/u/fortune* news

The Newsletter for Users of SCI/Fortune Computers January 1990 / Volume 7 Number 1

Fortune to DOS Connectivity Just Got Better:

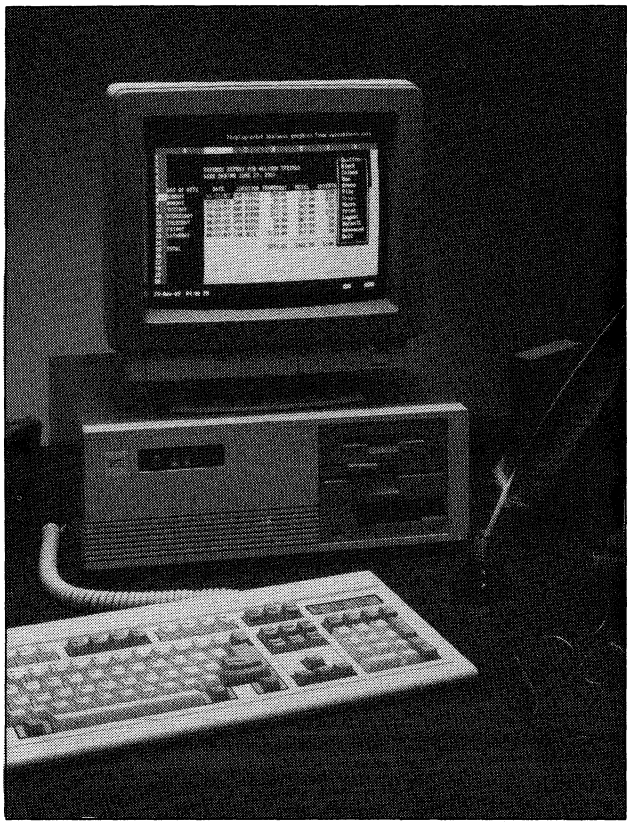
Fxfer Menu simplifies transfers and Fortune:Word to Word Perfect Conversion opens new vistas

See ***Page 12*** and ***Page 15*** *for more information.*

* * * * * * * * *

Also, Don't miss the retrospective of the 1980's given by Ray Wannall of the BASIC Advisor - Page 4.

Two New Extensions To Our PC-Architecture



5286

Smaller Cabinet

* Lower Cost
* More Desktop Space
* Improved Price/Perfomance

5386

25 MHz

* Additional power for more users
* Faster Response in existing applications
* Improved Price/Performance

Fortune 5000, a family of products ranging from single user workstations and standalone systems to low end and medium scale multi-user UNIX systems.

Call your dealer for pricing and more information.

SCI FORTUNE TECHNOLOGY

/u/fortune *news*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz

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, Inc. are affiliated with SCI/Fortune Systems.

Subscriptions cost $60 for one year within the USA. For foreign subscriptions, add $36 each year for added postage and handling costs.

Back issues are offered at:

Volume 1, Numbers 1 - 8	$2/each - $14/set
Volume 2, Numbers 1 - 7	$3/each - $18/set
Volume 3, Numbers 1 - 12	$4/each - $44/set
Volume 4, Numbers 1 - 12	$4/each - $44/set
Volume 5, Numbers 1 - 12	$5/each - $55/set
Volume 6, Numbers 1 -	$5/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $15 each, quantity discounts available. 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 the contents of **/u/fortune *news***.

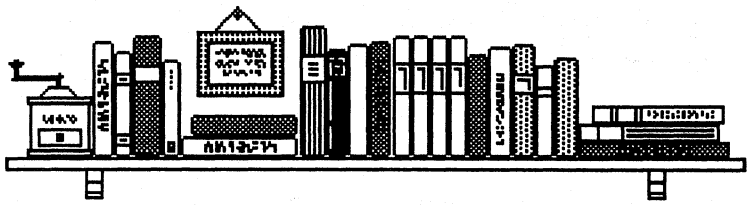
For more information, call or write:

The Cambridge Consortium, Inc.
225 Crescent Street
Waltham, MA 02154
(617) 894-6900

The text for this newsletter was prepared using an SCI/Fortune 5000 for both text input and typesetting. The typesetting was accomplished by using the Aldus Pagemaker program under VP/ix, which is a DOS emulation.

CONTENTS

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.

Yes, we can still get support for IDOL/BAS and BASIC, even on the Fortune 32:16. Some companies from last year no longer support Fortune, some never called us, and some, we are sure, will call late and ask to be listed next month. Regardless, here is our Fifth Annual IDOL/BAS Support Company Listing. A special thanks to all dealers who participated in this years listing.

SUPERIOR COMPUTER SYSTEMS, St. Louis, MO. Contact Steve Rosenfeld at (314) 731-3636. Support provided for end users and dealers on a subscription basis, but $80.00 per hour billing provided for first time callers.

INTERDISCIPLINE CONSULTANTS, INC., Wyckoff, NJ. Contact Joel Levine at (201) 848-8500. $75.00 to $95.00 per hour.

BEACON SYSTEMS, Palm Springs, CA. Contact Mike Eisen at (619) 323-4555. $65.00 per hour on-line service, $50.00 depot programmaing service.

UNI-KOMP, Houston, TX. Contact Dave Kloes at (713) 895-9900. $60.00 per hour.

UNICONCEPTS, Englewood, CO. Contact Tony Piacente at (800) 541-2082. $90.00 per hour.

DENICIN MANAGEMENT SERVICES, INC., Chelmsford, MA. Contact Dennis Sullivan at (508) 251-3063. $65.00 per hour.

ON ACCOUNT, INC. (Formerly Barry Dunn & Associates), Modesto, CA. Contact Tom Hansen or Dave Newaj at (800) 777-0463. Contract $50.00 per hour, non-contract $100.00 per hour.

BaSiC SOFTWARE CORPORATION, Baltimore, MD. Contact Lynne Crawford at (301) 448-9460. $65.00 per hour billed in fifteen minute blocks.

DESIGNING BUSINESS SYSTEMS, Tuscon, AZ. Contact Rick Garrett at (602) 299-7660. $40.00 per hour.

JOHN K. HARRIS & ASSOCIATES, Houston, TX. Contact John Harris or Kathy Bixley at (713) 667-1781. Contract billing and Time and Material billing at $65.00 per hour.

I.S.C. INTERNATIONAL, Milwaukee, WI. Contact Dave Meister at (414) 327-5809. $95.00 per hour plus expenses billed in fifteen minute blocks.

COMPUTER SERVICES OF ATLANTA, Norcross, GA. Contact Ken Palmer, Clark Hairston or Dr. Lee Allen at (404) 447-4151. $60.00 per hour billed in 6 minute blocks, 15 minute minimum. Quarterly contract rates available.

INNOVATIVE SOLUTIONS & TECHNOLOGY, INC., Foster City, CA. Contact AKI Eejima at (415) 578-0696. $60.00 to $70.00 per hour, contract billing.

GILL & PIETTE/PSG, Arlington, VA. Contact Richard Gill at (703) 761-1110. $65.00 to $140.00 per hour with flat fee arrangements under some circumstances.

BERKLEIGH COMPUTER SYSTEMS, Kutztown, PA. Contact Peter Keegan at (215) 683-3525. $65.00 per hour.

DAY PROM COMPUTER, INC., Dayton, OH. Contact Larry Corson at (513) 299-8555. $75.00 per hour, 7% discount if prepaid. Phone support billed in ten minute increments, on site billed hourly, add $1.00 per mile, one way, expense.

Next month we return to our standard format with LIFO/FIFO costing in BAS Inventory Processing, rumor squelching, the latest status on Solution-IV packages and more.

FOND FAREWELL TO THE 80'S.

We feel with the coming of the new decade it is fitting to look back on IDOL/BAS and Fortune computers in the 1980's. The 80's were probably the most exciting ten years in the small business computer industry. Here, for your amusement, is our unofficial history of the birth and growth of Fortune Systems, **/u/fortune** *news* and IDOL/BAS in the microcomputer environment.

1980: January: Science Management Corporation (SMC) of Bridgewater, NJ, began the decade as the new owner of the IDOL Data Base Management System, BAS Business Accounting Software and Business BASIC. The packages had been acquired by SMC when they bought a company called Data Technology Industries, Inc. of Riverdale, MD, in September of 1979. Data Technology had been owned by five ex-Navy men, one of whom, John-L Johnson, was the author of IDOL and the father of BAS. IDOL/BAS was available on MAI/Basic Four computers and, as of November, 1979, IBM

Series/1. Development of an independent dealer network was already in the works.

December: By year's end SMC had made IDOL/BAS available for REXON Business Computers, Pertec's PCC 2000 and Texas Instruments' 990 series.

1981: March: SMC announced the availability of IDOL/BAS and Business BASIC for the Unix operating system. The first ports were to the ONYX C8002, a Zilog 8000 based system, and Mercator's MSB 3000/4000, an INTEL 8086-based machine. By summer, plans were in the works to port BASIC to other 16 bit based computer systems, including machines with the Motorola M68000 processor.

May: Offices of SMC Proprietary Systems Division (PSD) were moved out of Riverdale and into Bridgewater, NJ. On the 20th the 200th vendor signed up with IDOL/BAS.

September - October: The Fourth Annual IDOL Dealer's Conference was held in Walt Disney World in Orlando, FL. It would be the last dealers' conference sponsored by SMC.

