similar in spirit is Accutrend's advice concerning a Procedure Manual:

We recommend that you keep a system notebook that contains problems that you have encountered, ideas you have, procedures, and tips that you have found. This will be an invaluable reference.

We highly recommend that each company develop and maintain a procedure manual for their computer operation. This manual should contain the steps, codes, schedule and the particular way your company uses the computer.

If you have any Software Express programs, the manuals have a schedule section which is very helpful in starting a manual. Be sure to include procedures for the manual operation of collecting the data to be input into the computer.

This manual may take a little time to put together, however it will more than pay for itself. It's a guide for new employees, a reference, a reminder for those things that are only processed once in a while and it will save on support cost.

Question *I keep quite a bit of information in my directory. In order to keep things organized, I use a lot of subdirectories. Sometimes I forget which files in my directory are subdirectories - is there any way I can find out which files in my directory are subdirectories?*

Answer Yes! And as is usual for UNIX, there are several ways you can do this. As you probably know, if you type a **ls -l** to the UNIX prompt, you will get a listing of all the files in your directory which will look something like this:

```
total 1
drwxrwxrwx 2 mbp      144 Aug 27 14:20 .tmi
-rwxrwxrwx 1 mbp        1 Aug 19  1986 0000
-rwxrwxrwx 1 mbp       87 Aug 19  1986 0000.dc.Z
drwxrwxrwx 2 mbp       96 Jun 26 14:43 abp
-rwxrwxrwx 1 mbp      578 Aug 21  1984 abp1.Z
```

Note the lines that begin with a **d** (e.g. `drwxrwxrx...`). These are files that are directories. So, one way of seeing which files are directories is to get a

New rem shellscript

```
:
#    This shell script will allow you to enter reminder messages into your
#    system and have them sent to you at predefined times.

#    in order for it to work, you must have at and cron installed on your
#    system.


echo "This command is used to enter a reminder message onto your screen"
echo
prompt  "Please enter time you would like reminder: "
read time
echo
echo "Please enter your reminder message - one line only"
read message
echo 'echo -n ''"'$message'" > '$TERMINAL';/usr/cmd/beep >' $TERMINAL | at $time
echo
echo "The following message:"
echo  $message
echo
echo "will be sent to you at "  $time
```

Old rem shellscript

```
:
#    This shell script will allow you to enter reminder messages into your
#    system and have them sent to you at predefined times.

#    in order for it to work, you must have at and cron installed on your
#    system.


echo "This command is used to enter a reminder message onto your screen"
echo
prompt  "Please enter time you would like reminder: "
read time
echo
echo "Please enter your reminder message - one line only"
read message
echo 'echo -n ''"'$message'" > ''`tty`''; /usr/cmd/beep >' `tty` | at $time
echo
echo "The following message:"
echo  $message
echo
echo "will be sent to you at "  $time
```

** *See the NEW products W-2 Link and 1099 Link* **

SOFTWARE FOR THE FORTUNE 32:16 AND FORMULA

NEW PRODUCT!! W-2 LINK ($495.00) - this product is designed for companies with over 250 employess who must report W-2 payroll information to the Social Security Administration (IRS) via magnetic tape or diskette. The product automatically takes information in the Business Accounting System (BAS) Payroll module and puts it onto diskette in the format required by the IRS. Available for PAC software also.

NEW PRODUCT!! 1099 LINK ($295.00) - this product produces Form 1099's required for submission to the IRS and is an extension to the standard BAS Payroll module.

DOS LINK ($495.00) - provides easy-to-use menus and programs that transfer data between any PC (including Laptops) running DOS and the Fortune system. Here are just a few examples of how this product can be used:

- Transfer Multiplan or Fortune:Word Documents from or to your DOS system.
- Transfer other ASCII or binary files.
- Use the PC or Laptop as a remote terminal on the Fortune system.
- Transfer THOROUGHBRED (SMC) BASIC data that has been extracted using one of our other products

Documentation and software for both the DOS and Fortune system are provided. Data may be transferred via a direct connection (9600 baud) between the two computers or via modem (at modem speed). You may also order additional hardware for your PC that may be required to accomodate the transfer.

If you are interested in purchasing a Laptop computer that can be used with this product for people on the road or at home, give us a call!! We also provide consulting services for interfacing your DOS system to the Fortune ($50/hr).

MULTIPLAN LINK - can pass data from BAS/IDOL/BASIC files to Multiplan and back. For example, it could be used to send data from any of the BAS files (Chart of Accounts, Customer Master, Inventory Master, etc). Allows selection of fields to be sent and provides logical retrieval and key range selection. $295.00

RECORDS PROCESSING LINK - can pass data from BAS/IDOL/BASIC files to Records Processing and back. For example, it could be used to take customer address data from the Customer Master and will automatically make the Records Processing List document for mail/merge. Allows selection of fields to be sent and provides logical retrieval and key range selection. $295.00

ASCII LINK - can pass data from BAS/IDOL/BASIC files to ASCII and back. Use to send data between machines or databases. For example, it could be used to send data from any of the BAS files (Customer Master, Inventory Master, etc). Allows selection of fields to be send and provides logical retrieval and key range selection. $295.00

TERMINAL/PRINTER LINK - provides capability to print data on a printer connected to a workstation. Can print BAS/IDOL/BASIC reports, Multiplan and Fortune:Word documents. BAS/IDOL reports can also be viewed before printing, converted to Fortune:Word format; accumulated for mass printing; backed up to diskette; printed on any system printer. An option to print BAS/IDOL reports on a LASER PRINTER is $50 extra. Many other options are included. $195.00 (Fortune:Word printing on a printer connected to a workstation is $50 extra)

TELEPHONE LINK - on line telephone message, telephone directory and inter-office mailbox/menu system. Telephone message entry resembles typical pink slip. Telephone directory can search by name or company. Mailbox/message system can send and receive Fortune:Word documents or Multiplan documents. Does much more! Give us a call. Written in Business Basic. $295.00

CALCULATION LINK - provides programs for amortization, depreciation, loan repayment, averaging (with graph), linear correlation and breakeven analysis. Written in Business Basic. $95.00

KOMPACT PERSONNEL ACCOUNTING - provides for data capture of personal, wage/salary, job, education, salary/wage, appraisal training, dental, medical and life insurance data. Includes over 80 pre-defined reports/worksheets. Written in Business Basic/IDOL. $995.00

WESTERN LINK - provides instruction on how to use your Fortune for sending/receiving telex messages using Western Union's Easylink. Can replace your telex! Fortune ITE or Handshake software required. $95.00

UNI-KOMP 9803 Rocktree Houston, Texas 77040 (713) 937-7782

listing like this and look at the first column to see which ones begin with a d - those are your directories.

However, this method is quite cumbersome if you have many, many files in your directory because it will mean that these files, directories and all, will flash by very quickly on your screen. This, of course, makes it very difficult to remember which files are directories.

A more useful method might be to use the **lf** command. This command is a special variant of the **ls** command and it lists all the files in your directory but it marks certain types of files in a special way. For example, it marks directories with a trailing / (slash). An example is shown here:

```
$ lf

.rolotemp
.tmi/
2yrsub.sn.Z
BALANCE.MP.Z*
BUDGET.MP*
abp/
```

This approach might be a little better than the former one since it will print out several columns across your terminal which means you may be able to see your whole directory on one screen. Then, your task would be to look through the list for all the files which end in a slash (e.g., `.tmi/` which is line two of the above printout).

However, there are a couple better ways of listing the subdirectories - and only the subdirectories - in your directory. The first way is by using the **find** command as follows:

```
$ find /u/mbp -type d -print
```

Ok, let's dissect this command. The first argument to the **find** command (`/u/mbp`) is the name of the directory in which you want to search. For this example I've chosen my home directory. The next argument is `-type` which is followed by a `d`. This instructs the **find** command to find only the files which are of type d where d stands for directory. Finally, the `-print` argument instructs the find command to list all of these files on your terminal. So, if I perform this command, I would see:

```
/u/mbp
/u/mbp/abp
/u/mbp/fred
/u/mbp/ralph
/u/mbp/ralph/jason
/u/mbp/reinhold
/u/mbp/church
/u/mbp/.tmi
/u/mbp/pies
```

Actually, this list shows me all the subdirectories in my account (e.g. abp, fred, ralph, reinhold and so on) but it also lists all the subdirectories in my subdirectories like `/u/mbp/ralph/jason`. So, although the **find** command does a pretty good job, it might give you more than you want to know. Similar in spirit to the find command would be the **dtree** command (which is found on our **Compressor's Delight Diskette**). This command would give you the same information as the **find** command discussed above, but it would also draw a little chart showing

We are the *largest* Authorized Fortune Master Dealer in the Pennsylvania and Delaware area. As part of the **P**rofessional **S**olutions **G**roup of companies with over 35 years in business automation, we have the *stability* that our customers demand

Our services include a factory trained field engineering team, a complete parts inventory including system upgrade capability and a software support department.

For all of your Fortune needs contact:

Mark Shields of PSG at
2550 Eisenhower Avenue
PO Box 762
Valley Forge, PA 19482

or you can call Mark at (215) 666-1400

the directory structure.

The last method will just print the subdirectories in your current directory. Remember the **ls -l** command? It prints out all the files in your directory and the lines that begin with a d are a directory. Well, we will use this information to cull through this potentially long list of files to get at just the ones that start with a d. To do this, we use the **grep** command as follows:

```
$ ls -l | grep '^d'
```

First we get the listing by performing an **ls -l**. We redirect the normal output from this command to the **grep** command. The `'^d'` tells the grep command to simply printout the lines that begin with a d. The single quotes are important, so don't forget them. The output from this command, on my /u/mbp directory would look like:

```
drwxrwxrwx 2 mbp      144 Aug 27 14:20 .tmi
drwxrwxrwx 2 mbp       96 Jun 26 14:43 abp
drwxrwxr-x 2 root     288 Oct 19 19:13 church
drwxrwxrwx 2 mbp       64 Jun 26 14:56 fred
drwxrwxr-x 2 mbp      144 Oct 20 15:14 pies
drwxrwxrwx 3 mbp       64 Oct 27 13:05 ralph
drwxrwxrwx 2 mbp       64 Jun 19 08:48 reinhold
```

And voila, all the subdirectories in my directory.

You may not have the **grep** command on your system (you probably won't unless you have Fortune's Development Utilities). However, we have a public domain program, called **bm**, which works like **grep**. We have packaged this goody on our **Compressor's Delight Diskette.**

Question: *We recently purchased a new Fortune 60MB Tape Expansion Cabinet and found that it was not compatible with the Fortune tape drive we already had, or the tape drive our software vendor has. The error message we get is "Far-file format error: empty Far-file". Have you heard of this problem, and more importantly, do you know of anyone who can help us since nobody locally can. We are even more concerned about the potential inability to restore backups in the event of tape drive replacement and/or catastrophic failure.*

Answer: This problem has only recently been brought to our attention. While there is no way of knowing if this is widespread, we do know of one company who states they have researched the problem and developed a hardware fix which can either make old drives compatible with new ones or vice-versa. They are DATACOMP Corporation, 300 N. Washington Street, Alexandria, Virginia 22314 - (703) 683-1181. Since the primary purpose of having the tape drive is for mass archiving and system backups, a drive incompatibility could create very serious problems and should be corrected.

Tri-Tech Computer Services, Inc.

UNBEATABLE COMPUTER SERVICE

DON'T SPEND ANOTHER DOLLAR UNTIL YOU'VE TALKED TO US!
(The Phone Call is Free)

1-800-622-TTCS

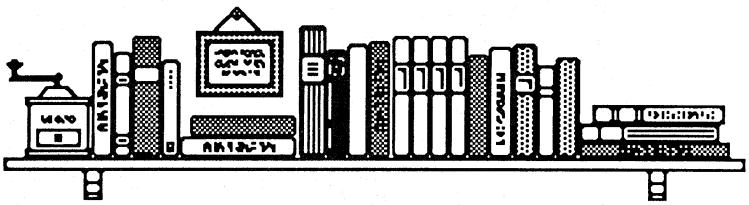
Unbeatable Specials:

Handshake Communications Software
Motherboard Upgrades (SX 2 MB & SX 3.5 MB)
70 MB Hard Disk Upgrade (Our Favorite!)
Additional RAM & Comm-A Boards

Connecticut:
106 Cynthia Drive
Kensington, CT 06037
(203) 828-9784

Massachusetts:
P.O. Box 1330
Brookline, MA 02146
(617) 965-4453

The BASIC Advisor



The BASIC Advisor is brought to us from Ray Wannall. Ray is President of BaSiC Software Corporation which is located in Baltimore, Maryland. This month Ray discusses a Concept Omega conference and the Fourth Generation Language that Concept Omega will introduce.

I promised, in an earlier edition, to report on Concept Omega's Thoroughbred Leader's Conference, and I take this opportunity to do so. Many of the 250 plus companies attending the Hilton Head, SC, affair on September 15-18 were past and present dealers of Fortune products. The occasion afforded me a chance to renew acquaintances from the 1986 conference as well as put faces with other names which have become familiar to readers of **/u/fortune *news*** over the years.

Those familiar included Joel and Naomi Levine of Interdiscipline Consultants, Inc., in Wyckoff, NJ, whom I inadvertently omitted from last year's conference report (honest, guys, I didn't know you subscribed!). The UNIX Wizard himself, Dave Kloes of Uni-Komp in Houston, TX, attended again this year along with his wife Sharon and faithful companion June Robinson (who informed me that it was one of the Uni-Komp "spouses" who won the golf tournament). We had a chance to chat with Steven Gray of E.C.S in New York, NY, one of Concept Omega's largest dealers. And a team from Generic Business Solutions, Inc., in Wilmington, DE, winner of this years "Dealer of the Year" award, shunned public transportation and braved the hazards of Interstate 95 to be a part of the activities. The magnificent seven were Ken Owens, Ken Lyons, Bruce McFarland, Larry Thorne, Tom Reale, Renny Kerr and Jack Speakman. They arrived late Tuesday evening, somewhat haggard but managing to display the relentless energy which earned them their top award.

In the "faces put to names" department, we were finally able to meet with **/u/fortune *news*** contributor Steve Rosenfeld of Superior Computer Systems, Inc., in Florissant, MO. Ben Alexander of Choice Technical Services in Marietta, GA, (a very pleasant gentleman) supervised one of the many Manufacturer's Hospitality Suites, and Jim Jones of Jones Business Automation Center in Charlotte, NC, introduced himself to me briefly at one of the evening activities. I had wanted to spend some time talking with Jim, but every time I was free it seemed he was not. (Maybe next year, Jim.) Earnest and Associates, Inc., of Millersville, MD, was represented by its three musketeers Tim Earnest, Stanley Mrazeck and Ken Kline.

The "obvious absentee" this year was Fortune Systems. We spoke with Elizabeth Muth as well as members of the Atlanta field office a week or two before the convention: we even sent them registration forms, but I guess we were too late. Fortune/SCI should be pleased to know, however, that one of their 32:16's was proudly displayed in the Software Fair by yours truly. It was borrowed from Kent LaCroy of Advertising Data Management Systems in Tucker, GA, who was not able to attend. Thanks again, Kent. We couldn't have done it without you.

We met literally hundreds of people, and I have probably missed a few. If I did forget you, please look me up at the 1988 conference, and be sure to tell me how much you enjoy **/u/fortune *news***.

The 1987 Thoroughbred Leaders Conference was a blast. The welcoming cocktail party and Low Country Beach Party were borderline revelry, and the International Theme Dinner was as elegant as it was palatable. The breakout sessions and software fair were informative and the keynote speeches, in general less dynamic than last year's crop, included a beauty by Fran Tarkenton, formerly of the Minnesota Vikings (I didn't know he owned a software company).

The big news from Concept Omega this year was the new Thoroughbred business computer and the Fourth Generation Language (4GL). The computer, called the TS-386, is manufactured by Wyse and features the new 32-bit 80386 CPU microprocessor chip operating at 16 megahertz. But we already have a great computer, so let's look at 4GL, which should be available for Fortune Systems within the next year or two.

4GL is the next logical step in the evolution of computer languages. The first generation was called "binary" code. It consisted of a series of ones and zeros:

```
01001101 11010001 01110110 10110001
```

Concept Omega compares this to "organizing the grains of sand in a cave". The second generation was "assembler":

```
f0 MOV a13ff0 1032eb
```

This, Concept Omega says, is "like scrubbing the bathroom with a toothbrush." Business BASIC is an example of a third generation language:

```
2010 IF Y$="YES" THEN GOSUB 7000
```

Concept Omega calls this "using brooms, mops, sponges, cleanser and buckets of water". With fourth generation we have "a housekeeper" doing the work.

A fourth generation language is more than a programming language in the classical sense. Much of the general housekeeping required by lower level languages is performed by 4GL. Some of the tools include an improved database management system, a dictionary-driven system, a screen developer, menu management system, report generator and, of course, a 4GL programming language. Concept Omega calls this Thoroughbred SCRIPT-IV, which we will discuss shortly.

The new Database Management System is called, are you ready for this, IDOL-IV. I will hasten to add that there has never been, nor will there ever be an IDOL-III. IDOL-IV is really the IDOL-II which has been on the market for about a year, and it is the cradle of the new 4GL. This past year Concept Omega has adjusted its Thoroughbred Business Accounting packages so that they operate under IDOL-II, but anyone wishing to integrate IDOL-II into present operations must either purchase a new set of applications or hire a programmer to make the necessary changes to existing programs. (Although we are assured our existing BAS or Thoroughbred products will be able to function under the new language, modifications will be needed.) I personally see no reason for end users to spend money in order to step half way into the fourth generation. Sometime in 1989 or 1990 there will be a new line of business applications written in 4GL. By then application development and modification will be significantly easier and less expensive.

Business BASIC is now evolving into SCRIPT-IV. We will soon say good-bye to maintenance programming, program exceptions, statement numbers and, I fear, artistic expression. The new language is condensed and easier to read:

```
PRINT SCREEN POCHECKS
CHANGE PODATA USING KEY RANGE FROM FIRST TO LAST
SELECT WHEN STATUS="OPEN" AND PURCHASE-AMOUNT>500
PROCESSING IS CLOSE-PURCHASE-ORDERS
```

It is "dictionary-driven" for easy maintenance. This means that if you need to change a definition, you do it once in a dictionary rather than in every program which uses the definition. SCRIPT-IV will be self-documenting and have its own maintenance utilities. Key access options include value, NEXT, FIRST, LAST and PREVIOUS and may be requested by RANGE with the options FROM, TO and ALL. Some of the other new commands include CHANGE, DO, COPY, and IF/THEN/ELSE/ENDIF. A line of SCRIPT-IV coding encompasses about ten lines of third generation Business BASIC. It's a new ball game, folks, but it still needs a technician. Maybe when we get to the fifth generation languages (i.e., voice activated) the end user will be able to define and modify his own applications. In the meantime, don't dismiss your programmer or software company.

In closing, I want to cover a few additional points of interest from the conference. First, if you are considering upgrading to the Thoroughbred 4GL, you should know that the programs require about 15% more storage than IDOL. Also, if you have purchased IDOL-II you may upgrade to IDOL-IV at no additional cost. And finally, for those interested, Concept Omega, looking for some "rapid growth" money, is considering an initial public offering. Their 1987 goal of $8.4 million has been exceeded, and they expect to bring in as much as $11 million by the end of the year. The majority of their sales have been Business BASIC and the TBOS operating system.

Next month we will return to our question/answer format. I have several questions in stock, but I always welcome your input. Call me at (301) 448-9460 or write to 5201 Powhatan Street, Baltimore, MD, 21207. I am especially interested in hearing from BASIC/IDOL/BAS support companies for our Third Annual Support Companies listing (has it been three years already?), and I'll be making my official request in a couple of months. I plan to publish the list in the January, 1988, issue of **/u/fortune *news***.

Expand your Fortune® without spending a fortune.

New equipment costs big bucks. The Good Fortune Trader sells fully tested burned-in equipment, refurbished to Fortune® standards at prices that won't make waves with your cash flow. Call or write today for a listing of our specials.

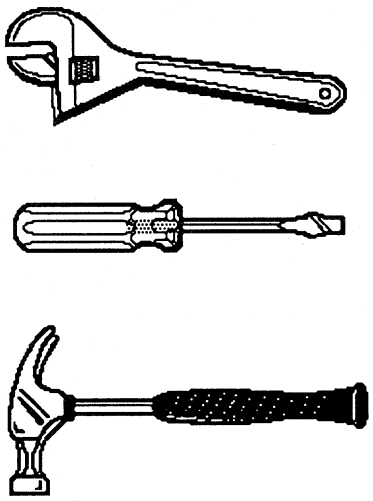
THE GOOD FORTUNE TRADER

60 DAY WARRANTY

a division of Computer Services of Atlanta

(404) 447-4151

5825 Live Oak Parkway, Suite 2-C, Norcross, Georgia 30093



Fortune/SCI's Development Utilities:

Tools for UNIX

In past articles, we have often alluded to a Fortune/SCI product known as "Development Utilities." In most cases, we have glossed over just what these utilities are and why you might want them. In the first part of this article, we will list and breifly describe what programs are included in the Development Utilities. In the second part, we will discuss 3 of the utilities in greater detail. In this section, we will also provide some C programs that mimic the basic purpose of the program so that you can use them if you need to. In future articles, we will discuss some of the other programs and present some C and/or shellscript equivalents. For example, several of the programs included on the Development Utilities are associated with what is known as SCCS (Source Code Consistency System). A future article will describe what this system is and how to use it.

List of Development Utilities

vi - Screen oriented text editor.

nroff - Text formatter.

col - Filter used with multicolumn output, converts whitespace to tabs to shorten printing time.

ex - A display based editor.

expand - Pre-processor for character files that contain tabs, expands tabs to spaces. **Unexpand** puts tabs back into data.

fold - Filter which folds the contents of a file, breaking lines to have a maximum width.

num - Lines in a file are preceded by line numbers.

ul - Does underlining. Reads a file and translates underscores for a particular terminal.

SCCS - Source Code Control System

admin - Create SCCS files and makes changes to the parameters of existing SCCS files.

cdc - Changes the commentary of an existing delta.

comb - combines two or more consecutive deltas into one delta that takes up less space.

delta - Creates new versions (deltas) of the text of existing SCCS files.

get - retrieves versions of SCCS files from the disk.

help - Prints information to explain a message from an SCCS command or explain the use of a command.

prs - Prints all or part of an SCCS file on the screen, according to your specifications.

rmdel - Removes a specified delta from an SCCS file.

sccsdiff - Indicates the differences between any two deltas of an SCCS file.

val - Determines if an SCCS file meets the characteristics specified by the val options you select.

vc - Controls the version that input is copied to output by arguments and control statements.

what - Searches an SCCS file for all occurrences of a particular pattern.

File Comparisons

bdiff - Calls diff and lists differences between large files.

comm - Lists lines common to two sorted files.

ctags - Makes a tags file for ex from the

specified C, Pascal, and Fortran sources.

diff - Compares two version of a file, or two directories.

diff3 - Compares three versions of a file, and publishes disagreeing ranges of text.

file - Determines file type.

grep - Searches a file for a pattern.

head - Prints the first 10 lines of each of specified files.

sed - A stream editor, edits according to a script of commands.

split - Reads a file and writes it in as many pieces as necessary onto a set of output files.

strings - Finds the ASCII strings in a binary file.

sum - Calculates and prints a 16-bit checksum for the named file and prints the number of blocks in the file.

tr - Copies the standard input to the standard output with translation of selected characters.

tsort - Produces an ordered list of paired items from a given file or from standard input.

uniq - Compares adjacent lines in a file and removes repeated lines.

wc - Counts lines, words, and characters in named files.

Other Programs

bc - An arbitrary precision arithmetic language that assembles C.

cal - Prints a calendar for a specified year. If a month (1-12) is also specified, a calendar for that month only is printed.

chlog - Changes the style in which operating system error messages are printed.

chowngrp - Changes both the owner and group identification of files.

clear - clears your terminal screen if possible.

clri - Writes zeros on the i-nodes with the decimal i-numbers of a given filesystem. Used to repair filesystem damage.

compact - Compresses and uncompresses files.

crypt - Using a password, files are encrypted for security purposes.

csh - A shell (command-interpreter) with C-like syntax.

dc - A desk calculator.

dcheck - Checks filesystem directory consistency.

del - Delete files which confirmation.

expr - Evaluates arguments as an expression and writes the result to standard output.

format - A menu-oriented disk formatter.

gets - Reads a string from standard input.

icheck - Examines a filesystem for inode consistency.

ln - Creates links, which are directory entries referring to files.

makekey - Generates an encryption key, used by the crypt command.

man - prints out manual pages.

ncheck - Generates a pathname and i-number list of all files on a set of default filesystems.

nice - runs commands at a low scheduling priority.

nohup - Runs a command immune to hangups and kills.

od - Dumps a file in one or more formats, octal being the default.

printenv - Prints out the values of the variables in the environment.

reconf - Reconfigures your system for various peripheral devices.

reset - Resets the teletype characteristics to a sensible state, useful after a program dies.

su - Sets a user id temporarily. Asks for the password, changes to that user id, and then invokes the shell. CTRL-D returns ownership to the original user id.

tabs - Sets the tabs on a variety of terminals.

tail - Copies the end of a file to standard output, beginning at a designated place.

tar - Saves and restores files on the flexible disk. Keys are used to specify which files are to be dumped or restored.

time - Prints the time elapsed during a command, the time spent in the system, and the time spent in execution of the command.

touch - Updates the modified time of a file or creates the file if it doesn't exist.

tset - Sets terminal modes.

users - Prints a list of login names of all users currently on the system.

which - Locates a program file including aliases and paths.

whoami - Prints the user id of the person currently logged in at the terminal.

yes - Repeatedly outputs "y".