December: The year ended on a personal down note for yours truly, John-L Johnson and other SMC personnel when Richard Spencer, a coauthor of IDOL/BAS and top SMC PSD vendor, died on New Years Eve at the age of 36 while on vacation in California. It was Richard who patiently tolerated and answered my endless questions for hours at a time and taught me most of what I know today about IDOL and BAS. We still miss you, Richard.

1982: May: SMC software products were now available on Micro Five, Plexus, and Fortune computer systems. Timing test comparisons of Business BASIC (now called SMC BASIC) showed that Fortune computers were significantly slower than most or all other systems tested. This problem was eventually corrected.

September: Technicians from SMC had been working hand in hand with Fortune Systems technicians, who were attempting to make each of the BAS packages independent and user friendly. The documentation was being rewritten by Fortune Systems.

November: John-L Johnson incorporated Concept Omega Corporation. Shortly thereafter he left SMC PSD.

December: Fortune, who had purchased the exclusive rights to market BAS Wholesale Distribution, was signing up hundreds of retail dealers around the country. The 32:16 had been released without the final version of its Unix operating system called For:Pro, but BAS and IDOL were becoming, for the first time, widely available to small business owners.

1983: March: Fortune Systems Corporation went public. Shares went on sale initially at 22 and quickly dropped to 2.

April: Fortune Systems had started the new year with a bang. Thanks to For:Word (the word processor later called Fortune:Word), BAS and the revolutionary retail marketing approach to business computer sales, Fortune's first quarter earnings were $3.3 million on sales of $20.7 million.

October: For:Pro was released as a free upgrade for all Fortune Systems 32:16's. In order to install the new version of Unix, the hard disk had to be backed up, erased clean and reformatted. It has never been confirmed, but we believe the antacid business was booming.

December: By year's end the retail marketing plan was falling apart. Fortune "orphans", or 32:16 end users who could not find reliable support, were popping up all over the country. Dealers were dropping out and sales were falling, reaching a low of $9 million in the 3rd quarter. Fortune ended the year with a loss of $15.4 million on total sales of $54.4 million. The original president of Fortune Systems, Gary Friedman, was replaced by James S. Campbell who, with his new management team, promised renewed growth for the new year. Meanwhile SMC, reasoning that Fortune had significantly altered the BAS software, released its own version of the original BAS under the name "Thoroughbred Business Accounting". Rumors of bad feelings between the two companies abounded.

1984: February: The first meeting of the Fortune User's Group of Boston was held on Thursday the 8th. The meeting was attended by Nancy Olson, a regional representative of Fortune Systems who, along with other regional representatives, was attempting to solve the "orphans" problem.

March: On the 17th one of the other meeting attendees, Josh Lobel, reported the minutes of the meeting and the information provided by Nancy Olson in a five page letter he called "/u/fortune newsletter". It later became the Volume 1, Number 1 issue of "/u/fortune news". Fortune orphans around the world breathed a sigh of relief.

April: Fortune Systems, having abandoned its retail marketing strategy in favor of a VAR (Value Added Reseller) network, finally began an upswing in sales during the first quarter.

June: Volume 1, Number 3 of /u/fortune news announced the availability of the Harris print spooler from John K. Harris and Associates in Houston, TX, for IDOL/BAS on Fortune computers. One of the authors of this software package, Dave Kloes, would soon become well known to all readers of /u/fortune news as the "Unix Wizard".

August: Mark Palmerino, now coeditor with Josh, was first given credit as a "Contributor" to /u/fortune news in Volume 1, Number 5.

1985: March: Fortune Systems held its first and last "Worldwide Software Exposition" in Redwood City, CA. Their marketing strategy was to continue with the VAR program and to encourage participation by ISV's (Independent Software Vendors). Although other micro systems had been introduced into the market, Fortune, by virtue of its two years

in the business, was still considered the leader.

July: The New York Times reported that Fortune Systems Corporation had finally shown a profit for the 2nd quarter, its first in almost a year and a half. Fortune's release of 1.2.1 BAS Payroll took BAS another step further away from SMC's Thoroughbred version.

August: Fortune Systems Corporation's "Stir It Up" tour arrived in The Big Apple. Fortune was on a multi-city tour introducing 24 new products such as Fortune:Windows and Fortune:LAN. Attending were myself and my resident snoop, Lynne Crawford, who spent much of her time accidently overhearing conversations. One of those conversations involved two distinguished looking gentlemen who were drumming up subscriptions for their newsletter. They were, of course, Josh Lobel and Mark Palmerino. Thanks to Lynne's timely interruption the BASIC Advisor was born.

October: The first BASIC Advisor appeared in Volume 2, Number 5. It announced the sale of PSD and Thoroughbred Software by SMC to a company called Concept Omega Corporation. John-L Johnson was back at the helm. A network of dealers silently thanked the powers that be.

1986: April: Fortune Systems Corporation reported its first operating profit for a quarter since it went public in 1983. For the first quarter there was a net income of $345,000 on sales of $10.7 million compared with a loss of 3.8 million on sales of 9.8 million for the first quarter of 1985. It was also during this month Fortune introduced the 2.0 BAS Accounting packages. This move forever separated BAS and Thoroughbred.

June: Tigera Corporation was created as a wholly owned subsidiary of Fortune Systems. Its purpose was to market a line of Unix office automation software for non-Fortune hardware. Its first product was WORD ERA, a Fortune:Word look-alike.

July: Fortune Systems recorded its second consecutive profitable quarter. Mark Palmerino wrote an article entitled "The Unix Directory" for volume 3, number 7.

August: David E. Kloes became the second regular "outsider" columnist for /u/fortune news in volume 3, number 8, when he picked up "The Unix Directory" and made it a continuing series. Fortune announced the availability of Thoroughbred's BASIC, version 6.5.12, for the 32:16. Purchase and installation of this product was necessary in order to take advantage of future software upgrades for the Fortune 32:16. It was, however, the last product offered from Thoroughbred for the 32:16.

September: After just one year back in the Thoroughbred business Concept Omega Corporation sponsored a Thoroughbred Dealers' Conference in San Diego, CA. It was the first Thoroughbred Dealers' Conference in five years. Three top level managers from Fortune Systems Corporation showed up at the conference. We all started to believe the owners of Thoroughbred and BAS would finally be able to get along.

December: As the year ended the industry was buzzing with news and rumors about Fourth Generation Language which was on the way from many software manufacturers including Thoroughbred. For volume 3, number 12 /u/fortune news adopted the new look for its cover which is used to this day.

1987: January: Fortune stock showed a profit of $.03 per share for the previous year, due in part to the sale of Fortune's German subsidiary.

April: SCI Systems, Inc., of Huntsville, AL, jointly signed with Fortune Systems a letter of intent pursuant to which SCI would acquire Fortune Systems assets including R & D, manufacturing, sales, marketing and support operations. Hollywood screenwriter Jeffrey Fiskin submitted a Fortune:Word article for /u/fortune news volume 4, number 4.

May: A picture of a tired looking wizard on page 4 of volume 4, number 5, ushered in part 1 of System Administration by Dave Kloes. Fortune announced their new 32-bit Formula computers operating in a Motorola 68000 environment.

June: With the stockholders approval SCI Technology, Inc., bought Fortune Systems Corporation, leaving in its wake a company called Tigera Group, Inc. Tigera would later be bought out by Wang.

September: Concept Omega Corporation held its Second Annual Thoroughbred Leaders' Conference in Hilton Head, SC. New products introduced included Thoroughbred's TS-386 computer, a souped-up Wyse with a 32-bit Intel 80386 chip operating at 16 Mhz, and the Fourth Generation Language (4GL), including IDOL-IV and SCRIPT-IV programming languages.

December: /u/fortune news moved from Jason Street in Arlington, MA, to 225 Crescent Street in Waltham, MA.

1988: February: Due in part to the size and age of SCI, and due in part to the turnover of personnel at the Fortune division, Fortune drifted further and further away from its orphans as /u/fortune news contacts dissipated.

May: Thoroughbred BASIC 7.3.2 was released on the Fortune Formula products, making Thoroughbred IDOL-II and Thoroughbred Report available on those machines as well. The 4GL products, however, required 7.3.4 BASIC and so were not yet ported to any Fortune computer. It became obvious to all that the 32:16 products were locked forever in BASIC 6.5.12.

August: Fortune introduced a 300 Mb hard disk drive for the Formula 8000.

September: Concept Omega Corporation held its third Thoroughbred Leaders' Conference in Newport Beach, CA. It was the last conference of the 1980's for Concept Omega. New products introduced at this meeting included two new 20 MHz, 386-based "supermicrocomputers", a desk top model and a tower model, and a sneak preview of three accounting

packages generated in 4GL (Accounts Payable, Accounts Receivable and General Ledger).

November: On a road trip SCI/Fortune introduced the new SCI/Fortune 5000 computer based on Intel's 80386 processor. The new computer offered a choice of three operating systems: Interactive Unix, Xenix and MS-DOS. One Interactive Unix version (VP/ix) offered a DOS under Unix option, integrating the two operating systems for the first time.

December: Concept Omega Corporation bought ADD+ON SYSTEMS and offered it as an alternative solution for Third Generation Language business applications users. This move was the beginning of the end of Thoroughbred Business Accounting and BAS. Concept Omega Corporation was listed number 20 in INC. Magazine's list of the fastest growing independently owned companies in the United States.

1989: January: Thoroughbred Accounting Modules (formerly BAS), version 5.1, became available. It was the last upgrade for these products. /u/fortune news began the new year with newly established contacts at SCI/Fortune. Forbes Magazine ranked SCI, parent company of SCI/Fortune, number 10 in the Computer and Electronics industry for return on equity. IBM ranked ninth. In a surprising turn of events, Concept Omega replaced its Wyse-based TS-386 line with computers manufactured by SCI.

March: Thoroughbred BASIC 7.3.4 was released for the Fortune 5000, making 4GL products available to Fortune users for the first time. Thoroughbred IDOL-IV officially replaced IDOL-II, and IDOL-II was declared obsolete.

April: A 20 MHz version of the Fortune 5000 was made available. SCI Systems, Inc., was listed for the fifth consecutive year in Fortune Magazine's "Directory of the 500 Largest U.S. Industrial Corporations". It ranked number 376 for 1988 as compared to 456 for 1987.

August: SCI/Fortune offered MS-DOS version 4.01 and a new 780 Mb disk on its Fortune 5000 model 5386. Due to extreme lack of interest Concept Omega announced that its cruise conference was sunk. A series of road shows was to be substituted.

October: On its tour of major U.S cities, Concept Omega Corporation introduced Thoroughbred QUERY-IV, an SQL-compatible query language, BASIC 7.5.1, Series 8 BASIC and SOLUTION-IV Accounting Software written in 4GL. The three accounting packages available immediately were Accounts Receivable, Accounts Payable and General Ledger. Payroll, Inventory Control and Order Processing were in beta testing and due for release in early 1990.

December: Concept Omega Corporation sold its ADD+ON Accounting division to its largest vendor. Thoroughbred Accounting and BAS were declared obsolete and unsupported. New SOLUTION-IV modules were delayed in beta testing, but were to be available sometime in 1990.

And the beat goes on.❑

System Administration: Part 33

No Longer requires a Degree in UNIX Wizardry

This series is a must for all owners and users of the SCI/Fortune computer. Dave is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the SCI/Fortune computer, and is past president of the Houston UNIX User's Group. He contributes independently to **/u/fortune *news***.

by Dave Kloes

In the last issue we started to discuss a "checklist" that can be used as a guideline for installing SCO Xenix. In this issue, we will continue to develop this checklist.

10. In SCO Xenix, you have the ability to setup "multiscreens". While terminals other than the "console" can be used for multiscreens, the console is best suited for this task. Multiscreens give you the ability to have up to twelve logins simultaneously. When you initially log onto the console, you are using the first multiscreen. This login can be accessed anytime by depressing the "Alt" and "F1" keys. Let's say that we are logged into the first multiscreen and we are running our accounting software. The accounting process we are doing will take some time to run and we want to do something else. By depressing the "Alt" and "F2" keys, we will get another login prompt. We can now login and do word processing, for example, on this screen. We could also login to another screen by depressing the "Alt" and "F3" keys. There are twelve multiscreens available on the console (function keys F1-F12). Moving back and forth between each of these separate applications is as simple as depressing the "Alt" and appropriate function key for the screen we want to work in. This is a real life example of "multi-tasking" - the ability to perform multiple functions from one terminal. In our example above, we would depress "Alt" and "F1" to check the status of our accounting program. We might have word processing on "Alt" and "F2" and be logged in as "root" on "Alt" and "F3". If you are using SCO Xenix and are not using the multiscreen functions, you are not taking full advantage of the power of SCO Xenix.

For systems that we install, we edit the "/etc/profile" file to give different background colors for each multiscreen (assuming you have a color monitor). This way it is easier to distinguish which multiscreen you are using. Here is an example where we set the background color of the first three multiscreens:

```
if test "`tty`" = "/dev/tty01"
   then setcolor hi_white blue
elif test "`tty`" = "/dev/tty02"
   then setcolor hi_white red
elif test "`tty`" = "/dev/tty03"
   then setcolor hi_white green
fi
```

Notice that each of the twelve multiscreens have a unique "tty" device name using "tty01" thru "tty12". The "setcolor" command can be used to set your foreground and background colors. In this case, we set multiscreens 1 thru 3 to blue, red and green, respectively. In each case, the foreground color is a high intensity white color. To see the options that are available with the "setcolor" command, simply enter "setcolor" at the prompt.

11. To make life for the computer user a bit more interesting, we usually install the game "fortune" (not to be confused with the Fortune system). Many of you know about this "game" from the Fortune 32:16. Basically, it brings up a random and usually funny "fortune cookie" saying. Once the game has been installed, we modify the "/etc/profile" file to automatically bring up the saying and have the program wait for the user to depress <RETURN> to give them enough time to read it. Here is what we add to the "/etc/profile" file:

```
/usr/games/fortune
echo -n "
Depress <RETURN> to continue: "
read bunk
```

If you remember from our shell programming articles, the "read" statement waits for input from the user.

12. The normal SCO Xenix runtime also includes the "banner" command. This was also a utility that was available on the Fortune 32:16. Basically, the banner command produces a large banner on the terminal screen of whatever the user wants to display. In our installations, we usually banner the company name and redirect the output to the "/etc/motd" file. If you remember, whatever is in this file is displayed whenever a user logs in. The syntax for the command would be: `banner "Uni-Komp" > /etc/motd`. Now whenever a user logs in, "Uni-Komp" is displayed in large letters on the terminal screen.

13. The "/etc/systemid" file contains up to 8 characters that uniquely identify the system. We usually redirect the company name or a portion of it to this file. As it so happens, this id is also displayed as part of the login prompt. On our system, for example, we would enter:

echo "Uni-Komp" > /etc/systemid

Now the normal login prompt for each user on the system would be: `Uni-Komp login:` - There is an 8 character limit on the name you enter and this name would also be used when transmitting files using "uucp".

14. In SCO Xenix, there are a few files that just keep getting larger as time goes on. For example, the "/etc/wtmp" file will grow infinitely if you do not take care of it occasionally. For those that are interested in turning on login accounting, this file is used in that process. Most installations, however, could care less about login accounting. The easiest way to solve the problem is simply to remove the file.

Other files accumulate data as well and should be cleaned out periodically. For example, the "/usr/adm/messages" file contains system messages and can grow quite large over time. To clear out a file of this type, you can use this technique: `echo > /usr/adm/messages` - In this case, the file is cleared with the "echo" command. You could also use this technique with the /etc/wtmp file if you like.

Other files, such as the /usr/spool/uucp directory files should also be cleaned out periodically. You only need to worry about this if you are actually using the "uucp" utility. This can be done by the administrator or it can be added to the "rc" files to be done automatically whenever the system is restarted.

15. The SCO Xenix "mkdev" command can be used to make "root" and/or "boot" floppies for the system. This should be one of the first steps you take after your system has been configured and all of your devices have been installed. The "boot" floppy allows you to boot the system from a floppy diskette on occasions where you need to access and repair damage to the hard disk file system. It can also be used to access the hard disk when you do not know or have forgotten the "root" password. The concept is similar to using the "Cold Boot" volume 1 diskette to boot the Fortune system. Here is what a typical session would look like to make the "boot" floppy:

```
# mkdev fd

Choices for type of floppy filesystem.

1. 48tpi, double sided, 9 sectors per track
2. 96tpi, double sided, 15 sectors per track
3. 96tpi, double sided, 9 sectors per track

Enter an option or enter q to quit:
```

Normally, we choose option 2 which is a high density diskette. If you have a 3.5" floppy drive, you may want to select option 3 which enables you to fit more files on the diskette.

```
Insert a 96ds15 floppy into drive 0.
Press Return to continue or enter q to quit:

Choices for contents of floppy filesystem.

1.  Filesystem only.
2.  Bootable only.
3.  Root filesystem only.
4.  Root and Boot (only available for 96tpi floppy).

Enter an option or enter q to quit:
```

Since we want a boot diskette with a root filesystem on it, we select option 4 and then see: `Would you like to format the floppy first? (y/n)` - If your floppy has not already been formatted, enter "y" otherwise answer "n".

```
isize = 320
m/n = 1 15

Filesystem creation complete.
0+1 records in
1+0 records out

Bootable filesystem creation complete.
0 blocks

Root filesystem creation complete.
```

After the root filesystem has been created, a filesystem check is done (fsck) and you now have a bootable diskette. There is enough room on the floppy to add some commands that are not automatically put on the floppy that may be needed when you boot from the diskette. For example, we usually add the following commands: more, tape, cat, lc, ps, rm, rmdir, mkdir, df and umount

Since this is a mountable floppy, we use the following commands to copy these commands onto the diskette:

```
# mount /dev/fd096 /mnt
# cd /usr/bin
# cp more tape /mnt/bin
# cd /bin
# cp cat lc ps rm rmdir mkdir df /mnt/bin
# cd /etc
# cp umount /mnt/etc
# umount /dev/fd096
```

To boot from this floppy after the system has been halted, insert the diskette into the floppy drive and reboot the system. When the "boot:" prompt appears, depress the <RETURN> key to boot from the floppy disk. After the boot sequence has completed, you will get the "#" prompt and will have all of the commands that are available on the diskette to use. To mount the hard disk, we would enter:

```
# mount /dev/hd0root /mnt
```

Assuming that the hard disk is healthy, the /mnt directory accesses our hard disk. For example, if we enter "cd /mnt", we are actually in the root directory on the hard disk. After we have completed our actions and if all is well, we can unmount the hard disk; halt the system; and reboot from the hard disk.

```
# cd /
# umount /dev/hd0root
# /etc/haltsys
```

16. The last thing we do on any installation is to install two user-friendly menus (shell programs). One is for the every-day user and one is for the System Administrator. While SCO Xenix has a user-friendly "sysadmsh" utility for the System Administrator, our menu provides selections for the tasks that need to be done on a daily basis by the administrator.