The next three commands that we will discuss have at least one thing in common. They all work on the contents of a file. Fold is useful for wrapping lines, head is useful for seeing the beginning of a file and tail is useful for viewing the end of a file. In the following sections we describe these programs and we

TRADE - UPGRADE - EXPAND
Your FORTUNE SYSTEM

Processors:
- *Fortune 32:16 SX*
- *Fortune Formula 4000*
- *Fortune Formula 8000*
- *Fortune Multiplier (Run multi-user For:Pro & Fortune applications on a PC)*

Hard Disks:
- *30MB*
- *45MB*
- *70MB*
- *85MB*
- *145MB*

Tape Drives:
- *20MB External*
- *60MB External*
- *60MB Internal*

Memory:
- *256KB*
- *512KB*
- *1024KB*
- *4096KB*

Controllers:
- *Comm-A 4 Ports*
- *Comm-A 6 Ports*
- *Bit-Mapped Graphics Controller*
- *Fortune:Link Controller*

BONUS - BONUS - BONUS

All Fortune Licensed Software Upgrades and/or Replacements
At No Additional Charge with Processor Upgrade/Trade!

We can perform these trades and/or upgrades within 24 hours of receiving your system. In fact, if you ship your system overnight on a Friday for Saturday delivery, we will have it back to you on Monday morning, all ready to plug in and run; not losing the use of your system for a single work day. And, we will issue you a full 90-day warranty on the entire system.

JFM provides services to Fortune Systems owners nationwide either direct or through our authorized Affiliated Dealers. If you haven't used our services, I invite you to discover JFM; you will be glad you did.

Sincerely,

Morris Frydman
President

Talk to us, then talk to our clients

JFM
BUSINESS
SYSTEMS
The Merchandise Mart
Lobby Suite 144
Chicago, Illinois 60654
Call (312) 644-0870

Authorized Fortune/SCI Systems Master Dealer

present a C program for both fold and head.

Fold

The first, **fold** does one thing. It "folds" lines in a file to a certain length. Suppose for a moment that you have a file that has lines in it longer than 80 characters. Suppose also that you are working on a terminal that does not "wrap" long lines. ("Wrap" means to break the line at 80 columns and then display the remainder on the next line). Well, if you were in this dilemma, you would need the services of **fold**

Fold is easy to use. The directions for fold are as follows:

```
fold [-width] [file...]
```

For example, suppose I had a file called **junk** and its contents were as follows:

```
This is a junk file.
Which contains nothing but
junk.
```

And now suppose I wanted to fold the lines so that each line was 10 characters wide. I would issue the following command:

```
$ fold -10 junk
```

```
This is a
junk file.
Which cont
ains nothi
ng but jun
k.
```

So, you see, **fold** simply reads in the file and then prints it out chopping off each line at precisely the length that you specify. If you type:

```
$ fold junk
```

fold will default to 80 characters.

There is no easy way to program **fold** in the shell language. However, we present a relatively straighforward C program which does what **fold** does (see Figure 1). If you have a c compiler on your computer, you could type this program in a file (using an editor like **ed**, **vi**, or **sc**). Once in a file, called **fold.c** you could compile it by typing:

```
$ cc -o fold fold.c
```

Head and Tail

The next two programs we discuss are called **head** and **tail**

Head prints out the first n lines of a file. It is sometimes useful for viewing the beginning of a file without having to see the whole thing. For example:

```
$ head -24 junk.dat
```

will print out the first 24 lines of the file called **junk.dat** The form for the **head** command is very similar to the fold command:

```
head [-count] [files...]
```

If you don't specify a count, **head** will default to 10. Thus, if you type:

```
$ head junk.dat
```

Figure 1- fold.c

```
#include <stdio.h>
main(argc, argv)
char *argv[];
{
    char *cp;
    int fd,i,target,back;
    FILE *fp;
    target = 80;

    while (--argc && ((*++argv)[0] == '-' || (*argv)[0] == '+'))
    {
          for(i=1;(*argv)[i] != '\0';i++)
                    if (!(isdigit((*argv)[i])))
                              usage("Bad number for fold.");
          sscanf(*argv,"%d",&target);
          if (target < 0)
                    target = target * -1;
    }
    fp = stdin;
    do {
        if (argc > 0 && (fp = fopen(*argv, "r")) == NULL) {
           fprintf(stderr, "fold: can't open %s\n", *argv);
           argv++;
           continue;
        }
        fold(fp,target);
        fclose(fp);
        argv++;
    } while (--argc > 0);
}

usage(string)   char string[]; {
    fprintf(stderr, "%s\n",string);
    exit(1);    /* replace X with suitable integer */
}

fold(fp,target)
FILE *fp;
int target;
{
          char c;
          int count=0;
          while ( (c = getc(fp)) != EOF)
          {
                    if (c != '\n') {
                              putchar(c);
                              count++;
                    }
                    else {
                              if ((count < target) && (count != 0)) {
                                        putchar(c);
                                        count = 0;
                              }
                    }
                    if (count == target) {
                              putchar('\n');
                              count = 0;
                    }
          }
}
```

Figure 2 - head.c

```
#include <stdio.h>
main(argc, argv)
char *argv[];
{
    char *cp;
    int i,target,back;
    FILE *fp;

    target = 10;
    while (--argc && ((*++argv)[0] == '-' || (*argv)[0] == '+')) {
          for(i=1;(*argv)[i] != '\0';i++)
                    if (!(isdigit((*argv)[i])))
                              usage("Bad number for head.");
          sscanf(*argv,"%d",&target);
          if (target < 0)
                    target = target * -1;
    }
    fp = stdin;
    do {
        if (argc > 0 && (fp = fopen(*argv, "r")) == NULL) {
           fprintf(stderr, "head: can't open %s\n", *argv);
           argv++;
           continue;
        }
        head(fp,target);
        fclose(fp);
        argv++;
    } while (--argc > 0);
}

usage(string)   char string[]; {
    fprintf(stderr, "%s\n",string);
    exit(1);    /* replace X with suitable integer */
}

head(fp,target)
FILE *fp;
int target;
{
          char line[256];
          int count=0;
          while ( ((fgets(line,256,fp)) != NULL) && (count++ < target))
                    printf("%s", line);
}
```

the computer will respond with the first 10 lines of the file junk.dat.

Tail turns out to be a slightly more complicated command. Its form, as found in any UNIX manual page, is thus:

```
tail  +-number  [lbc]  [rf]
[file]
```

Let's begin with the basic function of tail. Its main purpose is to print out the "tail-end" of a file. Thus, if you type:

```
$ tail -20 junk.dat
```

the computer will respond with the last 20 lines of the file **junk.dat** One common use of **tail** is to look at the ends of log files in UNIX. For example, some UNIX programs keep a file that keeps track of when the program ran and whether it was successful or not. Tail is great program for looking at the end of such a file in order to see how things are going.

In the above example, the **-20** instructs tail to read backwards from the end of

UNISOFT INC.

Experts in Computer Consulting - Sales - Service

Presents

FORTUNE SYSTEMS HELP

We Provide:

- Expert Trouble-shooting
- Years of programming and accounting experience
- *Free Initial Consultation*
- Training Available for:
  - * BASIC/IDOL/BAS and Multiplan
  - * UNIX and Ext. Fortune:Word
- Hardware Repairs & Upgrades
- FORTUNE trained experts at a cost that will surprise you!
- *Trade-Ins Welcome*
- Specializing in:
  - * Hard Disk Trade-In
  - * Hard Disk Upgrades

Authorized Fortune Systems Repair Depot
Fortune Systems National Value Added Reseller (VAR)
CALL: (303) 671-8600 EMERGENCY HELP: (303) 649-8505

UNISOFT INC. 14100 East Jewell Avenue Suite 27 Aurora, Colorado 80012

the file for 20 lines and then start printing out the file. An alternative method exists for getting at the end of the file. If you type:

```
$ tail +20 junk.dat
```

the **tail** program skips the first 19 lines and then starts printing from there to the end of the file. This particular use of **tail** can be modeled with the **more** command as in:

```
more +20 junk.dat
```

In its default mode, **tail** works on lines. When we specify **+20**, for instance, **tail** skips the first 20 lines and then starts printing. The `[lbc]` flags listed above can modify this, however. If we type:

```
$ tail +20c junk.dat
```

tail will skip the first 20 *characters* and then start printing. The **b** option, as in `tail +20b junk.dat`, will cause **tail** to skip the first 20 *blocks* of data (blocks on the Fortune are generally equivalent to 1024 characters).

Finally, we need to describe the **r** and **f** options. The **r** option instructs **tail** to print in *reverse* order. Perhaps the best way to understand this is to see an example. Suppose our file **junk.dat** looked like this:

```
1 - first line
2 - second line
3 - third line
4 - fourth line
5 - fifth line
6 - sixth line
7 - seventh line
8 - eigth line
```

Now, if we type:

```
$ tail -4 junk.dat
```

we would see

```
5 - fifth line
6 - sixth line
7 - seventh line
8 - eighth line
```

In other words, the last 4 lines of the file. However, if we typed:

```
$ tail -4r junk.dat
```

we would see:

```
8 - eighth line
7 - seventh line
6 - sixth line
5 - fifth line
```

Which is the last four lines, but in reverse order.

Finally, the **f** option causes **tail** not to quit at the end of the file, but rather wait and try to read repeatedly in the hopes that the file will grow. This would be useful for monitoring the progress of some UNIX program that was writing a file. By typing:

```
tail -f filename
```

the tail command would print out the end of the file. However, if the file grew any larger, it would keep printing this out as well.

Conclusion

In this article, we listed and briefly described the commands included in Fortune's **Development Utilities**. These are UNIX programs and utilities that are found on almost all implementations of UNIX. As a consequence, many other UNIX utilities (especially shellscripts) will use these commands. Thus, in order to have a smoothly functioning UNIX system, many of these commands are necessary.

Also, many of them just make life in the UNIX shell much easier. Some of the commands, like **head**, **tail** and **fold** make viewing files less time consuming and/or troublesome.

In addition to describing the programs briefly, we also discussed several in depth. We also presented some C programs to mimic the behavior of two of them. In the future, we will introduce a software diskette with these and other utilities included. □

Mark Palmerino

DO YOU KNOW...

there is an organization that has a thorough knowledge of

FORTUNE BAS
MULTIPLAN IDOL

We have been providing widespread support to our Fortune clients for more than 4 years.

If you need:

- training
- support
- maximum computer performance
- peace of mind relating to processing
- customization
- application results
- minimum errors and confusion
- better ideas on how to increase the power of your Fortune

Call Steve Rosenfeld at (314) 731-3636

SUPERIOR COMPUTER SYSTEMS, INC.
P.O. Drawer T
Florissant, MO 63031
Making computers work for you

Hints

Continued from page 1

menu and from there choose the **Index Generator** selection. After you have done that, you will be asked for the **Input Document**, the **Word List Document**, and the **Output Document**. The Input Document is your original document. The Word List Document is the document we have described above, and the Output Document is the name you choose for the actual Index.

Once all of these names have been entered, the computer goes to work and assembles the index. A sample Index follows:

```
INDEX

append
      1, 2, 5
editor
      1, 2, 3, 4, 5, 6
insert
      1, 2, 3, 4, 5
screen
      1, 2, 3, 6
Section
      1, 2, 3, 4, 5, 6
```

Note that the index words have been sorted into alphabetical order, and each page that the word appears on is listed. Because **indents** have been placed at the beginning of each line with page numbers, if there is more than one line full of page references, subsequent lines will also be indented.

If you don't like the format of the Index, it is possible to write a glossary entry to rearrange it. We wanted our Index to look like this:

```
INDEX

append.................1, 2, 5
editor.................1, 2, 3, 4, 5, 6
insert.................1, 2, 3, 4, 5
screen.................1, 2, 3, 6
Section................1, 2, 3, 4, 5, 6
```

To do this, we wrote the following glossary entry:

```
entry a
{
    call display(false)
    goto "b"
    while(!globerr)
    {
        search return indent execute
cancel
        delete goto indent execute
        insert
        while ((position) != 30)
        {
            "."
        }
        execute
    }
    call display(true)
}
```

This entry does three things. First, the "`goto b`" line moves upto the top of the file. Then it repeatedly searches for the `return indent` sequence that is found at the end of each line before the page numbers begin. Once it finds that sequence, it deletes the return and one of the indents. Finally, it inserts periods (.) until it gets to position 30 on the screen. In order for it to work correctly you will want to create a format line that has its first tab at position 30.

If you want to modify the entry, you can easily change the position where the numbers will start by changing the `30` to someother number. If you'd rather not have periods, you can obviously substitute a different character between the quotation marks.

Creating the Word List Automatically

In the August 1987 volume of **/u/fortune *news*** we discussed a glossary entry that automatically copied a selected word to the workpage of your document. We won't bother explaining the entry again here, but we will include it for your reference. It's really quite simple. Take a look at Figure 1.

```
entry b
{
        title = word    /*read the word at the cursor location*/
        command note    /*mark the cursor position in the document*/
        goto "w"        /*go to the workpage*/
        goto down       /*go to the bottom of the workpage*/
        call feed(title)        /*insert the word saved in thevariable*/
        goto note       /*return to the bookmark cursor position*/
}
```

Figure 1

Once you have accumulated your words, you will want to use the `SHIFT MOVE` or `SHIFT COPY` feature of **Fortune:Word** to copy the words to your wordlist document. If you would prefer, you may be able to write an entry that would open up a window for your wordlist document, and then copy the words directly into that window. This might be a better approach if you planned to add many words to the document.

Some special notes on Word Lists

If you look in the **Fortune:Word Reference Guide** you will find that there are some special situations to be aware of when creating word list documents. As was stated above, it is acceptable to use either words or phrases in this document. The special situations occur when you use both, and the words are the beginning part of the phrase.

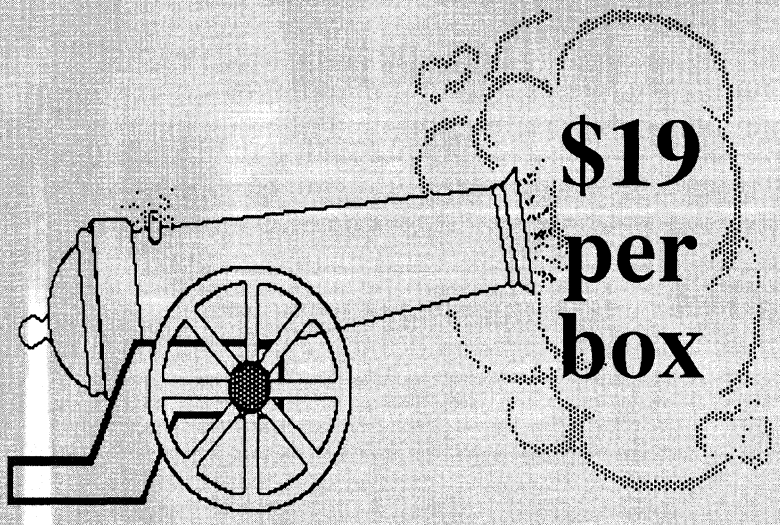
For instance, if your list resembles either of these lists, you may get unexpected results:

```
Fortune:Word
Fortune:Word Glossaries
Fortune:Word Menu
Fortune

Fortune
Fortune:Word
Fortune:Word Glossaries
Fortune:Word Menu
```

We suggest you check the manual and experiment on a test file to determine the outcome.

Josh Lobel



An Explosively Good Deal on Computer Paper

Recently, we found a *very* good deal on computer paper. We have arranged with a paper supplier to supply us with **20lb., 9.5 by 11 inch micro-perf** computer paper at the very competitive price of $19 per carton. Each box contains **2500** sheets but to keep these prices you must order a *minimum of ten cartons* and you must include payment with your order. We can also deliver five cartons for $24. See the "**Wow - What a Deal**" order card included in this issue for more information. *(This ad supersedes all previous offers).*

And Don't Forget Our Software Diskettes...

Please see the order card for prices. What follows is a brief description of the diskettes. These descriptions are not complete, but are meant to whet your appetite

Compressor's Delight

Utilities include:
dtree - produces a graphic picture of your directories and files.
grep - searches files for a string
compress - reduces sizes of stored files
menu.sh - shell script tool for UNIX novices
clock1 - another digital clock
gothic - create signs in Gothic font
loan - program to calculate loan payments
Games include:
hack - classic game of a journey through the past
cribbage - play the computer (Good Luck!)
craps - If you have to ask, you can't afford it

Fortune:Word Tutorial

This disk is useful for Fortune:Word beginners. It is a demonstration script with sample files which will guide you quickly through Fortune:Word's features. Includes Record Processing.

Perlman's Unix|Stat

This disk includes many programs for performing both descriptive and inferential statistics. Includes programs for multiple regression, ANOVA (betweeen and repeated measures designs suported) and time series analysis.

Fortune Utilities

Utilities include:
at and cron - execute commands at preset times
sc - a powerful and easy to use screen editor
hexed - like sc, but works on binary files
SWITCH - change baud on terminals temporarily
function keys - programs your function keys in UNIX
Games include:
banner - the standard banner produces signs
blob - probably the best action game
breakout - like the arcades, but slow
fortune - pearls of wisdom at a touch of a key

Ckermit

This is a later version of kermit which incorporates many new features. It is a very useful file transfer program. The new version has extensive on-line help, auto dials the modem, acts as a file server and more.

Unix Tools

Utilities include:
rolo - an Electronic Rolodex, a real favorite of ours
bm - a faster version of grep
cut and paste - extracts information from files
monitor - repeatedly run a command
Games include:
eliza - a computer psychiatrist
puzzle - move the blocks into the right order
block - types huge letters on your terminal
hack - a re-released version, better than previous one on Compressor's Delight

D.C. Grab Bag

Utilities include:
kermit/kmenu - a communications program
addglobal - add your programs to Global Menu
FTfix - restore your terminal when it fills with garbage
clock - puts large clock on your terminal
Games include:
backgammon - popular board game
biorhythm - if you like horoscopes, you'll like this
othello - another popular board game - the computer is ruthless
mastermind - guess what the computer has chosen

Zip Tease Delight

This disk contains files that cross reference phone number exchanges with zip codes. We have included some shellscripts that access this data. Also included is a recent program that appeared in UNIX/World magazine: telltime. Enter an area code and the program tells you the current time in that area. Never look up an area code or zip code again.

Editors Choice Diskette

This disk contains five editors. Included are Ted, e17 (Rand editor), and sc (also known as screen). These editors each have different strengths and are useful for editing ASCII files. Hexed is a special "sc" like editor that is built to edit binary files with ease. Also included is dired which is an editor that is useful for working with directories in UNIX. Some of these editors support windowing and all of them are essential for the well stocked Fortune Computer.

PROGRAMMERS

The leading manufacturer of status display panels for the fire alarm and security industry is seeking programmers and junior programmers to develop new applications and refine existing programs.

Successful candidates will be experienced in IDOL/BAS and SMC Business Basic for Fortune and Basic 4 systems. Familiarity with UNIX and system set-up and maintenance is desirable.

Candidates interested in joining a growing team-oriented organization located in Falls Church, Virginia should write or call with resume and salary history, to:

WSA
308 Hillwood Avenue
Falls Church, Virginia 22046
703 241-0687

FORTUNE SYSTEMS
an SCI Systems Company

No matter what your product or service, staying competitive requires control

Let QBC, your company's one source computer supplier, provide your business with solutions to:

Order Entry	Payroll
Accounts Payable	General Ledger
Accounts Receivable	Sales Analysis
Invoicing	Legal Time Billing
Inventory Control	Manufacturing Software
Office Automation	(MRP II Plus)

QBC is your one source for hardware, software, maintenance, customer training and data conversion.

MASTER DISTRIBUTOR FOR UPSTATE NEW YORK AND ERIE, PENNSYLVANIA

QBC

QBC Systems Inc.
375 NORTH FRENCH ROAD, AMHERST, NEW YORK 14120
(716) 691-5201 1-800-443-5770

/u/fortune news
73 Jason Street
Arlington, MA 02174

Address Correction Requested

/u/fortune* news

The Newsletter for Users of Fortune/SCI Computers — November 1987/Volume 4 Number 11

Software and Hardware Developments

Interview with Ron Borelli

Ronald Borelli is Vice President of SCI's Computer Division. As such, he is the top man with regard to Fortune/SCI strategy. Mr. Borelli graciously accepted our invitation to interview him about the current state of Fortune/SCI as well as future plans. The following interview took place in Huntsville, AL over lunch.

/u/fortune news: What is SCI? Who is SCI? What does SCI stand for?

Mr. Borelli: I'll tell you a little bit about SCI. The company was founded in 1961 by our current chairman, Olin King. SCI, then known as Space Craft Incorporated started out as a subcontractor to NASA in the Apollo Saturn space programs. Later, the company expanded into military programs for the Department of Defense. This activity which continues today forms the extensive technology base of SCI.

In 1971, management decided to expand into the commercial business and determined that the best way to do that was with high-volume, low-cost manufacturing.

SCI is considered today to be the largest subcontractor manufacturer in the world. We have facilities in Alabama, North Carolina, Scotland, Singapore, New York, South Dakota, Northern California and Colorado Springs. We recently opened up a joint manufacturing facility with a Mexican affiliate in Guadalajara, Mexico and plan expansion in the Boston area in the next year.

SCI is considered today to be the largest subcontractor manufacturer in the world.

/u/fortune news: In Boston? Where in Boston?

Mr. Borelli: We are looking at some sites right now - mainly for manufacturing with some board design.

The computer Division, which I manage, was started in the late '70's with the purpose of building on the technology base of the Government Division and SCI's manufacturing capability.

During the past five years, we have been mainly involved in designing and manufacturing products and systems for major customers. At the same time we have been developing

See **Borelli**, Page 17

Using Multiplan SYLK Files

From time to time readers have called us to ask if it is possible to import data into a Multiplan spreadsheet. They had numbers from the stock market, or financial packages or whatever, and they didn't want to bother rekeying them into their spreadsheet. We have said that although it is not as easy as it is on some other spreadsheet programs, that it is possible, using a variety of methods.

One method that we discovered some time ago is to use the UNIX redirection trick. Normally Multiplan expects you to enter information by typing it in from the keyboard. By using UNIX to redirect the input from the keyboard to a

See **Multiplan**, Page 12

In This Issue...

Interview with Ron Borelli...**Page 1**.

Multiplan SYLK files...**Page 1**.

Editorial, Help us help you...**Page 2**.

Fortune:Word Windows...**Page 3**.

Locating and then Sparing Bad Blocks...**Page 5**.

System Administration, Part 7, dealing with groups...**Page 8**.

The BASIC Advisor - extending record sizes...**Page 21**.

New Software Disk with a spelling checker...**Page 15**.

*/u/fortune *news* is published by The Cambridge Consortium, Inc. It is not affiliated with Fortune Systems/SCI Technology, Inc..

/u/fortune *news*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz

Technical Editor
Scott Bradner

Advertisting Manager
Mark Palmerino

/u/fortune *news* is published 12 times per year by The Cambridge Consortium, Inc. Neither **/u/fortune** *news* nor The Cambridge Consortium are affiliated with Fortune Systems/SCI Technology, Inc.

Subscriptions cost $48 for one year within the U.S.A. For foreign subscriptions, add $27 each year for added postage and handling costs.

Back issues are offered at:

Vol 1, Numbers 1 - 3	$1/each
Vol 1, Number 4	$.50/each
Vol 1, Numbers 5 - 8	$3/each
Vol 2, Numbers 1 - 7	$3/each
Vol 3, Numbers 1 - 12	$4/each
Vol 4, Numbers 1 -	$4/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $10 each. Please add $5 per order for foreign destinations.

Although every attempt is made to insure that our contents are error-free and without risk, The Cambridge Consortium is not responsible for losses that occur due to correct or incorrect interpretation of **/u/fortune** *news*.

For more information, call or write:

/u/fortune *news*
73 Jason Street
Arlington, MA 02174
(617) 663-7700, (617) 646-5603

The text for this newsletter was prepared using a Fortune 32:16 for text input. The information was then transferred to an Apple Macintosh Plus with kermit and formatted for typesetting using the Aldus Pagemaker program.

UNIX™ is a trademark of Bell Laboratories, Inc. Fortune 32:16™, Fortune:Word™ and FOR:PRO™ are trademarks of Fortune Systems/SCI Technology, Inc. Multiplan™ and MS-DOS™ are trademarks of Microsoft. IBM-PC™ is a trademark of IBM. Macintosh Plus™ is a trademark of Apple Computer. Pagemaker™ is a trademark of Aldus Corporation.

©1987 by the Cambridge Consortium, Inc.
11/87

From the Editors...

Helping You

We enjoy the Fortune/SCI hardware and software. If we didn't, we wouldn't have started **/u/fortune** *news*. We also enjoy producing **/u/fortune** *news* for another basic reason - we feel good about helping *you* use your Fortune/SCI equipment more effectively. We hope that each month brings you additional information that is helpful.

But this is ***NOT*** a one way street. Perhaps you have not thought about it in this way, but we need *your* help in order to help you. Now, one obvious way we need your help is through support of **/u/fortune** *news* via its subscription price. If you receive this magazine, we are assuming you have done your part in this way.

There is, however, another way in which you can help us help you. That way is through *feedback*.

We need feedback from our readers in many areas. For example, each month we put together another issue of **/u/fortune** *news*. We must decide what kind of articles to put into the magazine. What is the proper *mix* of Fortune:Word to Multiplan to Business BASIC to Basic UNIX to more advanced UNIX? Should the balance tip toward Fortune:Word or toward UNIX? What is the *most* helpful to our readers? We wrestle with these questions each month.

So, What Can You Do?

You can help us make better decisions by giving us feedback on the specific articles we publish. Tell us the good and the bad. What was helpful and what wasn't - in ***your*** opinion.