We will include these shell programs in the next issue and will discuss how to install and use them.❑

News From SCI/Fortune

Quickening Pulse - A VAR Success

The following letter was addressed to Bob Bozeman of SCI/Fortune Systems and expresses the continuing success of SCI/Fortune's current VAR program.

We at Pulse Computer Systems, Inc., specializing in the medical field, are relatively new VARs for SCI Fortune and the Wallaby Resident Medical Management Software. We knew we had to rapidly establish a quality sales force, large enough to cover a broad market area, in order to meet our goals. In this vein, we were fortunate to have made contact with an Indiana company, Acordia Business Benefits of Evansville, a subsidiary of Blue Cross/Blue Shield of Indiana. Fortunately, in October of this year, we were able to finalize a sales agent agreement with Acordia, under which their eight sales representatives will market our SCI Fortune/Wallaby products. This will greatly enhance our marketing efforts in our tri-state area of Indiana, Kentucky, and Illinois.

Our immediate problem was providing the training necessary to enable these eight new sales agents to effectively market medical computer systems and software. We turned to SCI Fortune for help. With the efforts of SCI Fortune's Mary Ann Boesch and Jim Jones, we were able to put together a five-day training seminar which we believe will allow us to realize our marketing plans for 1990. As we expected, SCI Fortune came through for us with flying colors.

We here at Pulse Systems wish to say to SCI Fortune, Mary Ann, and Jim, a big "Thank You." Due to your joint efforts, Pulse Computer Systems, Inc. is poised for a great 1990.

We hope that SCI Fortune never changes it policy of helping their VARs attain their goals. Thank you for your help and your ongoing commitment.

Sincerely,

Jim Gager - Vice President

File-It! From Informix Is No Longer Available

Informix Software, Inc., has advised SCI Fortune that due to lack of demand they will no longer produce File-It! products. Consequently, the following products will no longer be available from SCI Fortune: File-It! (SQL Version) 3.0 for the Fortune Formula, as well as both File-It! 3.3 and File-It! (SQL Version) 3.0 for the 32:16. In addition, separately orderable File-It! documentation will no longer be available.

Meanwhile, negotiations are in the works to make Informix Software products available for the Fortune 5000. Expect to see Product Announcements in an upcoming issue of this magazine.

Technical Tips

Question: What is "Shadow BIOS," and why do Fortune 5000 Model 5386 systems have it?

Answer: All of the Fortune 5000 Model 5386 systems have shadowing of memory. Shadowing of memory through the use of Shadow BIOS is common in the high performance PC area. The term is used to describe the capability of a computer to copy its slow ROMs (BIOS extensions such as EGA and SCSI) to high speed local RAM, and thereby improve performance.

Shadow BIOS can provide a significant performance increase because the ROMs would otherwise operate at a much slower speed than the CPU. For example, a system without Shadow BIOS might have a CPU operating at 20 MHz and I/O channel operating at 8 MHz. In this case, whenever the CPU accesses ROM (BIOS or EGA, etc.), it must switch to 8 MHz and then back to 20 MHz when finished. The CPU must access ROM whenever it wants to communicate with any peripheral, such as video, floppy, or hard disk. Therefore, without Shadow BIOS the CPU spends a large amount of time operating at a much slower speed than its 20 MHz capacity would indicate.

Question: Why does the memory between 640K and 1024K appear to be missing on my Fortune 5000 Model 5386 system?

Answer: The Shadow BIOS is mapped into this 384K section of memory between 640K and 1024K. This is why a 5386 system configured with 1024K memory gives a memory check on the screen of:

640K Base 0K Extended

Since the amount of Shadow BIOS is not indicated, it might seem that 384K is missing but it is not.

Question: Can we change the memory between 640K and 1024K to extended memory?

Answer: Extended memory CANNOT be changed on the Fortune 5000 Model 5386 systems. For example, a system with 1024K memory will only come up to "640K OK Extended." Memory cannot be changed to 512K Base and 512K Extended, or changed to 684K Base and 384K Extended.❑

Fxfer Menu

The Easy way to transfer files between Your Fortune and IBM computers. And Fxfer Menu can operate as a Friendly Front-End to Tar.

The **Fxfer Menu** is now available as a friendly front-end to the popular Fxfer program. As most of our readers know, we introduced **Fxfer** (pronounced *eff-x-fer*) several months ago. Fxfer is a software product that allows Fortune owners to read Fortune diskettes in an IBM computer with a 1.2 MB floppy drive. Thus, Fxfer allows easy, trouble-free transfer of information from the Fortune computer to the IBM and in the other direction, as well. That is, you can transfer information from your IBM to your Fortune computer - and all of this happens on the standard Fortune floppy.

The standard Fxfer program is modeled on the familiar and widely used Unix *tar* program. As such, it is a Unix command line driven program. For example, to copy some files from your Fortune computer onto a Fortune floppy, you would need to put a Fortune formatted floppy into the Fortune floppy drive and type at a Unix prompt the following:

```
fxfer cv letter letter.dc letter.fr
```

This command would copy the Fortune:Word document entitled letter and its accompanying files (letter.dc and letter.fr) onto the floppy. You would then take the floppy out of the drive, and put it into an IBM floppy drive and then on the IBM machine you would type:

```
fxfer xvF 0
```

Fxfer would then copy the files from the floppy onto the IBM's hard disk. We think that the capability of transferring files from Fortunes to IBMs and back again is incredible and we are still amazed at how easy it is - even with the standard fxfer command.

But now we've made it even easier with the Fxfer Menu program. The Fxfer Menu program is basically an easy-to-use front-end to the standard fxfer program and utilizes intuitively easy-to-understand pop-up menus to guide the user through the use of the product. In addition, there is context sensitive help available at all points in the program so that you can sit right down at your computer and use Fxfer Menu without reading the documentation. Of course, you should read the documentation, but the point is that we have designed Fxfer Menu to be really easy to use.

Please note that the Fxfer Menu works exactly the same on your Fortune and your IBM. You don't have to know Unix or DOS and you don't have to learn two different products - one for the Fortune and one for the IBM.

The Fxfer Main Menu

When you start Fxfer Menu you are presented with a Main Menu which is portrayed in Figure 1. The Main Menu allows you to select the three main processes that Fxfer can perform. First, you can read an Fxfer archive. Second, you can write to an Fxfer archive. Finally, you can list the contents of an Fxfer archive.

The user may select one of these options in a number of ways. First, the user may simply type 1, 2 or 3 to select Read, Write or List an archive respectively. The other method is to use the arrow keys to move the cursor among the three selections. The RIGHT ARROW moves the cursor to the right in a circular fashion. For example, if the cursor was on the Read option and the user hits the RIGHT ARROW, then the cursor will move to the Write option. If the RIGHT ARROW is depressed again, the cursor will move to the List option. If the RIGHT ARROW is hit once again, the cursor will circle around and end up on the Read option. You may select any option that the cursor is on by hitting the RETURN key or the EXECUTE key on a Fortune terminal. Thus, this method of selecting options should be very familiar to any Fortune user since it is the method employed by the Global Menu, Fortune:Word, etc.

On the Fxfer Main Menu, you may also hit the HELP key on the Fortune or the F1 key on the IBM to get help. And finally, to quit the program, you simply type the ESC key on the Main Menu.

The Extract Menu

Let's say you want to read some files from an Fxfer archive. From the Main Menu you simply type 1 and you will be presented with the Extract Menu which is also portrayed in

Figure 1. Option 1 on this menu will extract all the files that are in the Fxfer archive and copy them onto your hard disk. Option 2 allows the user a little more control over what is extracted by allowing the user to type in the names of the files that he or she wishes to extract. Options 3 and 4 are "toggle" switches and indicated as such by being between angle brackets (**<>**). Option 3 toggles between three options: *Overwrite*, *Query*, and *No Overwrite*. Overwrite means that Fxfer will overwrite any files on the hard disk that have the same name as the one being extracted. Query means that Fxfer will ask you on a file-by-file basis if you want to overwrite when there are filename collisions. Finally, No Overwrite means that if there is a file name collision, Fxfer simply will not copy the file from the archive to the hard disk.

Option 4 is also a toggle switch which toggles between *Single Volume* and *Multiple Volume*. This controls whether Fxfer will read past the end of the floppy for more files. Generally, the Multiple Volume toggle should be selected if the floppy was part of a multiple volume write.

On the IBM version of Fxfer Menu there is one more toggle switch which appears as *DOS conversion* and can toggle to *No Conversion*. When in the DOS Conversion setting, Fxfer will append the proper end-of-line sequence on the file as it is being extracted. Generally, when you are transferring text files, you should do DOS conversion. If you are transferring binary files, you should select No Conversion.

Of course, if you type the HELP key on the Fortune or the F1 key on an IBM, you will receive a helpful description of all the options on the Extract Menu and what they mean. Finally, to go back to the Main Menu, just type the ESC key.

The Write Menu

From the Main Menu just strike 2 in order to see the Write Menu. Your screen will look like Figure 2. The first option, *Write All Files*, will make a copy of all the files that are in the current directory and put them on the Fortune floppy. If you are on an IBM computer, the files will be written to the A: drive.

The second option, *User-defined List*, allows the user to selectively input the names of files to write to the archive. At this point, standard Unix wildcards are accepted. For example, if you wanted to write all the files in the directory that ended with ".let" then you could enter "`*.let`" because the "*" is a wildcard which will match anything.

```
        Write Menu

  1. Write All Files
  2. User-defined List
  3. Point and Shoot
  4. <Single Volume>

  Use UP or DOWN Arrow
    EXECUTE to select
```

Figure 2. The Fxfer Write Menu

The third option is a truly friendly way of specifying files to write to the floppy drive. When you choose the **Point and Shoot** option, all the files in the current directory are displayed in a list on your terminal. You can move up and down the list with the UP and DOWN ARROW keys. When you find a file that you wish to write to the floppy, simply hit the INSERT key (which will highlight that filename). This "flags" the file and adds it to a list of files to write. If you flag a file but later change your mind, you can use the DELETE key to delete it from the list. If the number of files in your directory is greater than will fit on one screen, you can use the NEXT SCRN and PREV SCRN keys to move back and forth among

```
 Fxfer Main Menu          Directory: /u/net/fm2

  1. Read Fxfer Floppy    2. Write Fxfer Floppy    3. List Fxfer Floppy

Use RIGHT and LEFT Arrow keys to move. EXECUTE Key to select. ESC to quit.

      Extract Menu

  1. Extract All Files
  2. User-defined List
  3. <Overwrite>
  4. <Single Volume>

  Use UP or DOWN Arrow
    EXECUTE to select
```

the screens.

When you have chosen all the files that you want, simply type the EXECUTE key on the Fortune or the RETURN key on the IBM.

The final option on the Write Menu on the Fortune is the *Single Volume* toggle switch. Like the toggle switches on the Extract Menu, this one will toggle between Single Volume and Multiple Volume. If you expect to write out more than one floppy full of information (about 780 KB), then you should use the Multiple Volume selection.

On the IBM version, there is one other toggle switch - the DOS Conversion switch which is like the switch on the Read Menu. In this case, however, if DOS Conversion is selected when you write files to the floppy, then the DOS end-of-line sequence will be replaced by the appropriate Unix compatible end-of-line sequence.

As in the other descriptions above, the HELP key (F1 on DOS) will give you help and the ESC key will take you back to the Main Menu.

The List Menu

The last option on the Main Menu is the List option. When you choose it, you will see the menu as portrayed on Figure 3. The List Menu is slightly simpler than either the Read or Write Menus and is used to get a list of the files that are in the Fxfer archive. To get a listing, just choose the List Archive option. If you want a listing from a Multiple Volume archive, then you will need to toggle the Single Volume option to Multiple Volume. Like the other menus, the HELP key (F1 on DOS) will get you some help and the ESC key takes you back to the Main Menu.

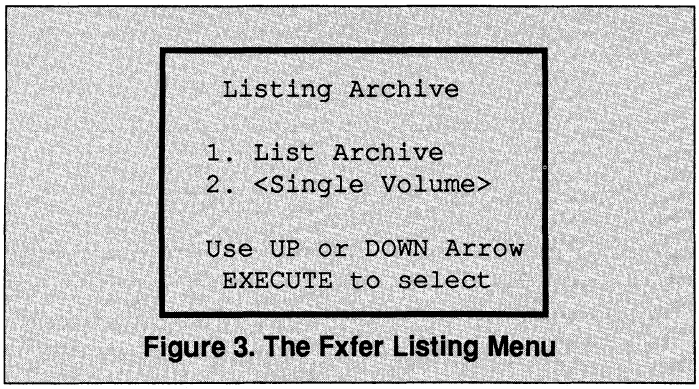


Figure 3. The Fxfer Listing Menu

A Friendly Front-End For TAR

Although the Fxfer Menu was written to simplify the use of Fxfer, we have built it so that it will function as a friendly front-end for the standard Unix *tar* command. This added benefit will help users and system administrators use tar more easily.

Fxfer Menu assumes two things when it starts up. First, it assumes that you will be writing to and/or reading from the floppy drive. Second, it assumes that you will be using Fxfer (as opposed to tar). If you want to modify either of these two assumptions, you would use command-line flags. The Unix command to start Fxfer Menu is simply *fm*. If you would like to start Fxfer Menu but have it use the *tar* command instead of the fxfer command, then you would type the following:

```
fm -t
```

If you want to read from some other device or file than the floppy drive, you can do so by supplying this information on the command-line. Let's say that you have created an Fxfer archive and it is in a file on your hard disk called "listing." To access this file instead of the floppy drive you would type:

```
fm -f listing
```

Finally, you can combine these two flags so that you are actually using the tar command instead of the fxfer command and so that you are reading the file called "listing" instead of the floppy. To do this, type:

```
fm -tf listing
```

or, alternately,

```
fm -t -f listing
```

Conclusion

Fxfer Menu is a friendly front-end for our Fxfer command and can be used as a friendly way of dealing with the standard Unix *tar* command. Its intuitive pop-up menus make transferring files between computer systems very easy - easy enough so that anyone can do it.

The Fxfer Menu system is available from The Cambridge Consortium, Inc. and is priced at $200. Remember, with Fxfer Menu you will have the ability to transfer files between your Fortune and IBM computers on Fortune formatted floppies. No longer do you have to use modems or wires or other slow and error prone ways of transferring information. In addition, with the Menu version, we have made the process as easy as it can be. And, of course, the Fxfer Menu can be used with *tar* to make the system administrator's life more hassle free.

We are now accepting orders for Fxfer Menu. If you are interested in placing an order, send a check for $200 to: **The Cambridge Consortium, Inc.**, 225 Crescent Street, Waltham, MA 02154. If you have any questions, please feel free to call us at (617) 894-6900.❑

Fortune:Word to Word Perfect Conversion Program Announced

We are pleased to announce that we have developed a companion product to our **fxfer** floppy transfer program. Our program converts **Fortune:Word** files to **Word Perfect** files and back again. **Word Perfect** is a well-known and widely used word processor used primarily in the DOS world. However, **Word Perfect** is not limited to DOS as it is now available under Unix.

Many of our readers are dedicated **Fortune:Word** users who have occasion to send files to PC users using the **Word Perfect** program. In the past, this has required the user to save the file in plain ASCII form, send the file to a PC via modem or direct connection, and then import it into **Word Perfect**. Besides being cumbersome, *all* of the formatting features of the document were lost -- underline, bold, margins, tab settings, etc. We felt that we could improve upon this state of affairs by creating a conversion program that would allow the **Fortune:Word** user to preserve almost all of the formatting that had been painstakingly put into a word processing document. So we created our conversion program.

With **Fortune:Word <-> Word Perfect**, almost all attributes such as bold and underline are maintained. So are margins, tabs, indents, centered lines, headers, footers, double underlines, overstrike -- pretty much all formatting functions except for footnotes (which will be added if there is sufficient interest). In fact, we think our conversion process makes it simple and practical to share **Fortune:Word** documents with PC users using **Word Perfect**. The conversion process works both ways, so you can procure files from people using **Word Perfect**, then translate them to Fortune:Word, edit them to your heart's content and, if need be, convert them back to **Word Perfect** and send them back to the original owner!

How The Conversion Works

Each **Fortune:Word** file consists of three files - the main file with the text and two associated files that contain formatting information - these files have a **.dc** and **.fr** extension. On the Fortune end, you run a program that converts the three files into a single file. Using a menu driven program that we provide, entire directories of **Fortune:Word** files can be converted, leaving non-**Fortune:Word** files undisturbed. The fact that our menu program can selectively find and convert all the **Fortune:Word** files in a particular directory makes the conversion process a snap. That is, it saves the user the trouble of looking through the directory and making a list of all the **Fortune:Word** files and then typing that list into the computer! Of course, the program can be used to convert just one **Fortune:Word** document just as easily as it can convert an entire directory of **Fortune:Word** files. The point is that we have attempted to make the conversion process as trouble-free as possible!

This file (or files) is then transferred to a PC using a modem or the **fxfer** program (See the article on page 12 which explains how easy it is to use **fxfer** now that we have **Fxfer Menu**).

On the PC, there is a second program that completes the conversion process - it is an easy-to-use program that can be learned in just a few moments. The entire process is very quick, and you're left with files for **Word Perfect** version 5.0 or 4.2. The process to go back to **Fortune:Word** is the same thing in reverse. We have had this product in Beta test for a couple of months now and we have already heard from some Fortune sites who are planning to use this conversion capability to improve productivity between their Fortune users and PC users.

If you want to go over to a 386 system running Interactive 386/ix Unix or SCO Xenix such as the Fortune 5000, you can do the PC end of the file conversion on the 5000 as long as you have the VP/ix product. The **Word Perfect** files will be compatible with **Word Perfect** 4.2.