On the one hand, we know we are not doing too badly because many of our subscribers decide to renew their subscriptions each year. However, we are somewhat saddened when we find out that a former subscriber decided not to renew because the magazine wasn't meeting his or her needs. We are sorry about that, but part of the blame must rest with them, *if* they have not expressed what they needed from us.

The purpose of this editorial is to spur YOU to write to us. We would be happy to get some sort of response from *every* one of our subscribers.

But you ask, "What can I say?" Well, here are some ideas. In this particular issue, there are several articles. Did you like them or dislike them? As we asked above, what was helpful and what was not helpful? You might have an idea for an article or an area that you would like to see addressed - write us.

You can also help others by sending in problems that you've encountered with your Fortune. Maybe you have even figured out the solution. But perhaps there are scores of other subscribers who could benefit from YOUR experience. Whether it is a problem you don't know the answer to or one which you *do* know the answer to - send us a description so that we might help others.

Also, we know that many of our subscribers are experts in their own right. As you share that expertise with us, you will benefit many others. We know that there are individuals out there who have worked up specialized printcap entries or termcap entries - why not send them to us. If we get enough we could put them together on a diskette. What about public domain software. Do you have any goodies running on your system that might be helpful to other Fortune owners? Send them to us. Perhaps you have discovered a commercial program that *really* meets your needs. Or maybe you were unlucky enough to spend your money on a dog. Let the world know.

You might also consider contributing a short article about some feature of Fortune:Word or UNIX or whatever you have wrestled with and conquered. We can almost guarantee that your experience will be helpful to many others.

So, become involved. Write to us and give us your opinions, suggestions and ideas. We know that if you do your part, you will also see **/u/fortune** *news* become even more helpful to you.

Using Fortune:Word Windows

In the past we have written about the wonders of **Fortune:Windows**, a software package that lets you flip between several applications on a single terminal. We use this package all of the time, and find it very helpful in our daily work. However, many readers don't own Fortune:Windows, and may not realize that they can do some windowing within Fortune:Word. In fact, the windows feature of Fortune:Word is more appropriate for certain word processing tasks than Fortune:Windows would be.

The Fortune:Word windows function allows a user to open as many as four windows on the terminal screen. The windows can be filled with four different documents, or different sections of the same document. Each file can be edited separately and text can be easily copied or moved between documents. The windows feature is useful any time you are working on multiple, related documents.

In order to open a window, you simply press the **COMMAND** key (**F13**) followed by the letter **w**. A bright square will appear along the edges of the screen and the message **Set Window** will appear. You can move the cursor along the top or side of the screen to set the point where the split should occur. If you split the screen horizontally, you will be able to see the full width of a document. A vertical split will yield a partial view of your file, unless the format line is shorter than the window. If this doesn't make sense as you read it, it will once you try it. For most purposes, a horizontal split will probably be more useful. This will result in a screen that looks like Figure 1.

Once you have selected a split point, you will be prompted for a document name. You can enter the name of any document that exists, including the one you are working on. Once you hit **EXECUTE**, that document will appear in the window. **IMPORTANT NOTE:** If you have two copies of the same document in different windows, the last one that is exited will be the version that is saved. For this reason it is important that you exit from the documents in the order you desire.

Working with Windows

There are two other **COMMAND** key combinations that are useful for working with windows, **COMMAND j** (Jump) and COMMAND **f** (Flip). The jump command changes the active window, or editing window, which is indicated by a bright frame. The flip command flips between windows, erasing all windows except for the active window. If you want to close a window, you **CANCEL** out of the document as you normally would, and make the choice for saving the changes. You do not have the option to print directly from the **END OF EDIT** options when you're working with Fortune:Word Windows.

Copy or Moving Text Between Documents

To copy or move text between documents, it is not necessary to use the **SHIFT COPY** or **SHIFT MOVE** function. All that is necessary is to highlight the text to be copied or moved as usual. When the system says "To Where?" you use the **COMMAND j** command to jump to the other window. Finally you choose where you want the text to go, and press **EXECUTE**. Note that in addition to normal scrolling with the cursor keys you can also use **GO TO PAGE**, **SEARCH**, **NEXT SCRN**, etc. to move through either document.

Josh Lobel

Figure 1

```
Doc multisylk    Page 1    Line 6    Pos 1
word Format 1 Spacing 1 Length 54          ■ 1024
1(1 .... 1... .... 2... .... 3... .... 4... .... 5... .... 6....
document.  Each file can be edited separately and text can be
easily copied or moved between documents.  The windows feature
is useful any time you are working on multiple related
documents.

In order to open a window, you simply press the COMMAND key
(F13) followed by the letter W.  A bright square will appear
along the edges of the screen and the message "Set Window" will
appear.  You can move the cursor along the top or side of the

 (1 .... 1... .... 2... .... 3... .... 4... .... 5... .... 6....
From time to time readers have called us to ask if it is
possible to import data into a Multiplan spreadsheet.  They had
numbers from the stock market, or financial packages or
whatever, and they didn't want to bother rekeying them into
their spreadsheet.  We have said that although it is not as easy
as it is on some other spreadsheet programs, that it is
possible, using a variety of methods.
```

*** See the NEW products W-2 Link and 1099 Link ***

SOFTWARE FOR THE FORTUNE 32:16 AND FORMULA

NEW PRODUCT!! W-2 LINK ($495.00) - this product is designed for companies with over 250 employess who must report W-2 payroll information to the Social Security Administration (IRS) via magnetic tape or diskette. The product automatically takes information in the Business Accounting System (BAS) Payroll module and puts it onto diskette in the format required by the IRS. Available for PAC software also.

NEW PRODUCT!! 1099 LINK ($295.00) - this product produces Form 1099's required for submission to the IRS and is an extension to the standard BAS Payroll module.

DOS LINK ($495.00) - provides easy-to-use menus and programs that transfer data between any PC (including Laptops) running DOS and the Fortune system. Here are just a few examples of how this product can be used:

- Transfer Multiplan or Fortune:Word Documents from or to your DOS system.
- Transfer other ASCII or binary files.
- Use the PC or Laptop as a remote terminal on the Fortune system.
- Transfer THOROUGHBRED (SMC) BASIC data that has been extracted using one of our other products

Documentation and software for both the DOS and Fortune system are provided. Data may be transferred via a direct connection (9600 baud) between the two computers or via modem (at modem speed). You may also order additional hardware for your PC that may be required to accomodate the transfer.

If you are interested in purchasing a Laptop computer that can be used with this product for people on the road or at home, give us a call!! We also provide consulting services for interfacing your DOS system to the Fortune ($50/hr).

MULTIPLAN LINK - can pass data from BAS/IDOL/BASIC files to Multiplan and back. For example, it could be used to send data from any of the BAS files (Chart of Accounts, Customer Master, Inventory Master, etc). Allows selection of fields to be sent and provides logical retrieval and key range selection. $295.00

RECORDS PROCESSING LINK - can pass data from BAS/IDOL/BASIC files to Records Processing and back. For example, it could be used to take customer address data from the Customer Master and will automatically make the Records Processing List document for mail/merge. Allows selection of fields to be sent and provides logical retrieval and key range selection. $295.00

ASCII LINK - can pass data from BAS/IDOL/BASIC files to ASCII and back. Use to send data between machines or databases. For example, it could be used to send data from any of the BAS files (Customer Master, Inventory Master, etc). Allows selection of fields to be send and provides logical retrieval and key range selection. $295.00

TERMINAL/PRINTER LINK - provides capability to print data on a printer connected to a workstation. Can print BAS/IDOL/BASIC reports, Multiplan and Fortune:Word documents. BAS/IDOL reports can also be viewed before printing, converted to Fortune:Word format; accumulated for mass printing; backed up to diskette; printed on any system printer. An option to print BAS/IDOL reports on a LASER PRINTER is $50 extra. Many other options are included. $195.00 (Fortune:Word printing on a printer connected to a workstation is $50 extra)

TELEPHONE LINK - on line telephone message, telephone directory and inter-office mailbox/menu system. Telephone message entry resembles typical pink slip. Telephone directory can search by name or company. Mailbox/message system can send and receive Fortune:Word documents or Multiplan documents. Does much more! Give us a call. Written in Business Basic. $295.00

CALCULATION LINK - provides programs for amortization, depreciation, loan repayment, averaging (with graph), linear correlation and breakeven analysis. Written in Business Basic. $95.00

KOMPACT PERSONNEL ACCOUNTING - provides for data capture of personal, wage/salary, job, education, salary/wage, appraisal training, dental, medical and life insurance data. Includes over 80 pre-defined reports/worksheets. Written in Business Basic/IDOL. $995.00

WESTERN LINK - provides instruction on how to use your Fortune for sending/receiving telex messages using Western Union's Easylink. Can replace your telex! Fortune ITE or Handshake software required. $95.00

UNI-KOMP 9803 Rocktree Houston, Texas 77040 (713) 937-7782

Finding and Sparing Bad Blocks

by Rich Suslick

This is the third in a series of articles written by Richard Suslick dealing with the intricacies of the UNIX file system. The last article addressed the fragility of the UNIX file systems and how the program fsck is used to maintain integrity. This article deals with how to correctly spare bad blocks and recover from any file system damage that may have occurred.

Rich is president of Cedar Cliff Systems Corporation which is located in Washington DC, (near Alexandria, VA). He provides UNIX training, professional software and management consulting. Mr. Suslick specializes in database and communication design issues and has had 5 years experience with Fortune Computers and 12 years experience with UNIX.

INTRODUCTION

If you have read the last article regarding the UNIX file system and fsck, you already know how delicate the file system structure under UNIX is. Many of you have had the unfortunate experience of suffering from an aging disk drive which has suffered a bad block failure. This article addresses how to correctly spare a block and how to identify which file(s) may have been affected.

The reasons for bad block failures are many and are beyond the scope of this article. It will suffice to say that occasional failures of the controller and/or disk drive are possible and the formatting information or data information on that block will become unreadable. We say "unreadable," but not necessarily lost. This article will attempt to provide you with the basics for salvaging that information, if possible.

There are a fair number of disk device errors that the system may identify. Until Release 2.0 of FOR:PRO, no bad block and disk error monitoring was provided by the operating system kernel. The *only* real way to properly identify disk failure problems is by running the standalone disk diagnostic software.

The FOR:PRO port of UNIX is incapable of providing accurate displays of intermittent or erratic disk behavior. However, there are diagnostic programs available that will perform this function. Copies of the **Hardware Diagnostic Disks** may be purchased from Fortune/SCI or duplicated from another copy.

Bad Block Sparing

Before starting the sparing operation users should always have a copy of the current disk configuration(s) for their system. To print a copy of your current configuration log in as root and issue:

```
/etc/rdconf  /dev/hd#0  | lpr -h
```

where the # in hd#0 is the drive number (usually 0).

The output should look something like Figure 1.

Much of this information was discussed in the first article of this series. Having a printed copy is very important. It as an excellent practice to make periodic copies of all the files in the **/etc** directory, as well as the configuration block, and keep these with your system backups. These files are critical in many hard disk crash recoveries. To copy the configuration block to a file, use the following command:

```
mkconf -c /dev/hd#0  filename
```

where the # in hd#0 is the drive number and the file name may be any UNIX file name. Usually, use the file **/etc/CONF#** which will then be included in your copy of the **/etc** directory.

Once a block is identified as bad it is necessary to spare it out. If you are fortunate enough to be able to successfully read the block after receiving a diagnostic, copy the file involved using the rules defined below. Once spared, delete the old file from the system.

You must have the **Development Utilities** software on your system to perform these operations. If the command **/etc/icheck** does not exist on your system, then you're out of luck.

Fortune provided 46 spare blocks available for this process. Most disks

come with some bad blocks. The printout of the configuration block above is typical for a 70 megabyte drive.

When a hard disk error occurs two numbers are provided, the disk physical block number and the file system block number.
For example:

hd error: CRC error in ID field on drive 0 in state move reading block 77559 error reading block 68855

block 77559 is the file system physical block number (**FSPB**) and 68855 is the file system (partition) block number. Note that the file system block number is the physical block number less the starting block of the partition it would fall into. In our example 68855 + 8704 = 77559. Obviously this error occurred in partition 2 of drive 0 (**/dev/hd02**).

To find the file which contains the bad block, convert the physical disk block number to the file system block number. As described above, subtract the starting block of the partition the error would be located in eg. 77559 - 8704 in our example. Take this number 68855 and divide by 2 dropping any remainder eg. 34427. This is the logical file system block number, the number **icheck** will work with.

The program icheck is a subset of **fsck** and is no longer used for checking the integrity of the inodes. It has a useful option, **-b**, which allows the logical block number to be specified during its check. When the block listed using the **-b** option is found, the inode for that block is returned. If no block is listed it means that the block is not allocated in the file system and may be spared with no data loss.

For example:

/etc/icheck -b 34427 /dev/hd02

/dev/hd02:
34427 arg; inode=5694, class=data (small)
files 5415 (r=4841,d=362,b=32,c=180)
used 56681 (i=922,ii=35,iii=0,d=55724)
free 7159

Notice the block 34427 has been found and is inode 5694. Now all we need do is identify the name of the file using the inode number. The **find** command or the **ncheck** command may be used for this purpose. Notice with **ncheck** we use the name of the file system on which the block is in error (**/dev/hd02**). Using **find** with the inum option the mounted file system is searched. If you have multiple drives this may yield names from all file systems which have matching inode numbers unless you restrict the search to just the file system in error.

/etc/ncheck -i 5694 /dev/hd02
/dev/hd02:
5694 /m/wp2/string/wpDICT.EN

or

find / -inum 5694 -print
/m/wp2/string/wpDICT.EN

Having determined the file in error, you can now spare out the bad block. Previous /u/fortune news' have discussed how to spare out a block number correctly. Please refer to the Volume 3 Number 1 of /u/fortune news. Also, see the side bar excerpted here entitled "Saving The Block."

One of the most common misconceptions about bad block sparing is that once spared the bad block seems to go away and the system is ok. THIS IS CLEARLY INCORRECT. Any file which contains a bad block which is spared may appear readable again but actually will have a block of garbage somewhere in it. YOU MUST ALWAYS RESTORE OR DELETE THE GARBAGED FILE OR UNPREDICTABLE RESULTS MAY LATER OCCUR.□

Figure 1

```
Configuration for: /dev/hd00
          Disk ID: N70 - 70 Mega byte          Volume ID: 562556972
      Format time: Thu 29 Oct 20:49 1987      Media type: Hard
      Modify time: Sun 22 Nov 23:39 1987      Disk type: N70
    Sectors/track: 17                    Drive attributes: 0
            Heads: 8                   Software interlace: 1
        Cylinders: 1024               Hardware interlace: 2
       Block size: 512                 System interlace: 2
    Landing zone: 1025                     Write precomp: 1025
Partition 0 start: 0                  Partition 0 length: 272
Partition 1 start: 272                Partition 1 length: 8432
Partition 2 start: 8704               Partition 2 length: 130288
     Boot 0 start: 68                       Boot 0 length: 86
      Total reads: 1211685          Total ECC/CRC errors: 19
     Total writes: 567028          Total soft ECC errors: 0
    Total retries: 0              Total sector ID errors: 0
     Total spares: 46                  Total other errors: 0
Diagnostic spares: 15-16 32-33 Unusable spares: 14 17-31 34-45
       Used spares: 0(42575) 1(77559) 2(137946) 3(72734) 4(113555) 5(110301)
                    6(131411) 7(72016) 8(73278) 9(123640) 10(136585) 11(104932)
                   12(120462) 13(8723)
```

Saving The Block

*(This side bar is actually a reprinted portion of an article that appeared in Volume 3 Number 1 of /u/fortune news. Written by **Guy Washburn,** this section describes how to deal with a bad block of data once it has been identified.)*

If for some reason there is a problem with the backup (not having one is definitely a problem) there is still a good chance of saving the information in the block with the next procedure.

With the **FSPB** in hand we do the following

```
# dd if=/dev/rhd02 of=/tmp/[FSPB number]
count=1 bs=512 iseek=[FSPB] <return>
```

This should return:

```
1+0 records in
1+0 records out
```

If it does not print this, or if a hd error message comes up, try again, it may take several attempts to get it to work, we are dealing with a flaky block remember.

Sparing The Block

Once the **dd** works we can spare the block. The process of sparing a block has the result of causing the system to use an alternate good block (located in partition zero) as a replacement for the spared bad block. The kernel performs this operation automatically based on the spares list kept in the configuration block which is read when the system boots up. To spare a block then we must make a new entry in the spares list for the block that has turned bad. To do this we use the following command.

```
# mkconf -i /dev/hd00 /dev/hd00 <return>
```

At this point the first entry in the configuration block will be displayed with '?' as a prompt. Press <return> to accept the default on each of the questions until you reach this question:

```
Number of Spare Blocks = 46 ?
```

Enter `46 <return>`

At this point any blocks that have been spared will appear after the spare entry number (they start at three, the first are used for diagnostics).

Press return to accept each one until you get to one that says 'Bad?'. On the first 'Bad?' entry enter the DPB number from the error message or diagnostic.

On the next 'Bad?' entry type `done <return>`.

(Note. If the spare entry number comes up with 'Free?' next to it, you should patiently go through the remainder of the spare blocks and enter

```
bad <return>
```

for each one. Otherwise the kernel will try to spare blocks for us, randomly blowing away blocks that may not even be going bad without saving the information the block contains. Not a good idea.)

By typing done, the configuration block is automatically written back out to the hard disk. We can then check the results of our efforts by using rdconf to read the conf block which should now show our changes.

At this point we shut down the system to allow the change to take effect.

```
# > /tmp/.powerdown <return>
# sync <return>
# sync <return>
```

Now press the reset button to restart the system and when it is up login as root.

Restore The Block

Now we have this image of the contents of the flaky block hanging out in /tmp let's put the data into the spare we just defined at the stand-in for the troublesome block.

```
# dd if=/tmp/[FSPB number] of=/dev/rhd02
count=1 bs=512 oseek=[FSPB] <return>
```

This will return

```
1+0 records in
1+0 records out
```

And now if everything has gone correctly the data should be restored.

It is a good idea at this point to run a file system check at this point to see if everything is ok.

Epilog

Now that we are finished with this digression (salvation?) it does not mean that backups are outmoded. This procedure will work if the error is a "PSOFT" error for those who have diagnostics. If you don't have the diagnostic, but still feel like taking a swing feel free. But please note

THIS PROCEDURE IS NOT FOOLPROOF.
IT IS NOT GUARANTEED TO WORK.
AND IT MAY NOT SAVE YOUR DATA.

But then it just might save the day, unfortunately this is the nature of computers.

It is also a good excuse to learn a bit about UNIX and how the system works - which is never a bad idea.

This article is based on a piece in the Fortune hardware reference guide but has been corrected, adjusted and translated into English for the average system operator.□

System Administration: Part 7

No Longer Requires a Degree in UNIX Wizardry

This series is a must for all owners and users of the Fortune computer. Dave is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the Fortune Computer, and is president of the Houston UNIX User's Group. He contributes independently to ***/u/fortune news.***

by Dave Kloes

In the last two articles, we discussed two of the **FOR:PRO** editors - "**ed**" and "**vi**". In addition, we told you about a third editor that is available through the Cambridge Consortium called "**screen**". In Part 7, we will continue our discussion of the "`/etc`" directory and talk about files that can be modified using any of these three editors. Again, we emphasize that these files are normally changed or modified by using selections from Fortune's "Global Menu" system. Our purpose here is to show you how the same functions can be accomplished directly from **FOR:PRO**.

The first file we will discuss in the /etc directory is the "`group`" file. Here is what the standard "`group`" file looks like when you first install **FOR:PRO**:

```
sys::0:bin
daemon::1:@,?
operator::2:
bin::3:
users::100:
proto::100:
```

First of all, we must caution you that for any of the `/etc` files we will be discussing, you should not modify or change the standard entries unless you are sure you know what you are doing. The concept of the "`group`" file is quite simple. It allows you to assign a new group to the system so that



members of that group can have joint access and similar permissions for a file or directory. Remember when we discussed the "permission bits" for a file, the second group of three bits showed the permissions for those that were a member of the group for that file or directory:

```
- r w x r w x r w x
  | owner | group | other
```

The "owner" of the file uses the first set of three permission bits, members of the the group use the second set of three bits, and everyone else uses the "other" set of permission bits. Remember that "`rwx`" stands for read, write, and execute permissions. You should also recall that the permissions for a file or directory can be seen in a given directory by issuing the "`ls -al`" command.

Let's take a look at one of these lines and break it down for you and then we will show you how to add your own "groups" to the system:

```
users::100:
```

There are four separate "fields" or pieces of information that can be identified here. Each field is "delimited" or separated by a colon (:). The first field of information is the group name which in this case is "users". The second field of information would be the encrypted password for the group. In this case, there is no password assigned as indicated by the "::". The third field is the group number which is 100. The last field of information would be the login name from the `/etc/passwd` file for each of the members of the group. **FOR:PRO** does not update this field of information so none are listed here.

Whenever you add a new user to the system using the "**newuser**" function, they default to group "`100`". Notice in the group file that there are two group names which have group id number "`100`" - users and proto. All users added to the system using the "**newuser**" function are automatically

assigned to the same group (100) and therefore everyone has read and write privileges to each other's files and directories. For security reasons, some of you may not want your system set up this way.

To add a new group to the system, you would simply edit this file and add the new group to the end of the file. Let's say we have three users on our system that we want to be members of the same group: maxb, harolda, and jackb and we want to name the group "projecta". Here is what the line we add would look like:

```
projecta::101:maxb,harolda,jackb
```

Here we added the new group "projecta" and assigned the next sequential id number of 101 to the group. We did not assign a password, because while the field is there, **FOR:PRO** has no way to assign a password to it. We also added the group members, separated by commas, to the end of the line. This would allow all three members of the group to access each other's information. We will come back to the discussion of the "group" file, but first we must divert to an explanation of the "/etc/passwd" file.

The /etc/passwd file and the /etc/group file work together. Like the /etc/group file which is used to add new groups to the system, the /etc/passwd file is used to add new users to the system. We will display a few sample lines so that we can discuss what each of the elements of a line entry are:

```
root:OP0vw/7mWOdsk:0:1:Maint.Account:/:
manager:qq4F3wNE:0:1:Man.:/:/bin/menu
shutdown::0:1:shut down computer:/:/etc/
shutdown
davek:1dlcDLaLu3kaA:101:100:prototype
account:/u/davek:/bin/menu
```

Again, like the group file, each field of information is delimited by a colon. Here is a breakdown of each of the fields in the "davek" line and what they are:

davek is the login name for the user.
1dlcDLaLu3kaA is encrypted password for the user.
101 is the user ID number.
100 is the group ID number (from /etc/group file).
prototype account is a comment field.
/u/davek is the user's home directory.
/bin/menu is the command to execute when they login.

Login names for the **UNIX** system are normally the user's first name and first letter of the last name. This convention allows us to easily identify which users are logged in to the system. One common situation we find that will get you into trouble is to have all users logging in with the same login name. **FOR:PRO** has a limit on the number of processes a user can perform. If all users are logged in with the same login name, the potential to reach that limit is increased. Another situation that we find is where all users log in as "manager". Because "manager" has permission and access to all information and files on the system, this is asking for trouble. Each user on the system should have their own unique login

We are the *largest* Authorized Fortune Master Dealer in the Pennsylvania and Delaware area. As part of the **P**rofessional **S**olutions **G**roup of companies with over 35 years in business automation, we have the *stability* that our customers demand

Our services include a factory trained field engineering team, a complete parts inventory including system upgrade capability and a software support department.

For all of your Fortune needs contact:

Mark Shields of PSG at
2550 Eisenhower Avenue
PO Box 762
Valley Forge, PA 19482

or you can call Mark at (215) 666-1400

change a user's password, but there is no way to find out what a user's password is. If a user forgets their password, "root" or "manager" can change the password by issuing the "**passwd**" command followed by the user's login name:

```
# passwd davek
```

The above example would allow us to change the password for "davek". A user can change their own password (assuming they have access to **FOR:PRO**) by simply issuing the "**passwd**" command but cannot change another user's password.

A password can be removed entirely by editing the /etc/passwd file and removing the encrypted sequence between the second and third colons:

```
davek::101:100:prototype
account:/u/davek:/bin/menu
```

After the encryption sequence is removed, the user will no longer be prompted for a password.

When the "**newuser**" function is used to add a new login account, the next user ID number in sequence is used. Notice that in our short sample above, that "root", "manager" and "shutdown" all have the same user ID number (0) and group number (1). That means that when we log in as any of these three accounts, each would have super-user or "root" permissions. If we create another user account using any other name and give it the same ID number and group ID number as root, it also will have "root" level permissions.

One common problem that you will see, especially after you have restored information to the hard disk, is that when you do an "ls -al" command, you will see a number where you would normally see the owner's name for the file.

```
-rw-rw-rw 1 101 523 Nov  7 23:30 file1
```

In order to understand what the problem is here, you must understand that **FOR:PRO** does not store the actual owners name for the file - it stores the user ID number. When we issue the "ls -al", it finds the ID number and then looks it up in the /etc/passwd file and converts it to the login account name. Whenever we see the number displayed instead of the name, that means that it could not find that ID number in the /etc/passwd file. This indicates that the administrator would have to either correct the passwd file or run the "**chown**" command on the file or directory to bring the file or directory back into sync with the passwd file. Failure to do this could mean a user will get "permission denied" errors when trying to work with the file. The out-of-sync condition is normally caused by either restoring an old passwd file or files/directories that were created before the passwd file was updated.

The comment field defaults to "prototype account" when a user is added using "**newuser**". It is a good idea for the administrator to put the person's full name, phone number, room number, etc. in this field, espe-

PROGRAMMERS

The leading manufacturer of status display panels for the fire alarm and security industry is seeking programmers and junior programmers to develop new applications and refine existing programs.