Conversion Service

We know that this product is not for everyone. In some cases, there will be the need to do a one-time conversion from **Fortune:Word** to **Word Perfect**. Although this product is very economical and would be worth the price for even a one-time conversion, we are making available a conversion service. That is, should any of our readers desire to have their **Fortune:Word** files converted to **Word Perfect**, we can do that. If you are interested in more details about this service, please call at the number given at the end of this article.

Conclusion

With the introduction of this product, we are stretching the ability of a Fortune computer to communicate with non-Fortune users. We think this is an exciting program, and we look forward to hearing from our readers. The cost of the **Fortune:Word <-> Word Perfect** is $150. It comes complete with all of the programs you'll need for both sides, along with a complete set of instructions. For more information, call Josh Lobel at 617 894-6900.❑

Preliminary Survey Results

We would like to do several things in this article. First, we want to give our readers some preliminary results of the survey we included in last month's **/u/fortune** *news*. Second, we would like to thank those who have returned the survey in a very timely fashion. Third, we want to encourage all those who haven't sent in their survey to do so as soon as possible.

Let's begin with a thanks. As of the writing of this article, we have received only 13 returned surveys. To those 13 individuals, thank you very much for your help and your input!

To the rest of our subscribers we would like to encourage you to take the time and fill out the survey. Why? Well, we are counting on you. The simple fact of the matter is that each month we try to craft an issue of **/u/fortune** *news* that will help **YOU** get the most from your Fortune computer. But it is clear that there are individual needs and wants with respect to computer usage - so you must talk to us.

There is a further point that we would like to make. We intend to use the information from the **returned** surveys to plan articles. That means that, at least at this point, 13 people will be shaping **/u/fortune** *news* for the months to come. As you will see in the preliminary results that we present below, this might produce a magazine that does not meet **YOUR** needs. So, what can you do? Well, you should fill out and send in your survey right away so that your opinions can be counted.

Preliminary Results

Types of machines: All of those who responded own at least one Fortune 32:16. About 15% own a Formula (4000 or 8000). One person owns a 5386 (Fortune 5000). About 15% mentioned that they use IBM PC's or compatibles.

Software used: Fifty-three percent of the respondents said that Fortune:Word was the most important software product that they used. An additional 38% said that it was the second most important program. Thus, over 90% of the respondents felt that Fortune:Word was the first or second most important software product they owned.

The next most used program was Multiplan or a spreadsheet. Fifty-three percent of the respondents felt this way. The following is a listing of other programs and their perceived importance: Databases - 38%, Communications - 38%, BAS - 31%, Accounting (besides BAS) - 23%, and Computer Languages (e.g., C, Fortran) - 23%.

Unix: Almost 62% of the respondents felt that they understood Unix "Not well at all" or "Moderately well" while 31% felt they understood Unix "Well enough" or "Very Well." However, 12 out of the 13 respondents, or 92%, wanted to know more about Unix.

Fortune:Word: Most respondents felt they knew Fortune:Word pretty well. Nevertheless, 9 out of 13 respondents (69%) want to know more about Fortune:Word.

Spreadsheet: Seven out of 13 respondents (54%) want to know more about spreadsheets.

System Administration: There was a 46 to 54% split between those who understand system administration. Again, 92% desired to know more about system administration.

Other topics to discuss: Shell scripts - 70%, Communications 62%, Product Reviews - 38%, C programming - 38%, Accounting - 23%.

Rating of articles: Unix Hints - 85%, System Administration - 70%, /u/help - 70%, Fortune:Word - 62%, Product Reviews - 54%, Basic Advisor - 46%, Product Announcements - 38%, Editorials - 38%, News From Fortune - 38%, Book reviews - 23%.

Preliminary Overall Conclusions

It seems that the major desire being presented by those who have returned their surveys is for *more* system administration related issues. Over 90% of the respondents want to know more about Unix and over 90% want to know more about system administration. In addition, the number one topic picked was shell scripts, followed by communications. As for the ratings of the "useful" articles, Unix Hints topped the list with System Administration and /u/help tied for second.

There was a secondary, albeit weaker, voice that wants to see more on Fortune:Word and other office management issues (accounting, BAS, product reviews, etc.).

Another Call

We hope that this article has been helpful for our readership. However, we suspect that the results that we have received are weighted a little heavy on the System Administration side. There may be some good reasons for this, but we would, once again, encourage every subscriber to take the time and fill out a survey and send it to us. That way, your voice will be heard too and you will have had a direct hand in shaping the content of **/u/fortune** *news*.

P.S. Several of the surveys that were returned complained that we had printed the survey on a page opposite an article. This destroyed the article if the survey were torn out of the magazine. We originally intended that the survey be xeroxed and then sent in (77% of the respondents did just this). If you don't have access to a xerox machine and send the original page, we will be glad to send you back a replacement copy of Volume 6 Number 12 of **/u/fortune** *news*.❑

/u/help

Question: *I need to copy all the files from one hard disk on my system to another hard disk that I've just installed. I would like to preserve the current directory structure and links. What should I do?*

Answer: You are quite correct in assuming (hoping?) that there is an easy way of copying one entire disk to another disk. Actually, there are probably several ways of accomplishing this task but we will discuss just one of them here.

The approach that we take is a standard Unix solution which uses the standard Unix *tar* command. (If you don't have tar, our **fxfer** program can be used instead.) Tar, which stands for **T**ape **AR**chiver, is a utility that is often used to copy files from the hard disk to a tape drive. However, *tar* is a very general and useful command in that its main purpose is to copy some set of files from one place (e.g., hard disk, a single directory, a floppy drive, a tape drive, etc.) to another place (e.g., again a hard disk, single directory...). Thus, *tar* can be used to make an exact copy of one hard disk and write that information to another hard disk, which is the problem that we must solve.

The way *tar* normally works is to make a copy of a file or directory and then write that "archive" to some file or device. For example, if I wanted to make a copy of some user's directory (say, /u/joe) and put it into a file called joe.back, I could type:

```
tar cvf joe.back /u/joe
```

So, /u/joe specifies the directory to back up. The copy will be put into a file called joe.back. The sequence "cvf" is actually the three separate arguments c, v and f. The **c** arguments stands for "create" and archive. The **v** argument stands for "verbose" and causes tar to print some information about the files it is copying into joe.back. Finally, the **f** argument specifies the name of the file or device to write the archive to. Notice that this argument is followed by the word "joe.back" which is the name of a file. The above *tar* command illustrates how tar is normally used to create an archive. To extract files from an archive, one would use a command like: `tar xvf joe.back /u/joe`

Note that the only change here is that the **c** has been changed to an **x** which stands for "extract." Also note that we are being very brief here and just giving some basic background.

Now, to make a copy from one hard disk to another, we will use the following set of commands. Let's assume that you want to make a copy of the hard disk mounted on /h1 to the hard disk mounted on /h2:

```
cd /h1
tar cf - . | (cd /h2; tar xf -)
```

This may look a bit complicated but it is understandable - and better yet, it works! First, we change directories to the disk from which we want to make a copy. Then we start the tar command with "`tar cf - .`". There are two curves here. First, the "`f -`" means write the copy NOT to a file but to the standard output. The beauty of this is that we can use the Unix pipe (|) to send this output to some other command which in this case is another *tar*. The second curve is the "." which simply means "the current directory". Thus "`tar cf - .`" means make a copy of the current directory (and all subdirectories) and write this copy to the standard output.

The second part of the command, "`| (cd /h2; tar xf - )`" does the job of writing the information on the second hard disk. The pipe symbol sends the output of the first tar command to the second tar command which is "extracting" the files and writing them onto the second hard disk.

We know that it looks complicated but it does work and it preserves the directory structure along with all links. This is important since other methods of copying might not preserve links and thus you would end up with more information on the second disk than the first.

There's only one other thing that you might need to consider if y ou are going to use this technique. If you only have two disks in total, then your second drive is actually mounted as a subdirectory of your first drive, e.g. /u, /h, /b etc. If you copy from your root (/) drive to the second drive, it will eventually start to copy again inside of itself. In other words, you'll get a /h/h directory filled with files, and eventually a /h/h/h and so on. It won't be long before the disk is filled.

There are lots of ways around this problem. What we did was to unmount the second disk from it's normal directory, e.g.

```
umount /dev/hd12
```

Then we make a directory called /z

```
mkdir /z
```

Finally, we remount the drive as /z

```
mount /dev/hd12 /z
```

Once all that is done, we can use the shell metacharacters to copy all of the files in the root directory, except those that begin with z

```
tar cvf - [0-y]* | (cd /z;tar xf - )
```

That will copy everything except the /. files. Use

```
cp -tous /.* /z
```

to get those. That should do it.❑

Classified

This Classified section of **/u/fortune** *news* is designed to serve our subscribers and Fortune computer users in general. In order to help everyone use their Fortune hardware and software to its utmost, we are providing a forum for the selling and the buying of used equipment. We will be including this Classified section in each issue of **/u/fortune** *news* as long as there is sufficient interest. So, if you would like to find something for your Fortune or locate a buyer for your equipment, you should not overlook this valuable resource. If you are interested in a listing in this section of **/u/fortune** *news*, please call us at (617) 894-6900.

FOR SALE: 32:16 Package

Updated system with 70 meg Hard disk and tape backup. Includes 4 terminals plus console. Expanded memory and Fortune:Windows. Software includes Fortune:Word, Multplan and Guardian Calender program. Available in 30 days. Price: best offer. Call Judy at 319 234-5701. 10/89

Fortune 32:16 System

30MB hard disk with 256KB, 800K flexible drive, 4 ports, console terminal plus 2 Fortune Intelligent Work Stations (12" monochrome display). Software: Fortune:Word, Multiplan and Sequitur, plus Hayes Smart Modem, 300 baud. Form more information, call Laurie at (602) 795-8870. 11/89

Fortune XP 32:16 with 4 terminals

1 Meg of RAM, 70 MB hard disk, with Fortune:Word, Multiplan, Business Basic, all documentation.

Call Don at (617) 267-6500 for more information. 11/89

TWO Identical Formula 8000's

Hardware: 4 meg memory, 145 meg hard disk, 60 meg tape backup, 3 6-port Comm-a boards totalling 22 ports

Software: FOR:PRO, Development Utilities w/Programmer's Manual, Fortune:Windows, Fortune:Word, Ultracalc Spreadsheet

Also available: 16 new style Fortune Terminals, sound-proof cabinet for letter quality printer, Diablo 630 Printer with tractor.

Please call Mike Barrow at (802) 823-5811 for more information.
10/89

Fortune 32:16 XP 70

1 Meg Memory
13-70MB Hard Disks
Ports: 12 serial & 1 Parallel
60 MB Tape Expansion
Fortune Basic Workstation

Will take Best Offer
Call Rena at (301) 907-8900 10/89

SX 70 T

This unit has a 70 MB hard disk, 60 MB Tape streamer, 1MB of RAM and has For:Pro 2.0. THIS IS AN EXCELLENT OPPORTUNITY TO GET A TAPE AND UPGRADE TO AN SX! Call the friendly folks at **/u/fortune** *news* for a good deal - (617) 894-6900. 9/89

Fortunes For Sale - Priced Right!

Two - 32:16's (20 & 30 MB HD)
One - SX70T (2MB memory)

For:Pro 2.1, Multiplan, Ext. Fortune:Word BASIC 6.5.12, IDOL, BAS, Dev. Utils., Communications, other software. Call **Russ** or leave message at (708) 510-0088. 1/90

Fortune 32:16 10MB

Hardware:
This system has 1 Meg of RAM, 10 MB hard disk, 4 port Comm-A, Hayes 1200 Modem, Console and one additional workstation, IDS 132 dot matrix and Brother letter quality printers, all cables and books.

Software:
Includes Fortune:Word, Multiplan, some business applications, ITE, Basic, Pascal, C,Development Utilities and disks from **/u/fortune** *news*.

Call **Jim** at (212) 724-8114 for more information. 6/89

Used Fortune Parts For Sale

WD06
CRT Electronics and Tube
256KB Memory Board
800KB Floppy Drive
Comm A 4 Port Board
Video Controller
1 MB XP Mother Board
10 MB Hard Disk
20 MB Hard Disk (C20)
and more...

All components are fully functional, supplies are limited.

For more information, call **Jim** at **(203) 828-4811** 12/89

Fortune 32:16 XP

Hardware: 45 MB Hard Disk, 2 MB RAM, Terminal and keyboard, 4 ports

Software: For:Pro V2.1, Rubix

All Cables and Books included
Price: Best Offer
Call: Brad Hunter for information at (313) 296-7130 1/90

Fortune 32:16
1 MB Memory
30MB Hard DIsk
Fortune:Word
Will take best offer.
Call David at
(718) 965-9857 9/89

20MB 32:16 - Includes For:Pro 1.7.4.1, 768K RAM, console, 2 terminals, 4-port Comm A and a **20MB Tape Expansion Box.** Software includes: SMC BAS modules, Extended Fortune:Word, Developtment Utilities, and a C compiler. Purchased direct from Fortune, Hardly used, will take best offer. Call Gerry Armel at (508) 285-3040. 9/89

/u/fortune news
225 Crescent Street
Waltham, MA 02154

/u/fortune* news

FEBRUARY, 1990

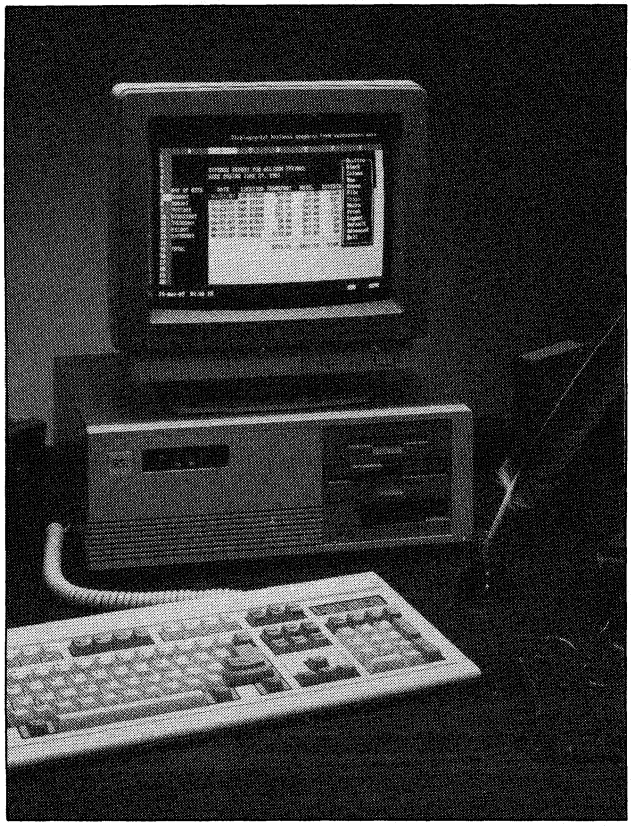
The Newsletter for Users of SCI/Fortune Computers January 1990 / Volume 7 Number 1

No Splash, just the Facts

Keeping your 32:16's hard disk going -- Page 15

Installing a Friendly Menu on Fortune 5000's -- Page 6

Two New Extensions To Our PC-Architecture



5286

Smaller Cabinet

* Reduced Price
* More Desktop Space
* Improved Price/Perfomance

5386

25 MHz

* Additional power for more users
* Faster Response in existing applications
* Improved Price/Performance

Fortune 5000, a family of products ranging from single user workstations and standalone systems to low end and medium scale multi-user UNIX systems.

Call your dealer for pricing and more information.

SCI FORTUNE TECHNOLOGY

/u/fortune *news*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz

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, Inc. are affiliated with SCI/Fortune Systems.

Subscriptions cost $60 for one year within the USA. For foreign subscriptions, add $36 each year for added postage and handling costs.

Back issues are offered at:

Volume 1, Numbers 1 - 8	$2/each - $14/set
Volume 2, Numbers 1 - 7	$3/each - $18/set
Volume 3, Numbers 1 - 12	$4/each - $44/set
Volume 4, Numbers 1 - 12	$4/each - $44/set
Volume 5, Numbers 1 - 12	$5/each - $55/set
Volume 6, Numbers 1 -	$5/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $15 each, quantity discounts available. 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 the contents of **/u/fortune *news***.

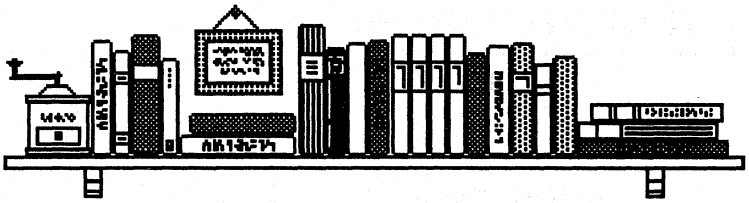
For more information, call or write:

The Cambridge Consortium, Inc.
225 Crescent Street
Waltham, MA 02154
(617) 894-6900

The text for this newsletter was prepared using an SCI/Fortune 5000 for both text input and typesetting. The typesetting was accomplished by using the Aldus Pagemaker program under VP/ix, which is a DOS emulation.

CONTENTS

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.

Question: The Basic Advisor has been quite helpful. Thank you for your continuing good work. We have encountered an inventory control problem that you may be able to address. The BAS software has LIFO, FIFO, and Average Cost options. We maintain our records on a FIFO basis and therefore put an "F" in field 21 of all Inventory Item Master records. The system will then assign cost on invoices using an average however, not true FIFO. Is this how the original software was designed? We have made several changes through the years. Perhaps we are the cause of this problem.

Answer: We cannot imagine one of our intelligent readers being the cause of any problems in BAS. BAS is a noble problem creator in its own right without your help.

Yours is only the second inquiry we have received concerning LIFO and FIFO lots processing in over four years. The first appeared in May, 1987 (/u/fortune news, volume 4 number 5), in which a very befuddled but heroically diligent lady wrestled with FIFO quantities which were not responding to memo updates. Her interest was in on-hand quantities whereas yours seems to be in sales analysis and profits. You may wish to review our response to her letter and make the suggested program fix on your system if you have not already done so.

It is obvious that LIFO and FIFO processing were added to the Order Processing and Inventory Control as an afterthought. The quantity adjustments were missing from memo update and now we learn the item costs are not properly adjusted according to LIFO and FIFO lot costs. I wish we could give you a simple fix, but this one is a bear.