Successful candidates will be experienced in IDOL/BAS and SMC Business Basic for Fortune and Basic 4 systems. Familiarity with UNIX and system set-up and maintenance is desirable.

Candidates interested in joining a growing team-oriented organization located in Falls Church, Virginia should write or call with resume and salary history, to:

WSA
308 Hillwood Avenue
Falls Church, Virginia 22046
703 241-0687

Expand your Fortune® without spending a fortune.

New equipment costs big bucks. The Good Fortune Trader sells fully tested burned-in equipment, refurbished to Fortune® standards at prices that won't make waves with your cash flow. Call or write today for a listing of our specials.

THE GOOD FORTUNE TRADER

60 DAY WARRANTY
a division of Computer Services of Atlanta
(404) 447-4151
5825 Live Oak Parkway, Suite 2-C, Norcross, Georgia 30093

cially on large systems so that the administrator can contact or identify the user in case there are any system problems.

When you use the "**newuser**" function, **FOR:PRO** adds a directory for that user in the /u directory. This directory is called the "home directory" and it is the directory that the user is placed in when they login. It also contains the files the user creates when using **Fortune:Word, Multiplan**, etc.

Also when you use the "**newuser**" function, the default program that is run for the user is "/bin/menu" which is the command that displays the **Global Menu**. This command could be any valid **FOR:PRO** command and can be used to do some interesting things as far as setting up our system. If you look at our samples above, you will see that when a user logs in as "shutdown", the "**/etc/shutdown**" command is run, which is the script for shutting the system down. If we had a user, for example, that we only wanted to do **Fortune:Word**, we could have **Fortune:Word** executed when they log in by changing the "/bin/menu" to "/bin/wp2". In this case, they would not even see the **Global Menu**. The first thing they would see would be the **Fortune:Word** menu. When they logged out of **Fortune:Word**, they would go directly back to the login prompt. Here are some other direct commands that could be executed:

Fortune:Word	- /bin/wp2
Multiplan	- /usr/bin/master
IDOL	- /usr/bin/IBASIC
BAS Accounting	- /usr/bin/BASIC
Basic Interpreter	- /usr/bin/DBASIC
BAS Training	- /usr/bin/TBASIC
FOR:PRO (shell)	- /bin/sh
FOR:PRO (C-shell, if avail.)	- /bin/csh

Since the /etc/passwd file is a file that can be edited, we now have the ability to directly control not only what directory a user is placed in when they log in, but also what command will be executed.

Let's go through an example where we create a new account for a user without using the "**newuser**" function:

1. First we add a line for the user in the /etc/passwd file:

```
jackb::102:101:Jack Brown, 656-4129:/u/
jackb:/usr/bin/BASIC
```

2. Then we add the directory for the user:

```
# mkdir /u/jackb
```

3. We make jackb the owner of his directory:

```
# chown jackb /u/jackb
```

4. We add jackb to the /etc/group file for the "projecta" group since we put him in group "101":

```
projecta::101:jackb
```

5. We add a password for the "jackb" account:

```
# passwd jackb
```

This is essentially what happens when a new user is added using the "**newuser**" function and the "S2" (System Management), "Accounts" selection. The System Administrator who understands how and where to change this information can set up his system for particular needs.

Let's go back to our "/etc/group" file. We wanted to discuss the /etc/passwd file so that you could see the relationship between the /etc/passwd file and the /etc/group file. As you saw above, we can control what group a login account belongs to by editing the /etc/passwd file "group ID number" field.

We can also tell what group a file or directory belongs by using the "g" flag on the "ls -al" command. If we issue the "ls -alg" command in a given directory, we will see the group name instead of the owner name that is displayed with the "ls -al" command.

We would also like to mention here that the "**newgrp**" command is available and can be used to change from one group to another. For example, if we are logged in as group "users" and want to change to the "projecta" group:

```
$ newgrp projecta
```

After this command is issued, we would be working as group "projecta". If we are an unauthorized member of the group (as specified in the /etc/group file), we will get a "Sorry" message. To change back to our default group, we would simply enter the "**newgrp**" command by itself with no following arguments.

Not everyone will care about setting up users into separate groups. For those of you that have the need, we have another suggestion. As you can see from the discussion of groups and the /etc/group file, maintaining groups in **FOR:PRO** takes more effort than most System Administrators have time for. Our recommendation would be to set up a separate login account for the group and only give the account name and password to the group members. If you do this and take away "rwx" permissions for that login directory for the "group" set and "other" set, you now have a secure area for those group members to work in. If a group member is removed, you merely have to change the password for that login account.

In our above scenario, we could have added a login account name called "projecta". We would create the home directory "/u/projecta" and would then issue the following command:

```
# chmod 700 /u/projecta
```

The permissions would look like this in **FOR:PRO** for the /u/projecta directory:

```
drwx——
```

We will discuss the "**chmod**" command in detail in another issue. This method of maintaining groups has a lot less overhead than working within the "group" structure.

Next time we'll talk about some other files that can be edited in the /etc directory.□

Multiplan

Continued from page 1

specified file, you can put Multiplan into the *player piano* mode. In order for this to work, you have to put every keystroke, that you would normally press on the keyboard, into a file. Any time that Multiplan expects a RETURN, you will need to insert a CONTROL-M. This method is useful if you just have a few numbers to insert into a sheet, or if you change one number in a set of sheets that are linked to each other and want to successively recalculate each sheet.

We won't discuss this method at great length here, but we will supply you with an example of what we mean. Let's say we want to load a spreadsheet called **budget**, go to the "sales" cell and enter the number 10000. Then we want to save the spreadsheet and quit Multiplan. If you don't do this last step, the computer will be stuck in Multiplan and you'll have to kill the process or reset the computer.

The file looks like this:

```
TLbudget^Aq^M
GNsales^Aq^M10000^Aq^M
TS^Aq^MY
QY
```

The **^A** is a CONTROL-A and the **^M** is a CONTROL-M. If you use the **sc** or **screen** editor, you type CONTROL-C before entering any CONTROL character. Note that in the second to last line we are entering **T** to Transfer, **S** to Save, **^Aq^M** to EXECUTE, and finally **Y** to answer Yes to overwrite the file.

You run this "program" by typing:

```
mp68 < mpauto
```

assuming that mpauto is the name of the file which contains the commands described above. Note that before this will work you must create the budget spreadsheet and assign the labels "sales" to one of the cells.

Please note that although this is a "cute" way to import data, and may work for many applications, it probably isn't the best solution. In our quest for perfection, we decided to research another method, the **SYLK** file format. (**SYLK** is pronounced like silk).

Using the SYLK format

Normally Multiplan saves its spreadsheets in its own special format that is not accessible by other programs or by individuals, like ourselves, who desire to import data. However, under the **Transfer Options** selection, there is an option to save or load files in the Symbolic, or SYLK format. Unfortunately the Multiplan documentation doesn't go into much detail on how the SYLK format works.

We began to experiment by saving our spreadsheets in the SYLK format. This format produces an ASCII file which can be edited using editors like **screen** or **vi**. Unfortunately, the SYLK format isn't entirely transparent. So we placed a call to Microsoft, the big company that was responsible for Multiplan in the first place.

Within a few days we received a 17 page document explaining the SYLK file format. Given the new found information, the files were slowly decoded. We can now report that SYLK files are not all that complicated and you should be able to use them without too much trouble.

There are basically two ways to use SYLK files. The first is to design a spreadsheet from scratch using the SYLK format. This approach will be useful if you have a lot of data to use in the spreadsheet. The second way is to set up a spreadsheet inside of Multiplan as you normally would, and then save it in the SYLK format. Once it is saved, you can go in and edit individual cells if you just have minor changes to make. We'll talk about both methods.

SYLK Basics

For the purpose of our discussion, we will work with a simple spreadsheet that is shown in Figure 1.

This is a simple budget form for a generic company. The spreadsheet is four rows by five columns. There is one obvious column of formulas, the product column in column five. The basic SYLK version of this spreadsheet is shown in Figure 2.

Figure 2 may look complicated at first—it is certainly more complex than the actual spreadsheet. However, it's fairly straightforward. Note that each row begins with a letter (or letters) followed by a semicolon. This tells Multiplan what *kind* of entry the row is. The two most common are **F;** for format and **C;** which presents the content of any particular cell. After the first identifier there is often a **Y** or an **X**. These pinpoint the column and row of the cell. **X** defines column addresses and **Y** designates rows. For example **X2Y3** stands for Column 2 Row 3. (I find this is the opposite of what I would expect). A **K** precedes the content of the cell if it is a number or alphanumeric string. Strings are enclosed in quotation marks. Finally, an **E** in the text says that there is an expression or formula that follows.

Let's look at our example.

It begins with certain standard features.

Figure 1

```
#1      1          2          3          4          5
   1 Budget Sheet for The ACME Computer Corp, Inc.
 2 Rent             12 mos @          2,500     30,000
 3 Utilities        12 mos @          1,000      1,200
 4 Salaries         12 mos @         15,000    180,000
```

Figure 2

```
ID;PMP
F;DG0G10
B;Y4;X5
C;X1;Y1;K"Budget Sheet ..."
C;Y2;K"Rent"
C;Y3;K"Utilities"
C;Y4;K"Salaries"
C;X2;Y2;K12
C;Y3;K12
C;Y4;K12
C;X3;Y2;K"mos @"
C;Y3;K"mos @"
C;Y4;K"mos @"
C;X4;Y2;K2500
C;Y3;K100
C;Y4;K15000
C;X5;Y2;ERC[-3]*RC[-1]
C;Y3;ERC[-3]*RC[-1]
C;Y4;ERC[-3]*RC[-1]
E
```

```
ID;PMP
F;DG0G10
B;Y4;X5
```

ID;PMP tells Multiplan that this is a Multiplan SYLK file. SYLK files are also used by other IBM-PC products like Microsoft Chart. The **F;DG0G10** supplies the global default format for all cells. We haven't totally figured that one out, but we'll talk about formatting shortly. The **B;Y4;X5** tells Multiplan to expect a spreadsheet four rows by five columns. These numbers have to be correct in order for the spreadsheet to load.

After the preliminaries are out of the way, we get into the heart of our work. Here's the contents of the first column:

```
C;X1;Y1;K"Budget Sheet for
...."
C;Y2;K"Rent"
C;Y3;K"Utilities"
C;Y4;K"Salaries"
```

All of these entries begin with the **C** or Contents identifier. Note that the very first row specifies **X1;Y1** which points to Column 1 Row 1. The **K** is followed by the text that will be entered into that cell. Note that the right half of this cell will be hidden because we haven't yet specified that the cell should have a continuous format.

The row that contains "Rent" does not have an **X** specification. If the **X** (or **Y**) designation is omitted, the previous row or column will be assumed, in this case column 1 (X1). Note that our Y pointer points to Y2. This is incremented for the next two labels in our spreadsheet.

Column 2 is simple for the moment.

```
C;X2;Y2;K12
C;Y3;K12
C;Y4;K12
```

Note that since we moved over to a new column the X2 notation was included. For the next 2 rows it's omitted as in column 1. Also since the 12 is a numeric value, there is no need to use quotation marks. Columns 3 and 4 are basically similar to the columns we have discussed. Column 5 introduces formulas into the spreadsheet to produce the product of the number of months times the unit cost.

```
C;X5;Y2;ERC[-3]*RC[-1]
C;Y3;ERC[-3]*RC[-1]
C;Y4;ERC[-3]*RC[-1]
```

The **E** prefix says a formula is on the way. The formula itself looks very much like the way it would within Multiplan.

Finally, the **E** in the last row designates the end of the spreadsheet.

Loading the SYLK file.

To load this spreadsheet you will need to create a file with an editor like ed, screen, or vi. The name of the file must end with a ".SL", like **budget.SL**. The ".SL" won't be seen once you're in Multiplan, but it tells Multiplan that it is loading a SYLK file. Regular Multiplan files have a ".MP" tacked onto the end of their names. After you create the file, go into Multiplan as you usually would. Next choose **Transfer Options** and select **Symbolic**. Finally, use the **Transfer Load** command as you normally would to load the file. Note that the ".SL" suffix doesn't show up. While Multiplan is reading in the

Vantine Computer Services

Professional Solutions for Business Microcomputer Users

Sales Service Consulting
Software Customization Support
Software/Hardware Upgrades

Over Three and a half years supporting Fortune Users

Vantine Computer Services
70 Walnut Street, Wellesley, MA 02181
(617) 239-8084
Authorized Fortune Dealer

file, it will report that it is reading in lines. For complicated spreadsheets this may take some time as the program translates from the SYLK format.

Sprucing Things Up

The first change is on the global format line. If we change it from F;DG0G10 to F;**K**;DG0G10, we change the Format Options to allow for commas. To take advantage of that we need to change some of the cells to Integer. To do this we add the following line after each of the cells we want to format:

```
F;FI0D
```

The **F** stands for a format line. The rest of the row is divided into three sections. The first section is the character format. **I** is the first field in our example. You can use any of the letters that are used in Multiplan, e.g. **C** for Continuous, **D** for Default, **F** for Fixed, **I** for Integer, **$** for Money, etc. The **0** is the second field. It contains the number of digits to display for decimals, etc. Finally the letter **D** is the third field which defines the alignment. These are analogous to the options within Multiplanthat we just outlined above. We don't specify X or Y in our example because we are formatting the same cell as the row before.

Next we want to format Row 1 Columns 1 to 5 as Continuous so the entire title will show up. The code for this is **F;FC0G**. This is inserted in the row after the title. In addition, we have to add extra entries for the other cells in the row that were empty but need to be formatted as continuous. Our first few rows now look like this:

```
C;X1;Y1;K"Budget   Sheet
for...."
F;FC0G
F;X2;FC0G
F;X3;FC0G
F;X4;FC0G
C;Y2;X1;K"Rent"
```

Since the cell pointer was moved over to column 4 in the last formatting line (which referred to row 1 column 4), it is necessary to amend the "Rent" line and reintroduce X1 which moves the pointer back to column 1.

Lastly, we'll insert a TOTAL cell for the budget. That involves the following few lines:

```
C;Y5;X5;ESUM(R[-4]C:R[-1]C)
F;FI0D
```

Since we have now made the spreadsheet 1 row longer, the **B** entry at the head is changed to **B;Y5;X5**.

That's it. We now have a new spreadsheet with proper formatting. The final file is shown in Figure 3.

Figure 3

```
ID;PMP
F;K;DG0G10
B;Y5;X5
C;X1;Y1;K"Budget Sheet..."
F;FC0G
F;X2;FC0G
F;X3;FC0G
F;X4;FC0G
F;X5;FC0G
C;X1;Y2;K"Rent"
C;Y3;K"Utilities"
C;Y4;K"Salaries"
C;X2;Y2;K12
C;Y3;K12
C;Y4;K12
C;X3;Y2;K"mos @"
C;Y3;K"mos @"
C;Y4;K"mos @"
C;X4;Y2;K2500
F;FI0D
C;Y3;K100
C;Y4;K15000
F;FI0D
C;X5;Y2;ERC[-3]*RC[-1]
F;FI0D
C;Y3;ERC[-3]*RC[-1]
F;FI0D
C;Y4;ERC[-3]*RC[-1]
F;FI0D
C;Y5;X5;ESUM(R[-4]C:R[-1]C)
F;FI0D
E
```

In Conclusion

If you are just planning to enter all of this information with screen in an ASCII file then you may as well just use Multiplan. The idea of using an ASCII file is to let a program or shell script create the file for you. If you have a spreadsheet that mostly stays the same with a few changes each time it's loaded, you may be able to use a shell script with a few variables in it to change the numbers. If you have a database, you can probably get it to output the standard spreadsheet and insert the appropriate values where needed. You could even use Records Processing to create the spreadsheet if you strip the \B\ control sequences and any other \ commands from the Fortune:Word file to create a regular ASCII file.

There are also some vendors, such as UNIKOMP which have software packages which will help you to produce the SYLK files.

Josh Lobel

If you have a spreadsheet that mostly stays the same with a few changes each time it's loaded, you may be able to use a shell script with a few variables in it to change the numbers.

New Disk #10: ispell

Yes, we are announcing yet another diskette with several useful programs. The main event, so to speak, is an interactive spelling checker. This program, called ispell, will search through an ascii file and detect misspelled words. It is very easy to use - and we will discuss it in greater detail later in this article. We have also included some other programs and utilities on this our tenth disk. They include:

Curse - this is mainly a program which allows you to use many of your terminal's capabilities. This can be used with terrific effect in shellscripts. For example, suppose you want to display a some word or phrase on your terminal in bold. To do this, you would simply issue the command:

curse -so ' PHRASE' -se

In general, curse gives the user a method of updating the screen from the shell. The options are two letter variable names from **curses.h** as detailed in Appendix A of **Screen Updating and Cursor Movement Optimization: A Library Package**. A manual page and other documentation is included on the disk.

Another utility included is stat - which gives statistics on files. For example, if I stat the program stat, I see the following display:

```
Filename: stat
Size: 12284          Filetype: Regular File
Mode: (0775/-rwxrwxr-x)        Uid: (   0/   root)
Gid: (    1/  daemon)
Filesystem Maj/Min (hex): 112      Inode: 2599
Links: 1
Last  read/access     :  Mon  Nov  30  10:01:22
1987(00000.00:00:00)
Last  write/create    :  Tue  Nov  24  14:27:36
1987(00005.19:33:46)
Write/change  inode:  Tue  Nov  24  14:27:36
1987(00005.19:33:46)
```

A third program is called vshell. vshell stands for visual shell and is touted as a useful shell for programming in. It is quite modifiable so that an individual can run commands with just one keystroke.

Finally, we do want to tell you a little bit about ispell. This is an interactive spelling checker which is very easy to use. If you have a text file that you would like to check, simply type `ispell filename.`

We have a little text file called spelltest which we checked. When we type ispell spelltest we see a screen like that displayed in Figure 1. Whenever ispell detects an error it prints out the misspelled word on the screen with a little bit of the surrounding text and then at the top of the screen presents you some alternatives for the misspelled word. In our example, ispell detected that hzve was misspelled and gave us the options of have and hive.

Figure 2 is what you would see if you ever type a ? (question mark) in ispell. This is a listing of the different options you have when a spelling error is detected. This display is self explanatory. ispell is a beautiful little program and no system should be without it. □

Figure 1

```
    hzve                File: spelltest

0: have
1: hive




We enjoy the Fortune/SCI hardware and software.  If we
didn't, we wouldn't hzve started /u/fortune news.
```

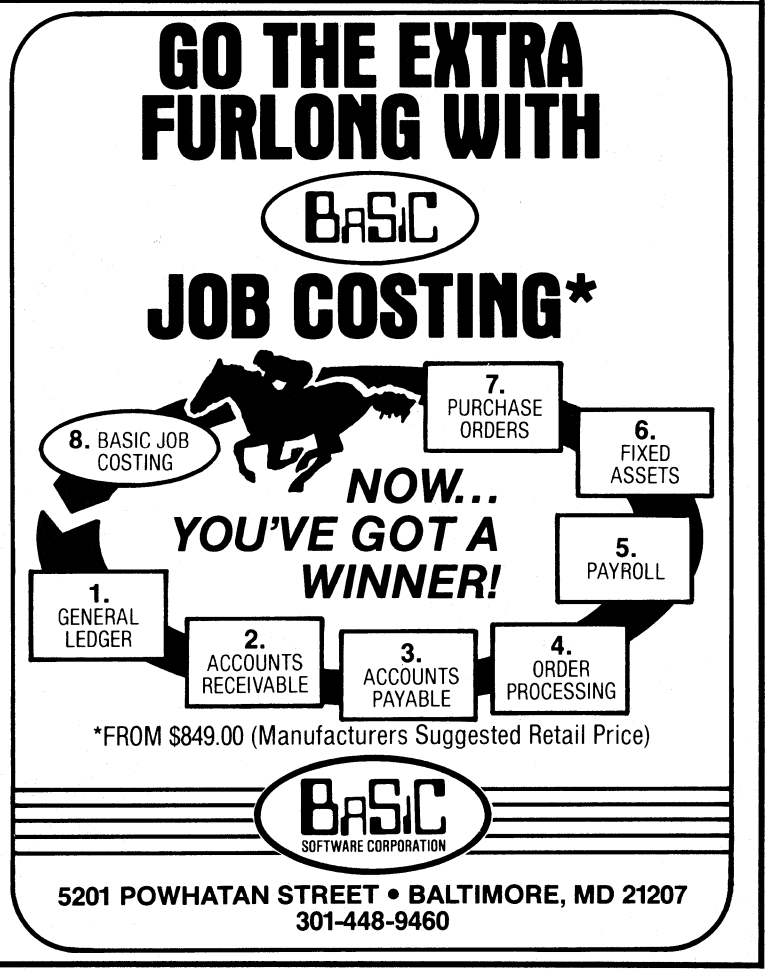
Figure 2

Whenever a word is found that is not in the dictionary, it is printed on the first line of the screen. If the dictionary contains any similar words, they are listed with a single digit next to each one. You have the option of replacing the word completely, or choosing one of the suggested words.

Commands are:

R	Replace the misspelled word completely.
Space	Accept the word this time only
A	Accept the word for the rest of this file.
I	Accept the word, and put it in your private dictionary.
0-9	Replace with one of the suggested words.
Q	Write the rest of this file, ignoring misspellings, and start next file.
X	Exit immediately. Asks for conformation. Leaves file unchanged.
!	Shell escape.
^L	Redraw screen.

— Type space to continue —



What An Opportunity



An opportunity like this doesn't happen very often.

Micro Resources is in a fix. The region's leading Fortune/SCI supplier's stock room is jammed with top-notch Fortune/SCI inventory — with more on the way.

So we're dropping prices to levels you probably thought you'd never see. Grab your wish list. This is a brief opportunity to upgrade your system you simply can't afford to pass up.

If you need more memory, additional terminals or faster motherboards, now's definitely the time to buy. But don't wait too long — **quantities are limited.**

Increasing your Fortune/SCI system's productivity isn't very difficult. And Micro Resources just made it a lot easier. Call Chuck Kennedy, 766-2335, today — before it's too late.

Micro Resources

255 Bradenton Ave., Dublin, Ohio 43017 614 • 766 • 2335

Borelli

Continued from page 1

our own family of workstations and multiuser UNIX based computers.

/u/fortune news: You have developed these not as a subcontractor, but as SCI's own products. Is that right?

Mr. Borelli: Yes, as SCI's own products.

/u/fortune news: And are those marketed as SCI's own products?

Mr. Borelli: Yes and no.

For example, we are selling SCI Banking products into large Regional Banks. We are also selling SCI Control System products into the Oil, Gas and Water industry.

We recently designed a new generation airline-ticket and boarding-pass printer for United Airlines which is being installed in O'Hare and Dulles airports.

We designed the product to United's specifications but we now have rights to that product.

/u/fortune news: I see, so you can market that to other airlines?

Mr. Borelli: Yes, and we have a number of units under evaluation by many of the major airlines.

We also have a number of major OEM customers that market our computers with their name on it and we have added some value to it by doing custom work.

By the end of this year, we will reach the point of having a full family of workstations and UNIX multiuser computers from 4 users up to 64 users with mainframe communications and Local Area Networks. We will also have the ability of running DOS as a subtask under UNIX. Then the decision was, "Now that we have the products, we need a distribution network."

/u/fortune news: You are anticipating my next question, which was, "Why buy Fortune?"

Mr. Borelli: That's the reason for Fortune. We bought Fortune mainly because they had an established Dealer network that they had invested quite heavily in over the last couple of years. And they were respectable - very good dealers. We felt with our manufacturing capability and the set of products we have and the products Fortune has with Fortune:Word, plus the distribution channel, that we could marry this together and make a successful business.

Our plan is to continue improving Fortune:Word - not only with graphics but other features are undergoing a major study right now.

/u/fortune news: So, as I understand it, SCI has developed UNIX machines from 4 to 64 users and other products. You say you eyed Fortune because of its strong Dealer channel but Fortune also has the hardware associated with it as well as software, which is clearly useful for your machines. But what about the dichotomy between the two types of machines - the Motorola 68000 microprocessor versus the INTEL 286's and 386's? Some people might look at that and ask "Does it make sense to keep both of those lines going?" Especially when you are "fortunizing" your equipment. Presumably, "fortunizing" means adding the Global Menu, Fortune:Word, etc., to your machines... Will there be a use for both types of hardware?

Mr. Borelli: Well, as you know, we have introduced both the Formula 4000 and the 8000. Both of those are being converted over to UNIX V.3 with standard UNIX networking. Once the Fortune applications have been converted to UNIX V.3, then we can t5transfer these applications over to the SCI computers which are already on UNIX V.3. At that time, we can have SCI 386 products and the Formula 68020 products in the same line.

/u/fortune news: Wouldn't it be possible to make some 386-based products that would support both 22 and 32 users - similar in some sense with the Formula 4000 and 8000?

Mr. Borelli: For the near term, our plan is to support both lines. We do have commitments to customers for both those lines and we are going to meet those commitments. And then what we are going to do is define a whole new line that goes from the bottom to the top which will probably take us another 18 months to two years to design, which will integrate both.

We will decide whether we want to go with 68030's or 68040's or 486 from INTEL, also busses: VME versus Multibus (we are doing a study right now). For the interim, we will meet the customer commitments. We will probably take the Formula line and extend it from a 68020 to a 68030 to enhance its performance and extend its life. We will take the SCI 3000 which is currently a Multibus I and make it a Multibus II.

We think we have enough product between what Fortune has developed and what SCI has developed to carry us beyond where we are today. And we will continue developing Fortune:Word.

/u/fortune news: What type of development?

Mr. Borelli: For example, a lot more graphics capability added to it over the long term.

/u/fortune news: Well, that is inter-

esting. One clear lack on the Fortune is graphics capability.

Mr. Borelli: Our plan is to keep enhancing Fortune:Word - not only with graphics but with other features that are undergoing a major study right now.

/u/fortune news: As I understand, SCI has the rights to Fortune:Word. Or are you contracting through Tigera?

Mr. Borelli: No, we have the rights to Fortune:Word.

/u/fortune news: Then you have your own people working on it?

Mr. Borelli: Yes, our own people. We are going in a totally independent direction from Tigera.

/u/fortune news: What will your strategy be for marketing your machines? You mentioned before value-added resellers. What will you be doing with the Dealer channel that now exists? Will you expand it and introduce new hardware into it?

Mr. Borelli: Our plans are not to introduce our SCI products into that channel until we are fully sure that we won't confuse the distribution channel.

We have basic commitments on the Fortune products, primarily 4000 and 8000, that we have to finish first.

We are also looking to expand the dealer network from 33 Master Dealers to 70 this year. We are also adding new sales programs and training for our sales people and our dealers.

/u/fortune news: Can you talk about that a minute? I mean, what does that entail? Do you have classes on running an SCI or material that you are sending them?

Mr. Borelli: No, we are actually sending sales management teams, sitting down with dealers and putting new programs in place which we believe will help improve productivity.

/u/fortune news: You said you want to expand the network to 70 dealers by the end of the year. Does that mean December, 1987?

Mr. Borelli: Our fiscal year ends June 30th.

/u/fortune news: Currently, the dealers are geographically constrained - are you considering keeping this type of a system in place?

Mr. Borelli: What we are doing now - which is part of our new sales expansion program - is looking at all those issues. For example, the geographic question, affiliate dealer relationships - to determine what's best for us and what's best for our dealers. Our plan is to grow - I made a statement a while back that our goal is to grow this business two to three times in the next 2 to 3 years - we are focusing on growing and I

UNISOFT INC.

Experts in Computer Consulting - Sales - Service

Presents

FORTUNE SYSTEMS HELP

We Provide:

- Expert Trouble-shooting
- Years of programming and accounting experience
- *Free Initial Consultation*
- Training Available for:
  * BASIC/IDOL/BAS and Multiplan
  * UNIX and Ext. Fortune:Word
- Hardware Repairs & Upgrades
- FORTUNE trained experts at a cost that will surprise you!
- *Trade-Ins Welcome*
- Specializing in:
  * Hard Disk Trade-In
  * Hard Disk Upgrades

Authorized Fortune Systems Repair Depot
Fortune Systems National Value Added Reseller (VAR)
CALL: (303) 671-8600 EMERGENCY HELP: (303) 649-8505

UNISOFT INC. 14100 East Jewell Avenue Suite 27 Aurora, Colorado 80012

think we can achieve this.

/u/fortune news: I would like to ask about new products? What type of new products will SCI be introducing?

Mr. Borelli: As I said, we have workstations and multiuser computers that run both DOS and UNIX. We also have some new applications such as printing and publishing which we are evaluating.

/u/fortune news: Any advance peeks at what you might be doing in the next 3 to 6 months in terms of introduction of new software and hardware.

Mr. Borelli: In the next 6 months we will introduce some 286 and 386 workstation products.

/u/fortune news: Workstations that are good for CAD/CAM applications?

Mr. Borelli: Workstations that run both DOS and UNIX applications including printing and publishing packages. A system might include a laser printer and scanner.

/u/fortune news: That is, of course, interesting to us since we are in that business. Would that be Macintosh type or would it be troff based?

Mr. Borelli: Both troff and postscript.

/u/fortune news: When you start porting software under UNIX V.3 and you basically have 386's and 486's - you will essentially have software that will run on anyone's hardware. How will you differentiate your hardware from other vendors' hardware? Will you try to keep the software proprietary in some sense?

Mr. Borelli: The plan we put into place was to develop an industry-standard state-of-the-art set of products. Standard networking, standard micro to mainframe communications, standard operating systems, standard bus. What we are planning to do is compete on price as well as packaged applications.

/u/fortune news: That is a very different strategy than Fortune is using now. In other words, Fortune never tried to compete on price.

Mr. Borelli: They never had the manufacturing capability.

/u/fortune news: Then what we will find is a line of computers backed up by your manufacturing capability - you will offer excellent software on good hardware at competitive prices.

Mr. Borelli: Yes.

/u/fortune news: What do you mean by "industry standard?" Perhaps I just don't understand the term. Is IBM standard?

Mr. Borelli: IBM is an industry standard from a mainframe point of view. If you want to develop a set of multiuser computers to sell into major accounts either directly or through master dealers, you need to have

NJBC

New Jersey Business Computers

Authorized Fortune Systems Master Dealer

Authorized Fortune Service

Authorized Fortune Training Center

NEW JERSEY BUSINESS COMPUTERS has the Largest Affiliate Dealer Network In the New York and New Jersey Metropolitan Area.

For The Most Reasonably Priced TRAINING, SERVICE and SUPPORT Available.

call

301 Route 17 North, Rutherford, New Jersey, 07070

201-438-1410

communications and networking software that talks to IBM networks and IBM mainframes - SNA, SDLC, LU 6.2, etc. From a Local Area Network point of view, you need to have ETHERNET, STARLAN, etc. Standard buses include Multibus I, Multibus II and VME. Operating systems would include UNIX V.3 and DOS 3.x. That's what I mean by industry standard. By offering industry standard products you provide your dealers access to a wide variety of applications and peripheral controllers.

/u/fortune news: The differentiating factor, then, is the price?

Mr. Borelli: The differentiating factor is competitive price, an excellent word processor in Fortune:Word, communications and networks and vertical applications software packages. In our Peripherals Operation, we manufacture a number of special purpose terminals such as electronic time recorders, factory data collection terminals, ticket printers and these terminals can be packaged with our computers and communications networks into turnkey systems. We have a great deal of capability and flexibility to address our markets. We just need to do it smartly.

/u/fortune news: In general, then, you feel good about the acquisition of Fortune.

Mr. Borelli: We have gone through some transitional pains but we feel positive and we think we can make it work and work well for everyone.□

FORTUNE SYSTEMS
an SCI Systems Company

No matter what your product or service, staying competitive requires control

Let QBC, your company's one source computer supplier, provide your business with solutions to:

Order Entry	Payroll
Accounts Payable	General Ledger
Accounts Receivable	Sales Analysis
Invoicing	Legal Time Billing
Inventory Control	Manufacturing Software
Office Automation	(MRP II Plus)

QBC is your one source for hardware, software, maintenance, customer training and data conversion.

MASTER DISTRIBUTOR FOR UPSTATE NEW YORK AND ERIE, PENNSYLVANIA

QBC

QBC Systems Inc.
375 NORTH FRENCH ROAD, AMHERST, NEW YORK 14120
(716) 691-5201 1-800-443-5770

Tri-Tech Computer Services, Inc.

UNPARALLELED COMPUTER SERVICES

Why Spend More Than Others For Fortune Equipment & Services?

1-800-622-TTCS

(Use Tri-Tech!!)

Pre-XMAS Specials:
Handshake Blazer & Handshake ITE Communications Software
Motherboard Upgrades (XP2MB, SX2MB & SX3.5MB)
70MB Hard Disk Upgrade (Our Favorite)
Additional Ram Memory & Comm-A Boards
Power Supply & Hard Disk Controller Upgrades
Fortune Workstations (New or Used)

TRI-TECH COMPUTER SERVICES, INC.— *Our Performance Is What Separates Us From The Rest. Let Us Perform For You.*

106 Cynthia Drive
Kensington, CT 06037
(203)828-9784
Jim Monterosso & Victoria Galo

P.O. Box 1330
Brookline, MA 02146
(617)965-4453
Norm Westlund & Allison Black

The BASIC Advisor



The BASIC Advisor is brought to us from Ray Wannall. Ray is President of BaSiC Software Corporation which is located in Baltimore, Maryland. This month Ray discusses a Concept Omega conference and the Fourth Generation Language that Concept Omega will introduce.

The following inquiry crossed my desk shortly after the July edition of **/u/fortune *news*** (Volume 4, Number 7) was mailed. It is a follow-up from our friend in the United Kingdom, and I apologize to him for the delayed response. Dealer conferences have a way of throwing disruptive order into my otherwise free and easy life style.

***Question**: Thank you most sincerely for your kind attention with my last letter of the 26th of June. I must admit that I had not expected to see the answer published with such grandeur in **/u/fortune news.** Super!*

There is quite an art to writing long technical descriptions in such a way that solutions can readily be given, and I fear that I may not have given you enough detail in order to solve the problem to which I originally referred.

*I would thank you for your comments regarding adjustment to the header field number 4, "NO OF ELMS", and your reference to increasing one of the four fields relating to the column positions and lengths. I had managed to sort that bit out. (The **IDOL** manual is one of the better manuals I've seen). However, having displayed the new field on the file screen, I was unable to enter data into it, and I subsequently find that the problem is not restricted to just that. Ever since the addition of the extra field, any F4 exit from any of the 13,000 records in the file results in an "I/O ERR 1, RETURN TO CONTINUE" prompt.*

I am certain that this will change the ball game considerably and I apologize for having to trouble you again with a problem that I should have explained more fully the first time.

The only other comments that I can make, that may or may not be of help, are that the original file consisted of a total of 17 fields, and that I had no problem adding two extra fields, one at a time as the specs changed, increasing the 'space' via the file transfer/expand and then using the new fields to add extra data. Only with this last 20th field have I had problems. Once the I/O error appears, it affects all F4 exits of all records of the file, not just on trying to enter data to the 20th field. All I am able to do at this stage is reload the data and old file format with the 19 fields whereupon everything reverts to normal.

*In the meantime, could you please advise me as to how I can find out what changes have been incorporated in the **1.3 IDOL** update? I would propose ordering through a UK based company, providing the update is useful enough and does not cost too much!*

*It is a pity that **IDOL** never caught on over here. It has much easier and far better report generators than the more cumbersome approaches utilized by **INFORMIX**. (Meant in the general sense, since a lot can be done to produce a wide variation of report styles/layouts with **INFORMIX** - once you have the hang of the BOURNE shell and the use of a good editor, like the "sc" contained on the D.C. Grab Bag diskette obtainable c/o **/u/fortune news.**)*

Answer: I must extend my thanks for the plugs on behalf of myself, John-L Johnson of Concept Omega, and Josh Lobel and (recently married, congratulations) Mark Palmerino of **/u/fortune *news***. Keep those cards and letters coming!

There is only one reason you get I/O ERR 1 during **IDOL** file maintenance: your record size is not large enough. When you add a field to an **IDOL** file, you must, as you have surmised, expand the record size with the Transfer/Expand utility. But when expanding it, you must remember to add a byte for a "field terminator" for each new field. The field terminator (hex 8A on Fortune Systems) is used in Business BASIC Direct, Indexed and Serial files to indicate where one field ends and the next one begins. Therefore, if you add a field of, say, 20 characters to a file which contained records of 100 bytes each, you must expand the record size of that file to 121.

When you first define a file data area with **IDOL** you are given a recommended record size which has been calculated by **IDOL**. This size is actually larger, by a byte or two, than is actually needed. I proved this on my system by creating a direct file under **IDOL** with a key size of 2 and two data fields of 20 characters each. The required record size should have

been 45: 2 (key) + 20 (data field 1) + 20 (data field 2) + 3 (required field separators). But when I went into the **DEFINE A FILE DATA AREA** utility, the computer told me the record size was 46. Your manual advises you to accept the calculated size "to use disk space efficiently". (I personally think that it would be more efficient to calculate the record size properly, but who am I to question the great minds that wrote this thing?) I suspect that when you originally created your file, you were given two "expansion" bytes which were used for the field terminators for fields 18 and 19 respectively. By the time you added field 20, you were out of courtesy bytes.

So why, you ask, were you able to bring an old record onto the screen in file maintenance only to get I/O ERR 1 when you hit F4? In Business BASIC files the element lengths may be variable, but when using **IDOL** the lengths of the elements are fixed. Here's a little program which demonstrates how BASIC handles file elements using the file layout I just defined:

```
10 BEGIN
20 ERASE "MYFILE",ERR=21
30 DIRECT "MYFILE",2,1,45,0,0
40 OPEN (1)"MYFILE"
50 WRITE (1,KEY="XX")"XX","FIELD 1","FIELD 2"
60 READ (1,KEY="XX")A$,B$,C$
70 PRINT "KEY SIZE = ",LEN(A$)
80 PRINT "FIELD 1 SIZE = ",LEN(B$)
90 PRINT "FIELD 2 SIZE = ",LEN(C$)
```

When run, this program shows that the size of the key field is 2 and `FIELD 1` and `FIELD 2` are each 7 characters long. No additional spaces were written into the file. But if we define the file under **IDOL** and add the record under **IDOL** file maintenance, the lengths of `FIELD 1` and `FIELD 2` are expanded to 20 characters. Instead of a variable list, **IDOL** uses a dimensioned string (`E$`) for reading from and writing to a file. This pads all of the elements with spaces to their maximum sizes. I guess **IDOL** figures that if a field has been defined at 20 characters, then, by golly, it is going to use all of them! (How's that for efficient use of space?)

I don't think it was necessary for you to reload the data into your file after you got the I/O ERR 1. A simple expansion of the record size should have fixed it. What probably happened is that after you added the new field definition, you entered file maintenance in the CHANGE mode. The record was read okay because its original length was within the defined limits, but an additional field of spaces plus one field terminator were added at the end of the record to account for new field 20. When you pressed F4 to exit, IDOL tried to write the expanded record back into the file, but it was too big. The error halted the write, so the old data should not have been destroyed.

It is actually amazing that **Fortune Systems** even provided the **1.3 IDOL** updates. Shortly after the diskette was released **Fortune** discontinued supporting **IDOL**, claiming it was an SMC product. The update diskette was primarily for **BAS** modifications and fixed all of the bugs defined through technical tip number 93. I cannot say specifically which problems were corrected with the **IDOL** updates, but I can provide you with a list of the diskette files which pertain to **IDOL**.

General file maintenance programs which are replaced include `CUTFA0`, `CUTFA3`, `CUTFA7`, `CUTFAN`, `CUTFAR`, `CUTGB0`, `CUTUI0`, `CUTUI1`, and `CUTUJB`. Programs involving the report generator are `CUTFIG`, `CUTFIH`, and `CUTLA0`. The rest of the **IDOL** programs are:

CUTDE1 - General data entry module
CUTDS3 & CUTDSB - Define a data entry screen
CUTERR - For handling untrapped program exceptions
CUTPWD - Set up operator permissions
CUTRF0 - Print selector screens
CUTRJ0 - Define selector screens
CUTRM0 - Print data entry screens
CUTRQ0 - Print operator statistics
CUTRR0 - Print the disk storage status report
CUTSA0 & CUTSA1 - Main selector driver
CUTUH1 - Provides help for selectors
CUTUM0 - Build selector load modules
CUTUS0 - Reconstruct cross reference files
DADSA3 - Print element documentation index
DOL - Console mode exit program

The diskette replaces `*DE112`, the data entry screen for the **Reconstruct Cross Reference Files** program, and `*FM031`, the file maintenance screen for the **File Information Records** entry in `CCNVZ` (file number 31). New entries for `CGLBD` (**The Global Element Dictionary**) include several for file 31, mostly of the "NOT USED" variety, and one for `UBSQ-A`, file number 1.

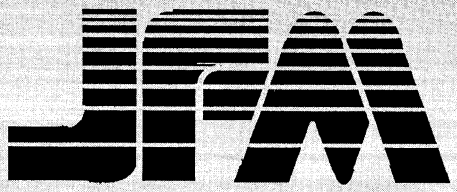
Finally, the following BASIC Utilities are replaced:

*BPSD - Build Fileset List
*CPSD - Compare Programs
*DPSD - Rename Programs/Files
*RPSD & RPSD1 - Reconstruct Direct Files

I should mention that `*RPSD` and `*RSPD1` are obsolete: they are not included with **Business BASIC 6.5.12**.

I hope we have finally addressed all of your questions. If not, please feel free to write again. Letters and calls from our readers are my bread and butter.

Are you a BASIC/BAS/IDOL support company? If so, we would like to include you in our third annual support company listing in the January, 1988, issue of /u/fortune news. Please call me at (301) 448-9460 and give me your company name, contact person's name, city, phone number and billing rate. A big response means less yakking from me in January, and that should inspire everyone! □



TRADE - UPGRADE - EXPAND
Your FORTUNE SYSTEM

Processors:
- *Fortune 32:16 SX*
- *Fortune Formula 4000*
- *Fortune Formula 8000*
- *Fortune Multiplier (Run multi-user For:Pro & Fortune applications on a PC)*

Hard Disks:
- *30MB*
- *45MB*
- *70MB*
- *85MB*
- *145MB*

Tape Drives:
- *20MB External*
- *60MB External*
- *60MB Internal*

Memory:
- *256KB*
- *512KB*
- *1024KB*
- *4096KB*

Controllers:
- *Comm-A 4 Ports*
- *Comm-A 6 Ports*
- *Bit-Mapped Graphics Controller*
- *Fortune:Link Controller*

BONUS - BONUS - BONUS

All Fortune Licensed Software Upgrades and/or Replacements
At No Additional Charge with Processor Upgrade/Trade!

We can perform these trades and/or upgrades within 24 hours of receiving your system. In fact, if you ship your system overnight on a Friday for Saturday delivery, we will have it back to you on Monday morning, all ready to plug in and run; not losing the use of your system for a single work day. And, we will issue you a full 90-day warranty on the entire system.

JFM provides services to Fortune Systems owners nationwide either direct or through our authorized Affiliated Dealers. If you haven't used our services, I invite you to discover JFM; you will be glad you did.

Sincerely,

Morris Frydman
President

Talk to us, then talk to our clients

JFM
BUSINESS
SYSTEMS
The Merchandise Mart
Lobby Suite 144
Chicago, Illinois 60654
Call (312) 644-0870

Authorized Fortune/SCI Systems Master Dealer

Affordable laser printing for Fortune

Laser Printing! Kiss the expense good-bye.

QMS — KISS laser printer $1995.00

- cost per page 1.4¢.
- 9 resident fonts
- Print cap and print utility included.
- add-on sheet feeder provides total of 3 trays and envelope feeder.

THE GOOD FORTUNE TRADER

QMS - KISS from
The Good Fortune Trader
1 Year Warranty

a division of
Computer Services
of Atlanta
(404) 447-4151

5825 Live Oak Parkway, Suite 2-C, Norcross, Georgia 30093

DO YOU KNOW...

there is an organization that has a thorough knowledge of

FORTUNE BAS
MULTIPLAN IDOL

We have been providing widespread support to our Fortune clients for more than 4 years.

If you need:

- training
- support
- maximum computer performance
- peace of mind relating to processing
- customization
- application results
- minimum errors and confusion
- better ideas on how to increase the power of your Fortune

Call Steve Rosenfeld at (314) 731-3636

SUPERIOR COMPUTER SYSTEMS, INC.
Making computers work for you

P.O. Drawer T
Florissant, MO 63031

/u/fortune news
73 Jason Street
Arlington, MA 02174

Address Correction Requested

/u/fortune* news

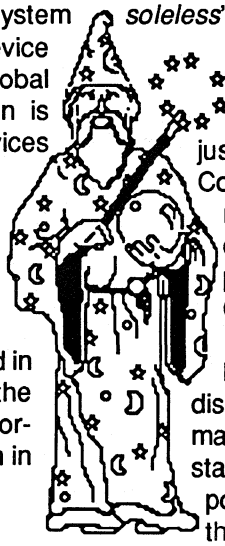
The Newsletter for Users of Fortune/SCI Computers | December 1987/Volume 4 Number 12

Read all about devices in...

System Administration: Part 8

by Dave Kloes

It seems like we have been talking about the **/etc** directory forever, but as you may have noticed by now, it really is the heart of For:Pro. Last time, we talked about the "group" and "passwd" files in that directory. In this issue, we will talk about the "ttys", "ttytype", "devtype", "motd", "disktab" and "fstab" files, which are also located in the **/etc** directory.



Have you ever used "S2" (System Management), #39 (Define device connections) through the Global Menu System? This selection is used to tell For:Pro about devices that are connected to the ports on the back of your system. These devices may be terminals, modems, printers, etc. The information that you enter through the Global Menu about devices on your system is stored in the **/etc/devtype** file (Note: the **devtype** file is unique to the Fortune system). This file is shown in Figure 1.

Looks like a mess at first glance, but let's take a look at each column. Column 1 is the name of the device from the **/dev** file. If you've forgotten what device names are and their relationship to the Comm A board, now is a good time to go back and review the material in our previous articles. The "console" is the terminal that is your main system terminal. "tty01" is the printer port that is located at the back of your system in the center. (Note: Some systems are now set up as "*consoleless*" systems and this port is used for the console. While it is often referred to as the "printer port", it can be used just like any other port on a Comm A board - for a printer, modem or terminal). The rest of the "tty" entries are serial ports associated with the Comm A boards.

Before we continue with our discussion of this file, let us make sure there is a clear understanding about the term "serial port". On computer systems, there are two kinds of ports that are used for communication devices -

See **Administration**, Page 12

Figure 1.

```
# cat /etc/devtype
console   FT      T   0   n   Master Work Station   1
tty01     PLAIN2  P   1   m   Anadex                1
tty02     FT      T   1   m   June                  1
tty03     FT      C   1   k   Modem-out             1
tty04     FT      T   2   i   Modem-in              1
```

Notices and Attractions

We are now in a NEW location!

Please note that our new address and phone number is:

The Cambridge Consortium, Inc.
225 Crescent Street
Waltham, MA 02154
617 894-6900

* * * * *

We will publish Fortune/SCI's Software Vendor Catalog in the January and February issues.

In This Issue...

Index of the contents of /u/fortune news for 1987...**Page 15.**

News From Fortune/SCI...**Page 3.**

Sorting the Customer Master File in the BASIC Advisor...**Page 5.**

Money and Time Savers...**Page 8.**

Do you leave your system on 24 hours a day?...**Page 9.**

Are you running out of space on your hard disk?...**Page 11.**

Kep up with System Administration, Part 8...**Page 1.**

*/u/fortune *news* is published by The Cambridge Consortium, Inc. It is not affiliated with Fortune Systems/SCI Technology, Inc..

/u/fortune *news*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz

Technical Editor
Scott Bradner

Advertisting Manager
Mark Palmerino

/u/fortune *news* is published 12 times per year by The Cambridge Consortium, Inc. Neither **/u/fortune** *news* nor The Cambridge Consortium are affiliated with Fortune Systems/SCI Technology, Inc.

Subscriptions cost $48 for one year within the U.S.A. For foreign subscriptions, add $27 each year for added postage and handling costs.

Back issues are offered at:

Vol 1, Numbers 1 - 3	$1/each
Vol 1, Number 4	$.50/each
Vol 1, Numbers 5 - 8	$3/each
Vol 2, Numbers 1 - 7	$3/each
Vol 3, Numbers 1 - 12	$4/each
Vol 4, Numbers 1 -	$4/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $10 each. Please add $5 per order for foreign destinations.

Although every attempt is made to insure that our contents are error-free and without risk, The Cambridge Consortium is not responsible for losses that occur due to correct or incorrect interpretation of **/u/fortune** *news*.

For more information, call or write:

/u/fortune *news*
225 Crescent Street
Waltham, MA 02154
617 894-6900

The text for this newsletter was prepared using a Fortune 32:16 for text input. The information was then transferred to an Apple Macintosh Plus with kermit and formatted for typesetting using the Aldus Pagemaker program.

UNIX™ is a trademark of Bell Laboratories, Inc. Fortune 32:16™, Fortune:Word™ and FOR:PRO™ are trademarks of Fortune Systems/SCI Technology, Inc. Multiplan™ and MS-DOS™ are trademarks of Microsoft. IBM-PC™ is a trademark of IBM. Macintosh Plus™ is a trademark of Apple Computer. Pagemaker™ is a trademark of Aldus Corporation.

©1987 by the Cambridge Consortium, Inc. 12/87

From the Editors...

Back and To The Future

Well, here it is again, the last issue of the year. Generally, this is a time for both reflection and for looking forward to a new and exciting year. We wish to do a little of both in this column.

This year we have seen quite a few changes in Fortune, the company. A metamorphosis from Fortune Systems Corporation to Fortune/SCI has taken place. Naturally, Fortune/SCI is still young and with youth there is often optimism. So far, we are optimistic, too. SCI will bring an established manufacturing capability to the Fortune and is planning to add both hardware and software to the Fortune line. As **Ron Borelli** discussed with us, they are committed to this line and want to see it be very successful. Thus, 1988 should prove to be *very* exciting with regard to new hardware and software.

Looking forward to the near future, we hope you will be anticipating the January and February issues of **/u/fortune *news***. In those issues, we will print Fortune/SCI's **Software Vendor Catalog**. This lists all the current vendors who provide software that will run on the computer. Thus, these two issues should be kept handy as *reference material* for when you need to explore your software needs.

Looking back, for a moment, we want to draw your attention to the *index* of **/u/fortune *news*** we have published in this issue. This, of course, is provided as a service to you, our readers, so that you can find information regarding important topics with ease. But it also serves as a map of sorts that charts our progress over the past year.

Speaking of a map, we are now conducting business in a **new location**. We have moved into new offices as of the beginning of the year (which is the main reason this issue is somewhat late). Please take note of our new address: **225 Crescent Street, Waltham, MA 02154 (Phone: 617 894-6900).**

We are extremely happy with our new setting and are looking forward to another successful year with **/u/fortune *news***. We hope that you will remain a happy subscriber and user of Fortune/SCI equipment. As always, we enjoy hearing from you. *Let us know how we can better serve you!*

News From Fortune

Coincident with its move to new headquarters, Fortune/SCI today announced that **Daniel Younkins II**, 53, has been appointed general manager of the division that manufactures and marketes all of its UNIX-based supermicrocomputers, an operation of SCI Systems, Inc., Huntsville AL, since last July.

Younkins, a sales executive with broad international private and public sector experience, reports to **Ronald F. Borelli**, senior vice president of SCI's Computer Division. As general manager, Younkins succeeds Robert A. Davis, who remains chairman of Fortune Systems International, the Monaco-based company which holds exclusive distribution rights to the Fortune and SCI supermicrocomputer product lines outside the Western Hemisphere.

Since January of 1986, Younkins has headed up the Washington DC operations of SCI. He had earlier been an independent marketing consultant and an officer of the Overseas Private Investment Corporation, a Federal agency which provided financing and political risk insurance for American companies with overseas business interests, in both the Carter and Reagan Administrations.

Younkins has also been president of American Fine Wire Corporation, a manufacturer of computer and instrumentation products, and vice president of the energy systems division of Wheelabrator-Frye, Inc., with responsibilities for all its public and private sector fossil fuel programs. For the previous 13 years, he held sales and management positions domestically and in Europe and Africa for IBM and Computer Sciences Corporation.

A 1956 business administration undergraduate of Emory University in Atlanta, Younkins completed graduate studies in international business at Florida State University, Tallahassee, during 1968. He is past president of both the American International Businessmen's Association and of Sales and Marketing Executives-International.

Of his new assignment, Younkins notes, "It'll be a challenge to blend the SCI style with the Silicon Valley culture, but I expect that assimilation to result in Fortune/SCI being uniquely positioned in some very promising markets."

Fortune/SCI's new headquarters' address is 75 Shoreway Drive, San Carlos CA 94070. Its telephone number, 415/593-9000, will remain the same. The move reflects Fortune/SCI's reduced need for space, given the transfer of manufacturing functions for the Fortune Formula 4000 and 8000 to Huntsville.

****

Advanced Industry Systems, Inc. (AIS) of Florida recently announced that it has sold 10 Fortune/SCI systems, complete with the Wallaby resident medical software package, during the period from July through September 1987.

Fortune/SCI is a developer and marketer of UNIX-based supermicrocomputer systems, including hardware and software, for the office automation and select vertical markets.

Joel Ingram, president of AIS, describes the Wallaby resident medical package by noting "...it can provide automated insurance and pageless claims as a stand-alone module; it enables a practice to start small and later expand; and it's flexible in its ability to be custom-tailored to a specific type of medical practice."

In conjunction with the health care industry, AIS has developed its exclusive **Wandasist** capability. Explains Ingram, "This 'magic wand' allows a super bill to be scanned in 12 seconds. And complete patient checkout, including forms printing, is handled in less than two minutes."

Ingram forecasts that his firm's sales of Fortune/SCI systems will reach $10 million by the end of its three year contract with the Company.

A combination of factors explains AIS's unusually high sales record. "We understand the needs and applications of the health care industry, since it is a solution business," says Ingram. "We also enjoy a large base of physicians in the western Florida area. Thirdly, AIS has a good reputation for service and support. And lastly, but certainly not least in importance, Fortune/SCI's supermicrocomputers are head and shoulders above its competition."

AIS, one of Fortune/SCI's 29 master dealers, joined the Company's reseller network in April 1987. Specializing in the medical, dental and legal vertical markets, AIS offers its customers the full range of Fortune/SCI software and hardware products, including the Fortune Formula 4000, a multi-user system based on the MC68020 microprocessor and designed for workgroups of two to 20 users.

"AIS' outstanding performance at the start of its contract is only the beginning of a successful three years," notes Rudy Prater, Fortune/SCI's district manager in Atlanta (GA). "I believe AIS will also achieve similar success within the dental, point of sale and legal fields." □

What An Opportunity



An opportunity like this doesn't happen very often.

Micro Resources is in a fix. The region's leading Fortune/SCI supplier's stock room is jammed with top-notch Fortune/SCI inventory — with more on the way.

So we're dropping prices to levels you probably thought you'd never see. Grab your wish list. This is a brief opportunity to upgrade your system you simply can't afford to pass up.

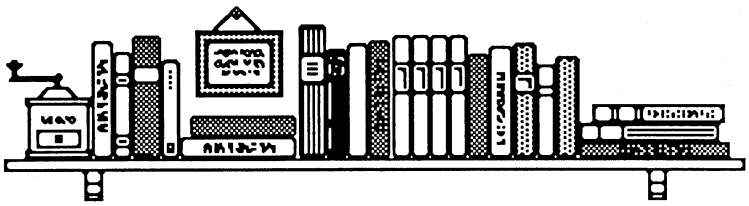
If you need more memory, additional terminals or faster motherboards, now's definitely the time to buy. But don't wait too long — **quantities are limited.**

Increasing your Fortune/SCI system's productivity isn't very difficult. And Micro Resources just made it a lot easier. Call Chuck Kennedy, 766-2335, today — before it's too late.

Micro Resources

255 Bradenton Ave., Dublin, Ohio 43017 614 • 766 • 2335

The BASIC Advisor



The BASIC Advisor is brought to us from Ray Wannall. Ray is President of BaSiC Software Corporation which is located in Baltimore, Maryland.

So far I have received one call from a company to be included in our third annual support company listing. I certainly hope that this is because everyone is too busy to take on new inquiries. But for those of you who didn't see my call in the November article, I will make my appeal one more time:

Are you a BAS, BASIC and/or IDOL support company? Would you like to be included in our third annual support company listing? If so, I will need your company name, contact person's name, city, phone number and billing rate. Please call me at (301) 448-9460 or write to me at BaSiC Software Corporation, 5201 Powhatan Street, Baltimore, MD, 21207. Because of a change in printing schedules, it will be necessary to delay the support company listing until February, so everyone should have ample time to respond.

Question: *When I copied a Business BASIC program which was created on a different Fortune system from floppy diskette into the /b/BAS/USR directory, I was unable to load or run the program once I was in BASIC. I kept getting an error 19. When I examined the program in the DISPLAY FILE PARAMETERS utility, I found it was an "Uninitialized Program". How can I access the program?*

Answer: Use the "FILE TRANSFER AND EXPAND" utility. Press the RETURN key to accept default values for all questions, and RETURN again to replace the file after it has been copied. The utility will initialize the program for you. I find this method much easier than traipsing through UNIX looking for permissions to change.

Question: *Would you please explain the mechanics of sorting information in a file for a printout? For example, I would like to get a listing of customers from the Customer Master File (CCSMS, file number 110) printed in order of Sales Rep ID, which is field number 29 in the file. I know it is done automatically in the IDOL report generator, but I am curious about how you do the sort using BASIC.*

Answer: You have certainly chosen a complicated file to be used as an example. I think that if I create a simple version of the Customer Master we can demonstrate the principles of sorting for you. Here are a few key elements from CCSMS:

VARIABLE	ELEMENT NAME	LENGTH
A1$	1-Customer Number	6
B1$	2 Customer Name	30
C1$	3 Sales Rep ID	2
A1	4 YTD Sales Amt	7

As is the case with CCSMS, our example file is "keyed" with the Customer Number, or A1$ (meaning that our file is stored in order of customer numbers). Just for yucks, let's call our file "CCUST". In order to see the records of CCUST in order of Sales Rep ID (C1$), we must create a temporary sort file to rearrange the way records are stored. The BAS applications supply each terminal with its own sort file for such purposes. These files are called "*SRTn", where "Tn" represents the terminal number using the file. You can find the number of the terminal you are using with the command:

```
PRINT FID(0)    'CR'
```

(The computer responds with T0 or T2 or whatever.) The *SRTn files are normally in the /b/BDATA/USR subdirectory which is disk 2 in BAS. In order to determine how large to make our temporary sort file we must know how many records are in use in the main source file, or CCUST. This can be found by examining the file parameters with the *FPSD - DISPLAY FILE PARAMETERS utility. Let us suppose that CCUST is defined as containing 10,000 records of which 5000 are currently in use. Next we must decide how large each record in *SRTn

needs to be. What we will be doing is creating a sort file (keys only) which will hold the Sales Rep ID (2 characters) plus the Customer Number (6 characters). Our key size, therefore, will be 8. Now we are ready to conduct step one of a three-step process in our program:

```
100 LET Q9$="*SR"+FID (0)
120 CLOSE (2,ERR=121)
140 ERASE Q9$,ERR=141
160 SORT Q9$,8,5000,2,0
180 OPEN (2)Q9$
```

At statement 100, we are coupling our terminal number, FID (0), with the string "*SR" and placing the resulting value in the variable Q9$. In statement 160, Q9$ is the file name, 8 is the key size, 5000 is the total number of records required, 2 is the disk number (subdirectory) in which the file is to be defined and 0 represents the sector (always zero on Fortune Systems).

In step two, we will read sequentially through CCUST and write sort keys into *SRTn:

```
200 CLOSE (1); OPEN (1)"CCUST"
220 LET K$=KEY(1,END=300)
240 READ (1,KEY=K$)A1$,B1$,C1$,A1
260 WRITE (2,KEY=C1$+A1$)
280 GOTO 220
```

When this routine has run its course, the program has created a file of keys in order of Sales Rep ID (C1$) plus Customer Number (A1$). Our final step involves rereading CCUST:

```
 900 READ (2,KEY="",DOM=901)
1000 LET K$=KEY(2,END=9000)
1020 READ (2)
1040 READ (1,KEY=K$(3))A1$,B1$,C1$,A1
1060 REM "DO YOUR PRINTING STUFF"
1080 GOTO 1000

9000 REM "PROGRAM EXIT"
```

Statement 900 is used to set the "pointer" for channel 2 to the first record in the file. (Heaven only knows which record is next after the sort routine is conducted). At statement 1040, we are reading CCUST with bytes 3 through 8 from the sort file key. These characters, of course, represent the Customer Number, i.e. the key to CCUST. We could just as easily have said READ (1,KEY=K$(3,6)), but I prefer to be brief when it comes to entering code. Both versions of the statement will work.

Now let's get a little fancier. Suppose you want a Customer Ranking List - a list of customers in order of total yearly purchases from highest to lowest. Obviously we must use field 4, YTD Sales Amount, as a part of our sort file key. Since the length of the YTD Sales Amount is seven, our sort key size must be 13 (7 plus the Customer Number length of 6). We must first change the way *SRTn is defined in step one, substituting 13 for 8 in the key size:

```
160 SORT Q9$,13,5000,2,0
```

When we use numbers in a sort key, it is necessary to "mask" them so that they will all be the same size. In our file, A1 with a length of 7 could be any number between 0 and 9,999,999. (Commas are used here for clarity: they are not stored in the file). The number 1 has a length of one, while 999 has a length of three. We can give them equal length by placing them into alphanumeric format using the STRing function and a mask:

```
LET A$=STR(A1:"0000000")
```

When A1 has a value of 1, A$ becomes 0000001. Similarly if A1 equals 999, A$ is 0000999. In both cases A$ has a length of seven. But look what happens when we mask our numbers and write to our sort file:

```
260 WRITE (2,KEY=STR(A1:"0000000")+A1$)
```

Once the sort file has been constructed, the records are in order of lowest to highest. (1 is sorted before 999 in the file). This means that the first customer read from CCUST will be the one with the lowest sales for the year. We prefer our listing to show the customer with the highest sales volume first. To reverse the order of the records (putting the customer with the highest sales sequentially first in the file), we use what some of us call a "nines complement". This involves subtracting 9,999,999 from the original value of A1 and writing the ABSolute value of the remainder into our sort file in place of A1. By using the ABSolute value, we eliminate the negative aspect of the number:

```
260 WRITE (2,KEY=STR(ABS(A1-9999999:"0000000"))+A1$)
```

Now that our sort file is in the proper order, we can assign a rank to each customer as the records are being read. In this example, the variable "R" represents each customer's rank:

```
 900 READ (2,KEY="",DOM=901)
 920 LET R=0
1000 LET K$=KEY(2,END=9000),R=R+1
1020 READ (2)
1040 READ (1,KEY=K$(3))A1$,B1$,C1$,A1
1060 REM "DO YOUR PRINTING STUFF"
1080 GOTO 1000
```

Notice how we set the value of R to zero at 920 and increased its value by one each time we found the next key to our sort file at statement 1000. By the time statement 1060 is encountered, R represents the rank of customer A1$ who has a total of A1 dollars worth of sales so far this year. The only thing left for you to do is send the information to the printer.□

** *See the NEW products W-2 Link and 1099 Link* **

SOFTWARE FOR THE FORTUNE 32:16 AND FORMULA

NEW PRODUCT!! W-2 LINK ($495.00) - this product is designed for companies with over 250 employess who must report W-2 payroll information to the Social Security Administration (IRS) via magnetic tape or diskette. The product automatically takes information in the Business Accounting System (BAS) Payroll module and puts it onto diskette in the format required by the IRS. Available for PAC software also.

NEW PRODUCT!! 1099 LINK ($295.00) - this product produces Form 1099's required for submission to the IRS and is an extension to the standard BAS Payroll module.

DOS LINK ($495.00) - provides easy-to-use menus and programs that transfer data between any PC (including Laptops) running DOS and the Fortune system. Here are just a few examples of how this product can be used:

- Transfer Multiplan or Fortune:Word Documents from or to your DOS system.
- Transfer other ASCII or binary files.
- Use the PC or Laptop as a remote terminal on the Fortune system.
- Transfer THOROUGHBRED (SMC) BASIC data that has been extracted using one of our other products

Documentation and software for both the DOS and Fortune system are provided. Data may be transferred via a direct connection (9600 baud) between the two computers or via modem (at modem speed). You may also order additional hardware for your PC that may be required to accomodate the transfer.

If you are interested in purchasing a Laptop computer that can be used with this product for people on the road or at home, give us a call!! We also provide consulting services for interfacing your DOS system to the Fortune ($50/hr).

MULTIPLAN LINK - can pass data from BAS/IDOL/BASIC files to Multiplan and back. For example, it could be used to send data from any of the BAS files (Chart of Accounts, Customer Master, Inventory Master, etc). Allows selection of fields to be sent and provides logical retrieval and key range selection. $295.00

RECORDS PROCESSING LINK - can pass data from BAS/IDOL/BASIC files to Records Processing and back. For example, it could be used to take customer address data from the Customer Master and will automatically make the Records Processing List document for mail/merge. Allows selection of fields to be sent and provides logical retrieval and key range selection. $295.00

ASCII LINK - can pass data from BAS/IDOL/BASIC files to ASCII and back. Use to send data between machines or databases. For example, it could be used to send data from any of the BAS files (Customer Master, Inventory Master, etc). Allows selection of fields to be send and provides logical retrieval and key range selection. $295.00

TERMINAL/PRINTER LINK - provides capability to print data on a printer connected to a workstation. Can print BAS/IDOL/BASIC reports, Multiplan and Fortune:Word documents. BAS/IDOL reports can also be viewed before printing, converted to Fortune:Word format; accumulated for mass printing; backed up to diskette; printed on any system printer. An option to print BAS/IDOL reports on a LASER PRINTER is $50 extra. Many other options are included. $195.00 (Fortune:Word printing on a printer connected to a workstation is $50 extra)

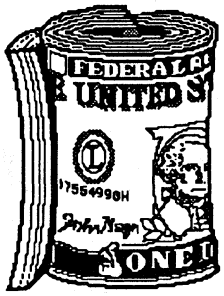
TELEPHONE LINK - on line telephone message, telephone directory and inter-office mailbox/menu system. Telephone message entry resembles typical pink slip. Telephone directory can search by name or company. Mailbox/message system can send and receive Fortune:Word documents or Multiplan documents. Does much more! Give us a call. Written in Business Basic. $295.00

CALCULATION LINK - provides programs for amortization, depreciation, loan repayment, averaging (with graph), linear correlation and breakeven analysis. Written in Business Basic. $95.00

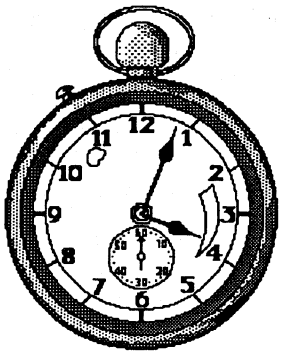
KOMPACT PERSONNEL ACCOUNTING - provides for data capture of personal, wage/salary, job, education, salary/wage, appraisal training, dental, medical and life insurance data. Includes over 80 pre-defined reports/worksheets. Written in Business Basic/IDOL. $995.00

WESTERN LINK - provides instruction on how to use your Fortune for sending/receiving telex messages using Western Union's Easylink. Can replace your telex! Fortune ITE or Handshake software required. $95.00

UNI-KOMP 9803 Rocktree Houston, Texas 77040 (713) 937-7782



Money and Time Savers



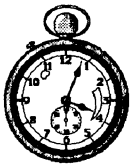
Starting with this issue, we will periodically publish, in the Money and Time Savers column, hints, tricks and just good sound advice for saving both time and money.

Don't Short Out Your Computer!



Dave Vantine of Vantine Business Systems wrote to us with the following warning. Manuals, notebooks, and other heavy objects should NOT be stacked on top of your CPU. The weight of the books causes the cover of the CPU to bend into the power supply, thereby shorting it out. This situation is an important one to be aware of since it is probably not covered by your Maintenance Agreement.

Run UNIX Commands From Inside Fortune:Word



It is possible to issue a UNIX command from inside a Fortune:Word document. This is done by pressing the **F13** Function key and then hitting an exclamation point. Any UNIX command can follow the **!**. For instance, **! date** will provide the time. Another use would be to enter Multiplan while inside Fortune:Word. Type **F13 ! master** and this will take you directly into the Multiplan Menu. The nice thing about this maneuver is that when you leave Multiplan, you return to the document you were editing in Fortune:Word.

Print Out Your Conf Block



Our hard disk crashed recently, and it turned out that the cause of the problem was that the configuration block had gotten scrambled. We knew we had a problem when we got a message that said "Drive 0 has gone bad, will not accept any more requests." The configuration block is located right at the head of the disk and it contains crucial information about the way the disk is formatted, which blocks aren't good, how much swap space you have, etc. Without it, the disk can't work. Fortunately, it is possible to recreate a configuration block — if you know what information it contained. There is a simple way to make a record of this, and we strongly encourage everyone to do it. All you have to do is turn on your printer, get into UNIX (**!sh** from the Global Menu), and type the following:

```
$ rdconf /dev/hd00 | lpr
```

This will start your printer going. If you have more than one disk, use the same command but change hd00 to hd10, hd20, etc.

If you have more than one disk, it is probably worthwhile to print out one other file as well. Type:

```
lpr /etc/fstab
```

This file contains other information about how the system is using your disks. Store these printouts in a safe place — they could save you a lot of time and money in the future.

We asked the Experts...

Do You Leave Your System Running 24 Hours a Day?

We read recently that some computer dealers suggest that it is best to leave your Fortune computer running 24 hours a day. The basic theory behind this suggestion is that each time you turn on your system, it is "shocked." You must use the switch, then a surge of current hits every component in the system (e.g., power supply, chips, etc.) and your hard disk(s) must spin up. Presumably, this represents wear and tear on your computer that could be avoided if it were never started - in other words, if it were kept running constantly.

This is an interesting suggestion and we wanted to know what some dealers thought about it and what their recommendations were. We also talked to Fortune/SCI and have their thoughts on the subject as well.

We contacted the following people:

Elizabeth Muth of Fortune/SCI

Morris Frydman of JFM in Chicago, IL

Dave Kloes of Uni-Komp in Houston, TX

Kieth Lamb of Micro-Resources in Dublin, OH

Guy Washburn of Vantine Business Computers in Waltham, MA

Mike Mead of UniSoft in Aurora, CO

Generally, none of these experts suggested leaving your computer on 24 hours unequivocally. As a rule, they all felt that there are costs and benefits of leaving your Fortune running constantly. We will discuss these costs and benefits.

Most of the experts contacted agreed with the idea that, in general, there is more stress put on the system when it is started. Thus, if one could avoid starting the system, there would be less wear and tear.

Elizabeth Muth of Fortune/SCI informed us that the expected life cycle of the semi-conductor components of the computer is approximately five years. Technical studies seem to indicate that this life would be shortened if the system is shutdown every day. However, she did say that there would be no shortening of the life expectancy if the system were kept running all week and shut down on weekends.

Caveats aside, all the experts said they generally leave their systems on. They have found that fan and other component failures are relatively rare.

The experts also gave other reasons why it may be important to leave a system running round the clock. For example, if your system is used from remote locations by modems, then you must keep it running. This suggests, then, that the *nature and needs* of your business may dictate whether the computer should stay on.

The Costs of Running

Almost with one voice our experts pointed to the possibility of *component failures* when no one is around. For example, should the cooling fan malfunction at 3 A.M. when no one is around, you would find, as Kieth Lamb colorfully put it, that your system committed suicide. Dave Kloes related a story concerning the COMM A board in one of his computers that died a fiery death when he wasn't there.

The experts also said that these problems are generally started by "power problems." That is, the power coming to your computer is extremely important for the smooth functioning of your unit. Many of the experts suggested not leaving the system unattended without a decent Uninterruptible Power Supply (UPS).

We learned that the older Fortunes are more susceptible than some of the newer models because they lack any type of heat sensors connected to the power supplies. Some newer Fortunes are now equipped with such sensors which protect the computer from "committing suicide." If your computer is equipped with Fortune's 300 Watt power supply, then it is protected with heat and air-flow sensors.

Mike Mead pointed out that although you may put less stress on the system by keeping it on, you are, however,

applying other stresses. For example, there are moving parts in disk drives that will probably wear out faster if kept running constantly.

Several of the experts suggested the following strategy: Keep your system running all week, but shut it down on the weekend.

One other thing to keep in mind is that although you may keep your system running constantly, it is not a good idea to leave your terminals on constantly. Everyone suggests either turning them off when not in use, or at least turning the brightness down. This will protect the screen from getting "burned in." Dave Vantine reminded us that newer FIS 1000's have automatic screen blanking which makes this unnecessary.

Conclusions

We have come to some clear conclusions with regard to this matter. First, there is no one single answer to the problem. You must understand that there are both costs and benefits to leaving your system running all the time or shutting it down every night. You would probably do best to consult with your dealer on this matter.

Second, caveats aside, all the experts said they generally leave their systems on. They have found that fan and other component failures are relatively rare. We, at **/u/fortune *news***, have kept our systems going constantly and have not had any problems. However, just because this user or that user hasn't had problems, doesn't mean that you won't! The problems, such as a fan failure, are rare but then DO happen sometimes.

Third, you should at least turn the brightness down on your terminals when they are not being used. The Fortune 1000's have a screen saver feature built in so this suggestion is less critical for these particular devices.

Fourth, there may good reasons to shut your computer down once a week. Shutting down the system has the effect of "house cleaning." Any defunct processes and any other leftovers from software will be straightened out. If not needed over the weekend, then this may be a good time to shut it down.

Fifth, be careful of your power. Surge protectors and UPS's are good investments. We also shut our systems down during storms when brown outs and power outages are more likely. These are the real culprits when it comes to computer problems. □

Happy New Year From...

Tri-Tech Computer Services, Inc.

UNPARALLELED COMPUTER SERVICES

Why Spend More Than Others For Fortune Equipment & Services?

1-800-622-TTCS

(Use Tri-Tech!!)

Specials:
Motherboard Upgrades (XP2MB, SX2MB & SX3.5MB)
70MB Hard Disk Upgrade
Fortune Workstations (New or Used)
Fortune Formula 4000 and 8000 systems

TRI-TECH COMPUTER SERVICES, INC.— *Our Performance Is What Separates Us From The Rest. Let Us Perform For You.*

106 Cynthia Drive
Kensington, CT 06037
(203) 828-9784
Jim Monterosso & Victoria Galo

P.O. Box 1330
Brookline, MA 02146
(617) 965-4453
Norm Westlund & Allison Black

Free Up Disk Space

Here is a time saver in the form of a shellscript. Sometimes, it becomes necessary to free up space on your hard disk. When this happens, the tendency is to hunt *haphazardly* around the system for large files that are candidates for deletion. This is often an inefficient way of clearing up space.

A more efficient method would be to use the **find** command to list out files that are big and old. The **find** command is well suited to this task and will list the names of files that meet certain criteria.

Figure 1 shows a shellscript that uses the **find** command to produce a list of files that are bigger than one parameter that you provide and that are older than a second parameter you provide. For example, if you wanted a listing of all the files on your system that were larger than 50 blocks and that were more than 30 days old, you could use this shellscript.

The shellscript is set up to be relatively easy to use. It prompts you for the information it needs to run. It needs three parameters. First, it asks for the "lower cutoff size of the file." This is the smallest size file for which you want to look. If you want to look for files that are larger than 50 blocks, you would type in 100. Uh-oh, *why 100*? Well, the **find** command thinks of file sizes in 512K blocks, but the UNIX operating system on the Fortune thinks of file sizes in 1024K blocks. Thus, a UNIX block is twice the size of blocks that **find** uses. Therefore, you need to double the size of blocks when you talk to **find**, which is what we are doing in this shellscript.

The second parameter is the age of the files in days for which you want to look. This one is easy, if you want to list files that are older than 30 days, just type in 30. The final parameter is the directory from which you want to begin the search. If you want to check the entire system, then you should input just a slash (/). If you want to look at all the users, you would type /**u**.

The shellscript then asks if you want this list to go to a file. Generally, you would type in a **y** so that you could peruse this list at your convenience. If you type anything other than a y, the output will come to your screen.

The **find** command normally just lists out the names of the files that it finds that meet your parameters. We have built a slightly more complicated form of the **find** command so that it lists out all of the information that comes from the **ls -sail** command (e.g., size, modification date, owner, links, etc.) so that you can decide whether to delete the file or not. This is accomplished by using the **-exec** option on the **find** command. If you take a look at the **find** command as it is listed in figure 1 (see either line 16 or 18), you will notice the sequence **-exec ls -sail {} \;**. This is the part of the **find** command that does this little extra information gathering.

Installation

You must use an editor of some sort to type this shellscript into a file. We called the shellscript **bignold** because it is used to find big and old files that we might want to delete. You can call it anything you want.

After you type it into a file, you need to make it executable by using the **chmod** command as in:

```
$ chmod +x bignold
```

Once this is done, you can use the shellscript simply by typing its name to the UNIX prompt. We wish you success with this command that should not only help you free up disk space but should also save you time in doing that..□

Figure 1

```
:
echo "This command will list big and old files: "
echo
prompt "Enter lower cutoff size: "
read lc
prompt "Enter number of days old you want: "
read age
prompt "Enter directory to begin search: "
read dir
ans=n
prompt "Enter y if you want output to go to a file: "
read ans
case $ans in
  y)
    echo "Output is going to file called bignold.$lc.$age"
    find $dir -size +$lc -mtime +$age -type f -exec ls -sail {} \; > bignold.$lc.$age
    ;;
  *) find $dir -size +$lc -mtime +$age -type f -exec ls -sail {} \;
    ;;
esac
```

Administration

Continued from page 1

serial and parallel. Standard serial ports use an interface called RS-232 and usually have 25, 15 or 9 pin connectors. The cables attached to these ports can be up to 150 (or more) feet in length. Parallel ports use a "centronics" interface with a 36 pin connector. These cables are usually shorter (6-10 feet) but can transfer data faster than a serial cable. Since the Fortune System has only serial ports on it, one thing to check for when purchasing a printer for the Fortune System is that it uses a serial interface. Another alternative is to buy a "parallel to serial" or "serial to parallel" convertor which will allow you to use a parallel printer on a serial port.

The second column of information in the **devtype** file shows the **/etc/termcap** or **/etc/printcap** entry that is associated with the device. For example, the "console" has the "FT" entry which is the **termcap** entry for the Fortune terminal. (Note: See our previous articles for a discussion of the **printcap** and **termcap** files). The "tty01" entry has the "PLAIN2" entry from the **printcap** file.

The third column tells us the type of device:

P = printer
T = terminal
C = communication

In this case, the **console**, **tty02** and **tty04** are terminals; **tty01** is a printer; and **tty03** is a communication device. Note that even though **tty04** has a modem device, it is still defined as a terminal.

Column 4 tells us the number for the device in its class. For example, the **console** is terminal 0, **tty02** is terminal 1, and **tty04** is terminal 3. The same would apply to modem and printer entries.

Column 5 tells us the baud rate for the device. The baud rate represents the speed for the device. Four different speeds are shown here:

m = 19200 baud
n = 9600 baud
k = 2400 baud
i = 1200 baud

We will see these again when we look at the "ttys" file.

The next column gives us the description for the device, if it was entered. As you can see, we have a pretty good representation of devices here, including printers, terminals and modems. Notice that we have two separate entries for a modem - one is for an incoming modem and one for an outgoing modem. In For:Pro 1.8 and below, we had the ability to define one modem port that could be used for either incoming or outgoing modem calls. Many of us have not been able to do this easily with For:Pro 2.0 so we won't discuss that here. For now, assume that if you need the ability to set your system for both incoming and outgoing calls, you'll have to call your favorite "guru" or set up two separate ports and modems to provide the capability. From what we understand, Fortune will include this capability in the next For:Pro release.

The last column of information tells whether the device is enabled or not. A "1" means it is enabled and a "0" means it is disabled. A port must be enabled before it can be used.

This file can be edited directly using a For:Pro editor, but it is not recommended. We use it to see the status of our terminal ports without having to go through the Global Menu. By the way, if you want to define device connections directly from For:Pro without having to go through the menu system, execute the command "/m/sysman/devconn".

The two files in **/etc** that really control devices in For:Pro (and UNIX in general), are "ttys" and "ttytype". The net result of defining devices through the menu system is that these two files are updated. The "ttys" file tells us:

1. Whether the device is a login device or not.

2. The baud rate of the device.

3. The **/dev** name for the device.

Here is what the ttys file would like for the entries listed above:

```
1mconsole
0ntty01
1ntty02
0ktty03
1itty04
```

Notice that the **console**, **tty02**, and **tty04** are "login" devices. A modem port is a login device when someone can dial into it. If it is used for calling out,

Expand your Fortune® without spending a fortune.

New equipment costs big bucks. The Good Fortune Trader sells fully tested burned-in equipment, refurbished to Fortune® standards at prices that won't make waves with your cash flow. Call or write today for a listing of our specials.

THE GOOD FORTUNE TRADER

60 DAY WARRANTY

a division of Computer Services of Atlanta

(404) 447-4151

5825 Live Oak Parkway, Suite 2-C, Norcross, Georgia 30093

it is not a login device. Printers are also not login devices. The baud rate is indicated by the second character and is the same as what we showed in the **devtype** file.

The "ttytype" file shows the terminal type for login devices. It normally shows "unknown" for other devices such as printers. The associated **ttytype** file for the information shown above would look like this:

```
FT console
unknown tty01
FT tty02
unknown tty03
FT tty04
```

Again, "unknown" is used for non-login devices such as printers and dialout ports.

Both of these files can be edited in For:Pro, but again, we do not recommend it. First of all, whenever you shut the system down and bring it back up again, these two files are rebuilt automatically from the **devtype** file. Therefore, any changes you make would also have to be made by editing the **devtype** file or by using "define device connections". You can edit the **devtype** file, but you take the risk of messing up a critical system file.

Another interesting file in the **/etc** directory is the "disktab" (stands for disk

Figure 2.

```
# rdconf /dev/hd00
Configuration for: /dev/hd00
          Disk ID: G45 - 45 Mega byte          Volume ID: 546702235
      Format time: Wed 29 Apr 07:43 1987      Media type: Hard
      Modify time: Mon 14 Dec 05:35 1987       Disk type: G45
    Sectors/track: 16                   Drive attributes: 0
            Heads: 7                   Software interlace: 1
        Cylinders: 645                 Hardware interlace: 2
       Block size: 512                   System interlace: 2
     Landing zone: 645                      Write precomp: 320
Partition 0 start: 0                   Partition 0 length: 224
Partition 1 start: 224                 Partition 1 length: 5152
Partition 2 start: 5376                Partition 2 length: 66864
     Boot 0 start: 64                        Boot 0 length: 86
      Total reads: 916517             Total ECC/CRC errors: 0
     Total writes: 455682            Total soft ECC errors: 0
    Total retries: 0                Total sector ID errors: 0
     Total spares: 60                    Total other errors: 0
  Unusable spares: 3-59
      Used spares: 0(8036) 1(8824) 2(8825)
```


UNISOFT INC.

Experts in Computer Consulting - Sales - Service

Presents

FORTUNE SYSTEMS HELP

We Provide:

- Expert Trouble-shooting
- Years of programming and accounting experience
- *Free Initial Consultation*
- Training Available for:
  - * BASIC/IDOL/BAS and Multiplan
  - * UNIX and Ext. Fortune:Word
- Hardware Repairs & Upgrades
- FORTUNE trained experts at a cost that will surprise you!
- *Trade-Ins Welcome*
- Specializing in:
  - * Hard Disk Trade-In
  - * Hard Disk Upgrades

Authorized Fortune Systems Repair Depot
Fortune Systems National Value Added Reseller (VAR)
CALL: (303) 671-8600 EMERGENCY HELP: (303) 649-8505

UNISOFT INC. 14100 East Jewell Avenue Suite 27 Aurora, Colorado 80012

table) file. This file contains the descriptions for all of the hard disk drives that can be used on the Fortune System. If we cat the file out, an entry in it would look something like this:

```
Atasi:3046:G45:1010:0:16:2:16:7:6
45:645:320:0:645:512:1:2:2:
G45 - 45 Mega byte:
0:224:0:224:7840:0:8064:641
76:0:0:0:0:0:0:
0:0:0:0:0:0:0:0:0:
0:64:82:0:0:0:0:0:
0:0:0:0:0:0:0:0:0
```

What you are looking at are all of the hard disk parameters for an **Atasi** 45MB disk drive. If you execute the "rdconf" command, you should be able to spot the relationship between the data in this file and the "rdconf" information. (The output from the readconf command is shown in Figure 2).

First of all, notice that the drive type is "G45". The drive type for your drive is normally shown on the back of your machine. If you are wondering what type and size drive you have, you can issue the "rdconf" command to find out. The sequence "16:7" in the disktab file, for example, indicates the Sectors/track and number of heads as you can see from the "rdconf" output. The "conf block" (configuration block) for your disk is built when the disk is first formatted. Sometimes the conf block can be damaged or lost. When this happens, your system will not startup and it must be rebuilt. The conf block also indicates bad blocks that are on our hard disk (in the above example, blocks 8036, 8824 and 8825 are bad blocks and have been spared). Occasionally, we must spare out bad blocks so that the system does not use them. We will discuss all of these procedures in a later article.

The "motd" (stands for **m**essage **o**f the **d**ay) file in the **/etc** directory is used to enter any messages we want displayed whenever someone logs in. It can be used, for example, to advise users of backup times, system down time or even display a company name. You can either edit the file using a For:Pro editor or redirect output to the file. Since you should know how to use the editors by now, here are a couple of examples of redirecting output:

```
# echo "Welcome to Uni-Komp!" > /etc/motd
```

In this example, each user will see the welcome whenever they log in after this command has been executed. The "echo" command normally echoes back to our terminal screen. In this case, however, we have told it to "redirect" the output to the **/etc/motd** file by using the ">" (Note: see Volume 4 Number 4 for a discussion of redirection).

```
# echo "The system will be down for
backup from 11:00 to 12:00.
> Please plan your work accord-
ingly." > /etc/motd
```

In this example, the first ">" is the secondary For:Pro prompt - we did not type this in. Since we started the "echo" command with a ", For:Pro is smart enough to know that if there is a starting quote, there must be an ending one. Therefore, we get the secondary prompt after we hit the <RETURN> key following the first line, which is For:Pro's way of letting us know that it is expecting more input. The second greater than sign again redirects the output to the **/etc/motd** file. Here's another example where we "clear" the motd file so that nothing is displayed when a user logs in:

```
# echo > /etc/motd
```

The "fstab" file(stands for file system table) indicates each of the partitions or file systems on our hard disks (Note: refer to our previous articles for a discussion of partitions/file systems). A typical file looks like this:

```
# cat /etc/fstab
/dev/hd01::swap:1:1
/dev/hd02:/:rw:1:1
```

There are five fields of information here. Let's use the second line as an example and explain each:

/dev/hd02 - the /dev name for the partition

/ - the directory that it is "mounted" on

rw - indicates that the device is readable and writable

1 - indicates partition is backed up for "dump" command

1 - determines order for "fsck" command

If another hard disk is added to the system, this file must be updated to show the partitions for the disk in order for the system to automatically "mount" and "umount" the disk when the system starts up. We'll continue next time with a discussion of the concept of mounting and unmounting devices in For:Pro.□

F1000 Terminals with Keyboard in Stock/ Great Price!

256 K Memory Boards
Call Now!

Fortune 32:16 XP
with 512 K

We Buy Fortune!

Fortune 32:16 SX
with 512 K

COMM-A 4 Ports
COMM-A 6 Ports

We buy and
sell Fortune Systems
AT+T Systems

NEC 7710 Spinwriter
70 MB Disk Drive

We Buy Fortune!

National Remarketing Group

Buy, Sell or Trade

614-965-5868

Index to /u/fortune *news*

Once again it is time for our annual index to **/u/fortune** *news*. We hope that it will prove useful for our readers. The format for each entry shows the name of the indexed item, the name of the article and the issue that contains the article. The issue is specified by the volume number and the issue number, followed by the page number. The names of regular columns are abbreviated according to the chart at the bottom of the page.

#
UD/Intro to Shell Scripts Part 6 (4.1:20)
$
UD/Intro to Shell Scripts Part 6 (4.1:17)
$!
UD/Intro to Shell Scripts Part 7 (4.2:12)
$#
UD/Intro to Shell Scripts Part 7 (4.2:11)
$*
UD/Intro to Shell Scripts Part 7 (4.2:11)
$@
UD/Intro to Shell Scripts Part 7 (4.2:11)
& symbol
UNIX Basics/Part 6, Special Characters (4.5:9)
*** symbol**
Unix Basics/Part 2 (4.1:12)
UD/Intro to Shell Scripts Part 6 (4.1:17)
UNIX Basics/Part 6, Special Characters (4.5:11)
^H
UD/Intro to Shell Scripts Part 7 (4.2:10)
^J
UD/Intro to Shell Scripts Part 7 (4.2:10)
^K
UD/Intro to Shell Scripts Part 7 (4.2:10)
`
UD/Intro to Shell Scripts Part 6 (4.1:17)
| symbol (pipe)
UNIX Basics/Redirection and Pipes (4.4:15)
; symbol
UNIX Basics/Part 6, Special Characters (4.5:11)
> symbol
UNIX Basics/Redirection and Pipes (4.4:13)
>> symbols, appending to a file
UNIX Basics/Redirection and Pipes (4.4:15)
? symbol
UNIX Basics/Part 6, Special Characters (4.5:11)
.profile
SA/Part 2, For:Pro Directories (4.6:13)
/b directory
SA/Part 3, Directories/Files/Devices (4.7:9)
/bin directory
SA/Part 3, Directories/Files/Devices (4.7:9)
/dev directory
SA/Part 3, Directories/Files/Devices (4.7:9)
/dev/console
UD/Intro to Shell Scripts Part 6 (4.1:19)
/dev/null
UD/Intro to Shell Scripts Part 6 (4.1:19)
/u directory
Unix Basics/Part 3: Unix Filesystem (4.2:13)
1986 4th quarter financial results
News (4.2:1)
32:16 to Formula Upgrade
News/Fortune Announces Formula 4000 (4.6:2)
3780 RJE for Formula/32:16
News (4.7:2)
Accounting Systems, options
BAS (4.9:16)
addglobal, shellscript
News Concerning Our Software Disks (4.3:20)
adding new fields/IDOL
BAS (4.7:17)
BAS (4.11:21)
adding new users
SA/Part 7, Group and Password files (4.11:7)
American Medical International
Physicians Using the Fortune (4.2:4)
Apple/Pagemaker
Fortune-Mac Connection, The (4.1:9)
appt program problem
Help (4.7:19)
area code program
Of Time and Location (4.9:5)
arguments to UNIX commands
UNIX Basics, Part 4/Options and Arguments (4.3:13)
Arrow Office Automation announced
News (4.7:3)
autodialer
Creating an Autodialer/A Program to Dial Your Phone (4.8:1)
autodialer from Fortune:Word
FW/Indexes, Bullets, and more (4.8:20)
autosave feature/Fortune:Word
Review of Fortune:Word 3.0 (4.5:1)
Help (4.10:8)
awk
UD/Intro to Shell Scripts Part 6 (4.1:19)
background processing (&)/UNIX
UD/Intro to Shell Scripts Part 7 (4.2:12)
UNIX Basics/Part 6, Special Characters (4.5:9)
backspace control character
UD/Intro to Shell Scripts Part 7 (4.2:10)
bad blocks
UNIX File System Maintenance/Finding and Sparing Bad Blocks (4.11:5)
BAS Payroll, 2.0
BAS (4.2:22)
BAS support
BAS (4.7:14)
BAS/Partial Payment of Invoices
BAS (4.3:19)
BASIC 1.4, upgrade to 6.5.12
News/SCI Systems Makes Offer (4.4:3)
BASIC files
SA/Part 3, Directories/Files/Devices (4.7:9)
Basic4 to Fortune 32:16 conversion
Help (4.4:17)
baud rate
SA/Part 8, Device files (4.12:12)
bignold shell script
UD/Freeing Up Disk Space (4.12:11)
binary file transfer
SST Data Releases BLAZER/Performs High Speed Modem Transfer (4.6:4)
bit-mapped display
Fortune-Mac Connection, The (4.1:9)
Bold face a word, automatically
FW (4.7:18)
boot block
UNIX File System Maintenance/Part 1 (4.8:4)
booting from floppy
UNIX File System Maintenance/Part 2 (4.9:12)
Borelli, Ron
Interview with Ron Borelli (4.11:1)
bulletin board
Physicians Using the Fortune (4.2:3)
bullets
FW/Indexes, Bullets, and more (4.8:19)
Business BASIC
Physicians Using the Fortune (4.2:4)
buyout
News (4.2:2)
call display(false)/Glossary
Ten Fortune:Word Hints (4.2:19)
call display(true)/Glossary
Ten Fortune:Word Hints (4.2:19)

BAS-The BASIC Advisor | News-News from Fortune | MP-Multiplan | FW-The Fortune:Word Glossary
Help-/u/help | SA-System Administration | UD-The UNIX Directory | (4.5:11)-Vol. 4 Num. 5, page 11

FW/Indexes, Bullets, and more (4.8:19)
CANCEL/DEL, trapping
UD/The Finale, Part 9 (4.4:8)
cash registers, connecting to
Help (4.1:10)
cat
UD/Intro to Shell Scripts Part 6 (4.1:19)
CATLINE
Medlars:Database for Physicians (4.4:4)
CBIUA2 LIFO/FIFO problem
BAS (4.5:15)
CCNVZ
BAS (4.6:21)
cd
Unix Basics/Part 2 (4.1:12)
Unix Basics/Part 3: Unix Filesystem (4.2:14)
character map overload error
BAS (4.8:16)
chmod 777
Ten Fortune:Word Hints (4.2:19)
chmod command/UNIX
SA/Part 3, Directories/Files/Devices (4.7:11)
Choice Systems
BAS (4.2:22)
Choice Systems responds
Letters to /u/fortune news/Choice Systems and Payroll 2.0 (4.4:20)
chown command
SA/Part 7, Group and Password files (4.11:11)
CIS-EMC
Physicians Using the Fortune (4.2:4)
CIS-MED
Physicians Using the Fortune (4.2:4)
CIS-SKD
Physicians Using the Fortune (4.2:4)
Citicorp buys Fortune stock
News/Fortune Announces Formula 4000 (4.6:3)
clear screen
Help (4.1:10)
Clinic Management System
Physicians Using the Fortune (4.2:5)
cmp command/UNIX
Help (4.5:17)
UD/Verified Copy Command (4.6:17)
cold boot diskettes
SA/Part 4, /etc directory (4.8:11)
Comm-A boards, options discussed
SA/Part 4, /etc directory (4.8:13)
COMMAND !
Ten Fortune:Word Hints (4.2:19)
COMMAND S
Ten Fortune:Word Hints (4.2:19)
Concept Omega
BAS (4.1:2)
Concept Omega Conference
BAS (4.7:14)
Concept Omega Conference, report
BAS (4.10:14)
conf block
SA/Part 8, Device files (4.12:14)
conf block, hard disk
UNIX File System Maintenance/Part 1 (4.8:4)
connecting with cash registers
Help (4.1:10)
consoleless system
SA/Part 4, /etc directory (4.8:14)
Consoleless system, w/ For:Pro 2.0
News/SCI Systems Makes Offer (4.4:3)
Contractor, The
News (4.9:3)
control characters
UD/Intro to Shell Scripts Part 7 (4.2:9)
CONTROL-Y
Ten Fortune:Word Hints (4.2:19)
copy command with verification
Help (4.5:16)
copying files in UNIX
Unix Basics/Part 2 (4.1:12)
copying files, options described
UNIX Basics, Part 4/Options and Arguments (4.3:16)
copyright symbol, print
Ten Fortune:Word Hints (4.2:19)
Courier
Desk Top Publishing (4.1:6)
cp -Broust
Unix Basics/Part 2 (4.1:13)
cp -i
Unix Basics/Part 2 (4.1:13)
cp -i command, linking files/UNIX
UNIX Basics, Part 4/Options and Arguments (4.3:16)
cp -n
Unix Basics/Part 2 (4.1:13)
cp -t command/UNIX
UNIX Basics, Part 4/Options and Arguments (4.3:16)
cp -X option, verify while copying
Help (4.5:16)
cron, problems with
Help (4.1:10)
crontab
Help (4.1:10)
CTRL V
UD/Intro to Shell Scripts Part 7 (4.2:9)
Cucumber Bookshop, Inc.
Unix Book Stores (4.1:20)
cursor movement modes/Fortune:Word
Ten Fortune:Word Hints (4.2:1)
cylinders, hard disk
UNIX File System Maintenance/Part 1 (4.8:4)
date in Fortune:Word
Ten Fortune:Word Hints (4.2:19)
date setting, auto override/UNIX
Help (4.7:19)
date sorting using IDOL
BAS (4.9:16)
DDL
Desk Top Publishing (4.1:6)
default values
Ten Fortune:Word Hints (4.2:20)
define device connections
SA/Part 8, Device files (4.12:1)
delete paragraphs
Ten Fortune:Word Hints (4.2:1)
delete sentence
Ten Fortune:Word Hints (4.2:1)
delete words
Ten Fortune:Word Hints (4.2:1)
deleting files in UNIX (rm)
Unix Basics/Part 2 (4.1:14)
Desk Top Publishing
Desk Top Publishing (4.1:4)
devconn command/device connections
SA/Part 8, Device files (4.12:12)
Development Utilities, listing of
Fortune/SCI's Development Utilities/Tools for UNIX (4.10:16)
device connections
SA/Part 4, /etc directory (4.8:12)
devices
SA/Part 3, Directories/Files/Devices (4.7:9)
devtype
SA/Part 8, Device files (4.12:1)
devtype file
SA/Part 4, /etc directory (4.8:12)
Digipower air flow sensor
Monthly Bug Report (4.3:2)
directories
UNIX File System Maintenance/Part 1 (4.8:7)
directories, managing
Help (4.10:10)
directory information, ls -sail
UNIX Basics, Part 4/Options and Arguments (4.3:13)
disk drive problems
UNIX File System Maintenance/Finding and Sparing Bad Blocks (4.11:5)
disktab file
SA/Part 8, Device files (4.12:13)
Document Processing Tools
Fortune:Word Hints/Using the Index Generator (4.10:1)
double space
Ten Fortune:Word Hints (4.2:19)
dtree
Unix Basics/Part 3: Unix Filesystem (4.2:17)
du
Unix Basics/Part 3: Unix Filesystem (4.2:16)

echo
UD/Intro to Shell Scripts Part 6 (4.1:16)
ed editor
SA/Part 4, termcap, printcap and ed (4.9:20)
editors, vi
SA/Part 6, vi editor (4.10:3)
Electronic Databases
Medlars:Database for Physicians (4.4:4)
electronic mail
SST Data Releases BLAZER/Performs High Speed Modem Transfer (4.6:6)
UNIX Mail (4.7:4)
end of file unexpected error/UNIX
UD/Intro to Shell Scripts, Part 8 (4.3:9)
environment variables
SA/Part 2, For:Pro Directories (4.6:13)
error 102
BAS (4.8:18)
Error 103, possible cause
BAS (4.5:16)
error 103/BASIC/Fortune:Windows
BAS (4.3:19)
error 11
BAS (4.6:21)
error 12 after tape restore/BASIC
BAS (4.3:20)
error 19
BAS (4.12:5)
error correction/file transfer
SST Data Releases BLAZER/Performs High Speed Modem Transfer (4.6:4)
error messages, system
System Errors - What to do about (4.3:5)
Error, I/O Error 1/BASIC
BAS (4.11:21)
error: character map overload
BAS (4.8:16)
escaping special characters
UNIX Basics/Part 6, Special Characters (4.5:12)
exec
UD/Intro to Shell Scripts Part 6 (4.1:20)
exporting
UD/Intro to Shell Scripts Part 6 (4.1:17)
expr command/UNIX
UD/Intro to Shell Scripts, Part 8 (4.3:10)
feed/Glossary command
FW/Indexes, Bullets, and more (4.8:1)
FFCP
SST Data Releases BLAZER/Performs High Speed Modem Transfer (4.6:4)
field description limitation/IDOL
BAS (4.7:17)
field lengths/BASIC/IDOL redefining
BAS (4.7:14)
BAS (4.11:21)
field terminator/BASIC
BAS (4.11:21)
file name restrictions
Help (4.1:11)
file system check
UNIX File System Maintenance/Part 1 (4.8:3)
file system inconsistencies
UNIX File System Maintenance/Part 2 (4.9:1)
find command
UD/Freeing Up Disk Space (4.12:11)
find command/UNIX
Help (4.10:12)
finding large files
UD/Freeing Up Disk Space (4.12:11)
finding lost files
Help (4.10:10)
fold command/UNIX
Fortune/SCI's Development Utilities/Tools for UNIX (4.10:19)
footnotes/Fortune:Word
Help (4.5:17)
For:Pro 2.0
News (4.3:1)

BAS-The BASIC Advisor	News-News from Fortune	MP-Multiplan	FW-The Fortune:Word Glossary
Help-/u/help	SA-System Administration	UD-The UNIX Directory	(4.5:11)-Vol. 4 Num. 5, page 11

Our Memory Doesn't Cost You A Fortune

Would you like to:

Speed up your computer?
Add more users?
Use the latest software, like Fortune:Windows™?

Compatibility: The F1MB is absolutely 100% compatible with Fortune's own 1 Meg memory board.

Affordability: USA is proud to have equaled and exceeded the quality and performance of our competition, while offering substantial savings for both the dealer and end user.

Dependability: All boards are burned in for 72 hours and tested for 48 prior to shipment. Our One Year Repair or Replace Warranty underlines our confidency in the product.

Repairability: The F1MB facilitates repair in the field as it is 100% RAM socketed. Now you can repair a board by simply replacing a defective chip rather than face an expensive board exchange -- reducing maintenance costs and system down time.

If you have a Fortune SX, XP45 or recent model XP30, you can use our board by simply plugging it in. Older systems can be upgraded to use our board with a simple and economical modification.

For Details Call:

Upgrade Service Associates

850 Lexington Street, Suite 15, Waltham, MA 02154 Phone: (617) 239-8147

Our Guarantee: If you're not completely satisfied within 30 days, USA will gladly refund your money.

And now, from the company that made it easy to improve your memory. . .we've made it easier for you to sleep.



Tape backup doesn't cost a fortune either. Protect your priceless information today.

60Mb System DC600A DEALER $2395/incl PIO

For Details Call: **UpgradeServiceAssociates** RETAIL 2695/" "

850 Lexington Street, Suite 15, Waltham, MA 02154 Phone 893-4757

DEALER $1195/wo PIO

Our Guarantee: If you're not completely satisfied within 30 days, USA will gladly refund your money.

Laser/Type from InfoText
Desk Top Publishing (4.1:6)
Laserprinter Interface
Desk Top Publishing (4.1:6)
leaving your system on
We asked the Experts/Do You Leave Your System on Always (4.12:9)
legal citations/Fortune:Word
Help (4.5:17)
lf command/UNIX
UNIX Basics, Part 4/Options and Arguments (4.3:15)
Help (4.10:12)
LIFO/FIFO Bugs in Order Processing
BAS (4.5:14)
link command
Ten Fortune:Word Hints (4.2:19)
UNIX Basics, Part 4/Options and Arguments (4.3:15)
logging a user out
UD/Intro to Shell Scripts, Part 8 (4.3:10)
login devices
SA/Part 8, Device files (4.12:12)
login names
SA/Part 7, Group and Password files (4.11:9)
lost+found directory
UNIX File System Maintenance/Part 2 (4.9:10)
ls
Unix Basics/Part 2 (4.1:12)
Unix Basics/Part 3: Unix Filesystem (4.2:17)
ls -s
Unix Basics/Part 3: Unix Filesystem (4.2:17)
ls -sail command/UNIX
UNIX Basics, Part 4/Options and Arguments (4.3:13)
ls -tlr command, sort by date/UNIX
UNIX Basics, Part 4/Options and Arguments (4.3:15)
ls command/UNIX
UNIX Basics, Part 4/Options and Arguments (4.3:13)
Help (4.10:11)
Mac Plus
Fortune-Mac Connection, The (4.1:8)
MacDraw
Fortune-Mac Connection, The (4.1:9)
MacKermit
Fortune-Mac Connection, The (4.1:9)
MacPaint
Fortune-Mac Connection, The (4.1:9)
MAFSYS, manufacturing pgm announced
News (4.7:3)
mail shellscript for Global Menu
UNIX Mail (4.7:7)
Mail, Electronic
UNIX Mail (4.7:4)
mailing labels
FW/Printing Labels from Records Proc (4.3:1)
major/minor devices
SA/Part 3, Directories/Files/Devices (4.7:12)
manufacturing software
News (4.1:3)
Master Dealers, Listing of
News (4.1:3)
MEDI:FOR
Physicians Using the Fortune (4.2:3)
Medical Billing
Physicians Using the Fortune (4.2:5)
medical database
Medlars:Database for Physicians (4.4:4)
Medline
Physicians Using the Fortune (4.2:7)
Medlars:Database for Physicians (4.4:4)
memory caching
UNIX File System Maintenance/Part 2 (4.9:1)
Menu, for basic UNIX options
UD/The Finale, Part 9 (4.4:9)
message of the day
SA/Part 8, Device files (4.12:14)
meta characters
Unix Basics/Part 2 (4.1:12)
UNIX Basics/Part 6, Special Characters (4.5:9)
meta-character
UD/Intro to Shell Scripts Part 6 (4.1:17)
Microsoft Word
Fortune-Mac Connection, The (4.1:9)
mkconf command
UNIX File System Maintenance/Finding and Sparing Bad Blocks (4.11:7)
mkdevs command/UNIX
SA/Part 3, Directories/Files/Devices (4.7:12)
mkdir
UD/Intro to Shell Scripts Part 6 (4.1:20)
Unix Basics/Part 3: Unix Filesystem (4.2:15)
mklost+found
UNIX File System Maintenance/Part 2 (4.9:10)
mknod command/UNIX
SA/Part 3, Directories/Files/Devices (4.7:12)
mode, for files
UNIX Basics, Part 4/Options and Arguments (4.3:15)
modem
Creating an Autodialer/A Program to Dial Your Phone (4.8:1)
modem usage with For:Pro 2.0
SA/Part 8, Device files (4.12:12)
motd
SA/Part 8, Device files (4.12:14)
mp68 program
Help (4.4:17)
multi-volume backup
Monthly Bug Report (4.3:2)
Multiplan
Physicians Using the Fortune (4.2:7)
Multiplan 1.4
Help (4.1:11)
Multiplan printing problems w/HP
Help (4.4:17)
Multiplan symbolic format
Using Multiplan SYLK files (4.11:1)
Multiplan, blanking out zeros
Multiplan - How to Handle Zeros (4.3:5)
Multiplan/Converting to non-Fortune
Using Multiplan SYLK files (4.11:1)
Multiplier card, problems over 6mhz
News (4.3:2)
mv
Unix Basics/Part 2 (4.1:13)
National Library of Medicine
Physicians Using the Fortune (4.2:7)
ncheck
UNIX File System Maintenance/Finding and Sparing Bad Blocks (4.11:6)
New Dealers
News (4.1:3)
new master dealers
News (4.2:2)
newuser function
SA/Part 7, Group and Password files (4.11:8)
Note command/Fortune:Word
FW/Indexes, Bullets, and more (4.8:19)
options for UNIX commands
UNIX Basics, Part 4/Options and Arguments (4.3:13)
Order Processing, LIFO/FIFO bugs
BAS (4.5:14)
Pagemaker
Fortune-Mac Connection, The (4.1:9)
parallel port
SA/Part 8, Device files (4.12:12)
Partial Payments/BAS
BAS (4.3:19)
partitions, hard disk
UNIX File System Maintenance/Part 1 (4.8:5)
Pascal, new release
News (4.9:3)
passwd file
SA/Part 7, Group and Password files (4.11:9)
paste command
UD/Verified Copy Command (4.6:19)
PATH variable
SA/Part 2, For:Pro Directories (4.6:13)
pathname
Unix Basics/Part 3: Unix Filesystem (4.2:15)
Payroll, BAS 2.0
Letters to /u/fortune news/Choice Systems and Payroll 2.0 (4.4:20)
PCL
Desk Top Publishing (4.1:6)
permissions for file access
SA/Part 3, Directories/Files/Devices (4.7:11)

For:Pro 2.0 Review (4.7:21)
For:Pro 2.0 Installation Info
News/SCI Systems Makes Offer (4.4:3)
form feeds from IDOL, correction
BAS (4.1:2)
Formula
News (4.2:2)
Physicians Using the Fortune (4.2:4)
Formula 4000
News/Fortune Announces Formula 4000 (4.6:1)
Formula 4000 vs Formula 8000 chart
News/Fortune Announces Formula 4000 (4.6:2)
Formula hangs/problem configuration
Help (4.7:20)
Fortran 77, new release
News (4.9:3)
Fortune-Mac Connection
Fortune-Mac Connection, The (4.1:8)
Fortune:Link 1.0, 1.1
News/SCI Systems Makes Offer (4.4:3)
Fortune:Link PC announced
News (4.7:2)
Fortune:Windows/BASIC
BAS (4.3:19)
Fortune:Word
Physicians Using the Fortune (4.2:6)
Fortune:Word 3.0 installation
Help (4.4:17)
Fortune:Word 3.0 installation Info
News/SCI Systems Makes Offer (4.4:3)
Fortune:Word 3.0 new features
Review of Fortune:Word 3.0 (4.5:1)
Fortune:Word 3.0/glossary disk incl
FW (4.7:1)
Fortune:Word Adv Glossary Manual
FW/Dealing with Glossary Errors (4.1:22)
Fortune:Word bugs
Help (4.7:20)
Fortune:Word Glossary, attaching
FW (4.7:18)
Fortune:Word to Multiplan
Ten Fortune:Word Hints (4.2:19)
Fortune:Word, Formatting screenplay
What I Really Want to Do is Direct (4.4:18)
four megabyte board for Formula
News (4.2:2)
News (4.3:1)
Front Desk/Message announced
News (4.7:3)
fsck command
UNIX File System Maintenance/Part 1 (4.8:3)
UNIX File System Maintenance/Part 2 (4.9:10)
fsck messages described
UNIX File System Maintenance/Part 2 (4.9:14)
fstab
SA/Part 8, Device files (4.12:14)
Global Menu, adding programs to
News Concerning Our Software Disks (4.3:20)
Global Menu, options
Behind the Scenes @ the Global Menu/Little Known Options (4.3:17)
glossaries, global access/F:W
Ten Fortune:Word Hints (4.2:19)
glossary
Ten Fortune:Word Hints (4.2:19)
glossary errors
FW/Dealing with Glossary Errors (4.1:1)
Grafsman announced
News (4.7:3)
graphics program, Grafsman announced
News (4.7:3)
Grateful Med
Medlars:Database for Physicians (4.4:4)
greater than symbol/UNIX
UNIX Basics/Redirection and Pipes (4.4:13)
grep
UD/Intro to Shell Scripts Part 6 (4.1:19)
grep command, flags/UNIX
Help (4.7:19)
group file
SA/Part 7, Group and Password files (4.11:8)
HACK fix to termcap
News Concerning Our Software Disks (4.3:20)
Handshake Blazer
SST Data Releases BLAZER/Performs High Speed Modem Transfer (4.6:4)
hard disk errors, flaky data
UD/Verified Copy Command (4.6:16)
hard disk problems
UNIX File System Maintenance/Part 1 (4.8:3)
UNIX File System Maintenance/Part 2 (4.9:12)
UNIX File System Maintenance/Finding and Sparing Bad Blocks (4.11:5)
hard disk, 145 meg
News (4.3:1)
head command/UNIX
Fortune/SCI's Development Utilities/Tools for UNIX (4.10:19)
Helvetica
Desk Top Publishing (4.1:6)
HiLine, point of sale software
News (4.9:2)
home directory
Unix Basics/Part 3: Unix Filesystem (4.2:13)
Hospital Incident Reporting System
Physicians Using the Fortune (4.2:4)
HP laser Multiplan 16.6 pitch prob
Help (4.4:17)
I/O ERR 1/BASIC
BAS (4.11:21)
icheck
UNIX File System Maintenance/Finding and Sparing Bad Blocks (4.11:5)
IDOL
Physicians Using the Fortune (4.2:4)
IDOL 1.3 Update, description
BAS (4.11:22)
IDOL II vs. IDOL 1.2
BAS (4.7:17)
IDOL IV
BAS (4.10:15)
IDOL report features
BAS (4.9:16)
IDOL reports
BAS (4.6:21)
IDOL/BAS support comp. list
BAS (4.1:1)
if command/UNIX
UD/Intro to Shell Scripts, Part 8 (4.3:9)
if test -f
UD/Intro to Shell Scripts Part 6 (4.1:17)
Imagen
Fortune-Mac Connection, The (4.1:8)
Importing data to Multiplan
Using Multiplan SYLK files (4.11:1)
Index Output Document, reformatting
Fortune:Word Hints/Using the Index Generator (4.10:22)
Index, creating with Fortune:Word
Fortune:Word Hints/Using the Index Generator (4.10:1)
Informix
Product Releases from Fortune/SCI (4.7:22)
Informix printing help
Help (4.1:10)
inodes
UNIX File System Maintenance/Part 1 (4.8:6)
Inventory Monthly Totals
BAS (4.6:21)
ipl files
BAS (4.8:16)
ispell command
New Disk #10: ispell (4.11:15)
ITE
SST Data Releases BLAZER/Performs High Speed Modem Transfer (4.6:4)
kermit
SST Data Releases BLAZER/Performs High Speed Modem Transfer (4.6:4)
kill -9 0, logging a user out
UD/Intro to Shell Scripts, Part 8 (4.3:10)
L9 variable/IDOL, BASIC, redefining
BAS (4.7:14)
Laser Interface, Info re: For:Pro 2
News/SCI Systems Makes Offer (4.4:3)
Laser Printer
Desk Top Publishing (4.1:4)

SA/Part 7, Group and Password files (4.11:8)
PIO diagnostic
Monthly Bug Report (4.3:2)
pipes (|)
UNIX Basics/Redirection and Pipes (4.4:15)
Placement Power, personnel package
News (4.7:3)
Police Package announced
News (4.7:3)
ports, considerations when adding
SA/Part 4, /etc directory (4.8:13)
Postscript
Desk Top Publishing (4.1:6)
preventive maintenance
We asked the Experts/Do You Leave Your System on Always (4.12:9)
PRINT FID(0), Terminal Users/BASIC
BAS (4.3:19)
printcap file
SA/Part 4, termcap, printcap and ed (4.9:18)
printer problems, hung printer
Help (4.7:20)
printers, using unsupported
SA/Part 4, termcap, printcap and ed (4.9:20)
process id
UD/Intro to Shell Scripts Part 7 (4.2:12)
Progress
Product Releases from Fortune/SCI (4.7:22)
prototype document/Fortune:Word
Ten Fortune:Word Hints (4.2:20)
Help (4.10:8)
pwd
Unix Basics/Part 2 (4.1:12)
rdconf command
UNIX Basics/Redirection and Pipes (4.4:16)
SA/Part 8, Device files (4.12:14)
read
UD/Intro to Shell Scripts Part 6 (4.1:16)
Records Processing List Document
FW (4.7:1)
Records Processing/Mailing Labels
FW/Printing Labels from Records Proc (4.3:1)
redirection of standard output
UNIX Basics/Redirection and Pipes (4.4:13)
redirection of the standard input
UNIX Basics/Redirection and Pipes (4.4:15)
Registered symbol, print
Ten Fortune:Word Hints (4.2:19)
Release levels for all software
Release Levels for Fortune Products (4.3:3)
rem shellscript, reminders
Help (4.10:8)
renaming files in UNIX (mv)
Unix Basics/Part 2 (4.1:13)
Retail Data System announced
News (4.7:3)
reverse video
UD/Intro to Shell Scripts Part 7 (4.2:10)
rm
Unix Basics/Part 2 (4.1:14)
Unix Basics/Part 3: Unix Filesystem (4.2:13)
rm -i
Unix Basics/Part 2 (4.1:14)
rm -r
UD/Intro to Shell Scripts Part 6 (4.1:20)
rmdir
Unix Basics/Part 3: Unix Filesystem (4.2:16)
rolodex problems
Help (4.10:8)
root directory
Unix Basics/Part 3: Unix Filesystem (4.2:15)
SA/Part 2, For:Pro Directories (4.6:12)
RS-232
SA/Part 8, Device files (4.12:12)
SCI
Interview with Ron Borelli (4.11:1)
SCI Buyout approved
News/Fortune Announces Formula 4000 (4.6:2)
SCI Systems buyout offer
News/SCI Systems Makes Offer (4.4:1)
News/SCI Update (4.5:1)
Screenplays, formatting w/Fortune:W
What I Really Want to Do is Direct (4.4:18)
SCRIPT-IV
BAS (4.10:15)
sector, hard disk
UNIX File System Maintenance/Part 1 (4.8:4)
sed command/UNIX
UD/Verified Copy Command (4.6:19)
serial port
SA/Part 8, Device files (4.12:1)
set command/UNIX
SA/Part 2, For:Pro Directories (4.6:13)
setdt options/UNIX
Help (4.7:19)
shift
UD/Intro to Shell Scripts Part 6 (4.1:20)
Shifted keywords/Fortune:Word
Ten Fortune:Word Hints (4.2:1)
Shifted vs. Unshifted keywords/GLOS
Ten Fortune:Word Hints (4.2:1)
Shop Analyst announced
News (4.7:3)
shutdown system, change w/For:Pro 2
Help (4.4:17)
Sierra Dental Systems
Physicians Using the Fortune (4.2:5)
Sierra Medical Systems
Physicians Using the Fortune (4.2:5)
signal 2, trapping
UD/The Finale, Part 9 (4.4:8)
single space
Ten Fortune:Word Hints (4.2:19)
Software Diskettes
New Disk #10: ispell (4.11:15)
software release level, determining
Release Levels for Fortune Products (4.3:4)
Software release levels
Release Levels for Fortune Products (4.3:3)
sorting BASIC files
BAS (4.12:5)
sparing bad blocks on hard disk
UNIX File System Maintenance/Finding and Sparing Bad Blocks (4.11:5)
Special Characters
UNIX Basics/Part 6, Special Characters (4.5:9)
spelling checker for ASCII files
New Disk #10: ispell (4.11:15)

GO THE EXTRA FURLONG WITH
BASIC
JOB COSTING*
8. BASIC JOB COSTING
7. PURCHASE ORDERS
6. FIXED ASSETS
NOW... YOU'VE GOT A WINNER!
5. PAYROLL
1. GENERAL LEDGER
2. ACCOUNTS RECEIVABLE
3. ACCOUNTS PAYABLE
4. ORDER PROCESSING
*FROM $849.00 (Manufacturers Suggested Retail Price)
BASIC SOFTWARE CORPORATION
5201 POWHATAN STREET • BALTIMORE, MD 21207
301-448-9460

Spelling Checker vs. Editor
Ten Fortune:Word Hints (4.2:20)
standard input/output
UNIX Basics/Redirection and Pipes (4.4:12)
Stock Repurchase
News (4.1:3)
stty
UD/Intro to Shell Scripts Part 7 (4.2:9)
stty -echo
UD/Intro to Shell Scripts Part 7 (4.2:9)
stty echo
UD/Intro to Shell Scripts Part 7 (4.2:9)
stty intr
UD/Intro to Shell Scripts Part 7 (4.2:9)
sub-directories
Unix Basics/Part 3: Unix Filesystem (4.2:13)
super block, hard disk
UNIX File System Maintenance/Part 1 (4.8:6)
Super-Script Pharmacy Software announced
News (4.7:3)
support/BAS Software
BAS (4.7:14)
syntax errors, shell programming
UD/Intro to Shell Scripts, Part 8 (4.3:9)
syntax errors/Fortune:Word
FW/Dealing with Glossary Errors (4.1:22)
system administration, job overview
SA/Part 1 (4.5:4)
system error messages
System Errors - What to do about (4.3:5)
tail command/UNIX
Fortune/SCI's Development Utilities/Tools for UNIX (4.10:19)
Tape drive incompatabilities
Help (4.10:13)
Tape Restore, BASIC error 12
BAS (4.3:20)
Tape streamer prob w/SCSI Formula
News (4.7:3)
termcap file
SA/Part 4, termcap, printcap and ed (4.9:18)
Terminals, using non-Fortune
Behind the Scenes @ the Global Menu/Little Known Options (4.3:17)
SA/Part 4, termcap, printcap and ed (4.9:18)
test command, UNIX
UD/Intro to Shell Scripts, Part 8 (4.3:9)
The Resident
Physicians Using the Fortune (4.2:3)
Thoroughbred
BAS (4.1:2)
Thoroughbred 4GL
BAS (4.10:15)
Tigera Corporation
News (4.2:2)
News/SCI Systems Makes Offer (4.4:3)
Tigera Group (TYGR)
News/Fortune Announces Formula 4000 (4.6:2)
time zone information
Of Time and Location (4.9:8)
Times typeface
Desk Top Publishing (4.1:6)
TOM Software
Physicians Using the Fortune (4.2:3)
tracks, hard disk
UNIX File System Maintenance/Part 1 (4.8:3)
TrailBlazer modems
SST Data Releases BLAZER/Performs High Speed Modem Transfer (4.6:7)
trap command/UNIX
UD/Intro to Shell Scripts, Part 8 (4.3:10)
troff
Desk Top Publishing (4.1:6)
Fortune-Mac Connection, The (4.1:8)
ttys
SA/Part 8, Device files (4.12:12)
ttys file
SA/Part 4, /etc directory (4.8:14)
ttytype
SA/Part 8, Device files (4.12:13)
ttytype file
SA/Part 4, /etc directory (4.8:14)
underline sentence
Ten Fortune:Word Hints (4.2:19)
underline word
Ten Fortune:Word Hints (4.2:19)
UNIFY
Product Releases from Fortune/SCI (4.7:22)
Uninitialized program error
BAS (4.12:5)
UNIX file system
Unix Basics/Part 3: Unix Filesystem (4.2:13)
until loop
UD/Intro to Shell Scripts Part 6 (4.1:16)
Upgrading Software, Considerations
SA/Part 2, For:Pro Directories (4.6:10)
USER variable
SA/Part 2, For:Pro Directories (4.6:13)
uucp
SST Data Releases BLAZER/Performs High Speed Modem Transfer (4.6:4)
variable length, finding in UNIX
UD/Intro to Shell Scripts, Part 8 (4.3:10)
vcp command/UNIX (verified copy)
UD/Verified Copy Command (4.6:16)
verified copy command
UD/Verified Copy Command (4.6:16)
verified copy, cp -X
Help (4.5:16)
verifying
FW/Dealing with Glossary Errors (4.1:1)
version number, software
SA/Part 2, For:Pro Directories (4.6:10)
vi
SA/Part 6, vi editor (4.10:3)
video attributes
UD/Intro to Shell Scripts Part 7 (4.2:10)
VoiceComm announced
News (4.7:3)
vt100
Fortune-Mac Connection, The (4.1:9)
wait command/UNIX
UD/Intro to Shell Scripts Part 7 (4.2:12)
Watts, Frank — hired as VP Sales
News (4.7:1)
what command/UNIX
Release Levels for Fortune Products (4.3:4)
SA/Part 2, For:Pro Directories (4.6:10)
Wheels file
Desk Top Publishing (4.1:7)
wild card symbol *
UNIX Basics/Part 6, Special Characters (4.5:11)
windows
FW/Using Fortune:Word Windows (4.11:3)
WORD ERA
News (4.2:2)
News/Fortune Announces Formula 4000 (4.6:3)
work page, copying to it automatically
FW/Indexes, Bullets, and more (4.8:1)
work page/Fortune:Word
FW/Dealing with Glossary Errors (4.1:22)
WYSIWYG
Fortune-Mac Connection, The (4.1:9)
Younkins II, Daniel
News (4.12:3)
zeros, blanking from Multiplan
Multiplan - How to Handle Zeros (4.3:7)
zip code program
Of Time and Location (4.9:4)
Zip Tease Delight diskette
Of Time and Location (4.9:4)



TRADE - UPGRADE - EXPAND
Your FORTUNE SYSTEM

Processors:
- *Fortune 32:16 SX*
- *Fortune Formula 4000*
- *Fortune Formula 8000*
- *Fortune Multiplier (Run multi-user For:Pro & Fortune applications on a PC)*

Hard Disks:
- *30MB*
- *45MB*
- *70MB*
- *85MB*
- *145MB*

Tape Drives:
- *20MB External*
- *60MB External*
- *60MB Internal*

Memory:
- *256KB*
- *512KB*
- *1024KB*
- *4096KB*

Controllers:
- *Comm-A 4 Ports*
- *Comm-A 6 Ports*
- *Bit-Mapped Graphics Controller*
- *Fortune:Link Controller*

BONUS - BONUS - BONUS

All Fortune Licensed Software Upgrades and/or Replacements
At No Additional Charge with Processor Upgrade/Trade!

We can perform these trades and/or upgrades within 24 hours of receiving your system. In fact, if you ship your system overnight on a Friday for Saturday delivery, we will have it back to you on Monday morning, all ready to plug in and run; not losing the use of your system for a single work day. And, we will issue you a full 90-day warranty on the entire system.

JFM provides services to Fortune Systems owners nationwide either direct or through our authorized Affiliated Dealers. If you haven't used our services, I invite you to discover JFM; you will be glad you did.

Sincerely,

Morris Frydman
President

Talk to us, then talk to our clients

JFM
BUSINESS
SYSTEMS
The Merchandise Mart
Lobby Suite 144
Chicago, Illinois 60654
Call (312) 644-0870

Authorized Fortune/SCI Systems Master Dealer

Affordable laser printing for Fortune

Laser Printing! Kiss the expense good-bye.

QMS — KISS laser printer $1995.00

- cost per page 1.4¢.
- 9 resident fonts
- Print cap and print utility included.
- add-on sheet feeder provides total of 3 trays and envelope feeder.

THE GOOD FORTUNE TRADER

QMS - KISS from
The Good Fortune Trader
1 Year Warranty

a division of Computer Services of Atlanta
(404) 447-4151

5825 Live Oak Parkway, Suite 2-C, Norcross, Georgia 30093

FORTUNE SYSTEMS
an SCI Systems Company

No matter what your product or service, staying competitive requires control

Let QBC, your company's one source computer supplier, provide your business with solutions to:

Order Entry	Payroll
Accounts Payable	General Ledger
Accounts Receivable	Sales Analysis
Invoicing	Legal Time Billing
Inventory Control	Manufacturing Software
Office Automation	(MRP II Plus)

QBC is your one source for hardware, software, maintenance, customer training and data conversion.

MASTER DISTRIBUTOR FOR UPSTATE NEW YORK AND ERIE, PENNSYLVANIA

QBC

QBC Systems Inc.
375 NORTH FRENCH ROAD, AMHERST, NEW YORK 14120
(716) 691-5201 **1-800-443-5770**

We now have a new address. Please make the appropriate changes to your records. Any future correspondence should be sent to us at:

The Cambridge Consortium, Inc.
225 Crescent Street
Waltham, MA 02154
617 894-6900

/u/fortune news

225 Crescent Street
Waltham, MA 02154

FIRST CLASS