Item costs are captured in the invoice detail file (CIVDT) during Direct Sales Entry and Order Invoice Entry The Order Entry detail file "COODT" does not carry costs. This system allows you to change item costs in the Inventory Master File after invoices have been entered or after orders have been invoiced without altering costs on items already shipped. In all cases the cost captured for each item is the Accounting Cost, or "average cost" from the Inventory Item Master File. Even though each of the six LIFO or FIFO lots has its own cost, these numbers are ignored during invoicing. In order to reflect LIFO or FIFO costs in lieu of accounting costs and maintain our ability to change costs after invoicing it would be necessary to alter all programs which write invoice lines. This would amount to more programming than we are equipped to handle within these pages. We would have to recommend you hire the services of one of the fine support companies listed last month. But there is something we can do with an invoice update program.

During invoice update the true LIFO/FIFO costs are calculated and used when updating the Month To Date and Year To Date cost (fields 22 and 24) in the Customer Master File (CCSMS) and the Customer Invoice Cost (field 14) in the Invoice, Debit Memo and Credit Memo Journal File (CIMRC). So your Sales Journal figures are calculated on the LIFO/FIFO costs which were in effect during the invoice update (as opposed to costs in effect during invoice or direct sales entry). Also, if you were to define an IDOL report using CCSMS showing "Profit (Sales Amount minus Sales Cost) it too would reflect true LIFO/FIFO costs. Your sales commissions and monthly sales analysis costs, on the other hand, are calculated during invoice and direct sales entry, so they represent accounting costs. There is not much we can do for sales commissions, but maybe we can play with sales analysis.

The invoice update is a two part operation. In part one (program CBIUA2) updates are made to the Inventory files as well as the Customer, Salesperson, Open Receivables and Sales Journal files. Part two (CBIUA3) updates the Open Order, Back Order and Monthly Sales Analysis files. Since LIFO and FIFO costs are calculated in part one, we can write them back to the invoice line item file before they are used again for part two of the update. (Because sales commissions are pre-calculated and then updated in part one, we cannot easily recalculate them.)

Feel brave? Try this. DO AN END OF DAY PROGRAM BACKUP!! (And keep in mind we are assuming you have pure BAS programs.) At any BAS menu type, in capitol letters, BASIC and press <CR> (the "Return" key). You will go to the BASIC console mode and see a > prompt. At this prompt enter the following in caps:

```
LOAD ''CBIUA2''    <CR>
99 IOLIST A2$,B2$,C2$,D2$,A2,B2,C2,D2,E2,F2,G2,H2$,I2$,H2,I2, J2$,K2$    <CR>
1125 READ (2,KEY=A2$,ERR=8000) IOL=99     <CR>
2085 WRITE (2,KEY=A2$,ERR=8000) IOL=99    <CR>
SAVE    <CR>
RUN ''DOL''    <CR>
```

This procedure should make all of your sales analysis costs reflect the true LIFO/FIFO costs effective at the time of invoice update. If you get into trouble trying to make the changes, DO NOT TYPE "SAVE"! Just RUN "DOL" and you are safe. If you do happen to destroy the program, restore your programs with a START OF DAY.

Question: We are a dealer and purchase our Business BASIC from Thoroughbred as well as other manufacturers. Recently one of the salesmen from a Thoroughbred competitor told us that Concept Omega Corporation was in financial difficulty and may even be on the block. Would you mind checking on this for us?

Answer: I'll bet all of this was told to you in strict confidence. I promise I won't tell anybody.

What we think happened is that Concept Omega enjoyed a little too much success in 1988. If you recall, they made number 20 on Inc. Magazine's list of fastest growing privately owned companies in December of that year. As a result, the venture people who helped John-L Johnson buy Thoroughbred from Science Management Corporation in 1985 decided it was time to collect their profits and head off to sunnier climates. This sudden departure of capital created a need for some "painful cutbacks"; we lost a few personal friends when the sales division was replaced with a simple order desk. But that's progress.

At one point in November or December of 1989 John-L was talking with a bank or two, but it turned out he didn't need the help. This is probably where the "on the block" rumor started.

We cannot believe Concept Omega is in trouble. Their sales were around $15 million for 1989, up from $10-12 million the year before. Furthermore they showed a profit for the fourth quarter and probably for the whole year. Concept Omega is about twice the size of the company for which your source works. Maybe all of this is just a bit of sour grapes. Nevertheless, it is good to know that competition exists, just in case. I get a warm fuzzy feeling knowing my BAS programs will always have a BASIC interpreter.

Question: We are now a beta test site for Solution-IV Inventory Control, Order Processing and Payroll. Back in November you quoted me as saying the packages will not be released until late 1990 if that soon. We received our beta packages early this year and feel that based upon testing needed there is no reason why they cannot be released in the third quarter or at least by the end of the year. - Steve Rosenfeld, Superior Computer Systems.

Answer: That's good news. We have already purchased copies of General Ledger, Accounts Receivable and Accounts Payable. If the three you mention are as good as these, we are certainly looking forward to their release. We will report in more depth on Solution-IV if our readers voice an interest.

Question: Please correct our support company listing in your January article. Our phone number and billing rate were wrong.

Answer: Our apologies for the error. Your listing should have read:

UNICONCEPTS, Englewood, CO. Contact Tony Piacente at (303) 340-3355. $60.00 per hour.

Dear Mr. Wannall, Well, here is a predicted request for change in your future listing of support companies for IDOL/BAS. We received the January, 1990, volume of /u/fortune news yesterday.

DayProm Computer no longer supports this software. Possibly you sent a request for updated information to DayProm before you printed the list and it never got forwarded to us. I hope this doesn't cause confusion for our customers who subscribe to the newsletter (although DayProm still refers their calls here).

Our company has been in the business of supporting BAS/SMC/PAC and BASIC for a number of years. On May 15, 1989, Tim Furman, our President, officially took over software support of Fortune customers from DayProm Computer, Inc., in Dayton, Ohio.

This situation also gives me a chance to write and tell you we both enjoy reading your column! (You have a way with words and often make me laugh.) I always find something of interest. Your recap of the 1980's was particularly interesting to me. Sincerely, Rosanne Furman, Vice President.

Dear Ms. Furman: We thank you for your kind words and hope you are planning to send in your survey form to the publishers of this newsletter.

For future support company listings we will include something akin to the following:

EXCELSIOR COMPUTER SERVICES, LTD., Kettering, OH. Contact Tim Furman or Rosanne Furman at (513) 294-6651. Billing rate is in line with that of DayProm Computer, Inc., we think.❑

System Administration: Part 34

No Longer requires a Degree in UNIX Wizardry

This series is a must for all owners and users of the SCI/Fortune computer. Dave is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the SCI/Fortune computer, and is past president of the Houston UNIX User's Group. He contributes independently to **/u/fortune *news***.

Creating Friendly Menus on a 5000 *by Dave Kloes*

As promised, in the next few issues we will present two main shell programs that we install on our client systems. One is a front end menu for the System Administrator and the other is a front end menu for normal users. We will also show you a few of the smaller programs that are executed from the main programs. It has been at least a couple of years since we presented a series of articles on shell programming so we will also go through these shell programs as a refresher. For those that are interested in learning more about shell programming, you can order our book entitled "The New Shell Game - in UNIX/XENIX". The cost of the book, including freight, is $24.95. See our ad in this issue for the mailing address.

First of all, you need to create a new login account that will be used to access the menu. Use the "mkuser" command (or the user friendly "sysadmsh" utility) and create a login using the account name "admin". In setting up this account, specify the following:

1. use next available id? - y
2. use default group? - y
3. enter password
4. select standard (Bourne) shell

After you have added this new account, change directory to "/etc". Make sure the "passwd" file has the correct permissions:

```
# chmod 644 passwd
```

NOTE: This procedure is not for the light hearted. If you are not familiar with a Unix editor or the "/etc/passwd" file, we DO NOT recommend that you do this procedure.

Edit the password file using "vi", "ed" or any other Unix editor and modify the line for "admin". Change the user id number and group number to "0". The line should look something like this when you are finished:

```
admin:yu3r2xwz9Z5tW:0:0:comment:/usr/admin:/bin/sh
```

In effect, we have created a new login account that has all of the powers of "root". In order to execute the items that we will be presenting on our menu, we will need root permissions. We prefer to do this rather than to mess around with the normal functioning of the "root" login.

When this new account was created using "mkusr", the home directory and the ".profile" file for this account were automatically created. Remember that the ".profile" file is automatically executed for that user whenever they login. Using "vi", "ed" or any other Unix editor, add the following lines to the "/usr/admin/.profile" file. Note that the line numbers are not part of the script and are for reference only:

```
1   # This is appended to the .profile file
2   trap '/usr/bin/kmenu' 1 2 15
3   /usr/bin/kmenu
4   clear
5   kill -9 0
```

Line 1 is a comment - remember that any line that begins with a "#" is a comment line.

The "trap" in line 2 is designed to trap the interrupt key. Anytime the user depresses the interrupt key (usually the 'del' key), the command between the " is executed. In this case, we will be creating a shell script in "/usr/bin" called "kmenu" so this program is rerun anytime the 'del' key is depressed. Essentially, this brings the user back to the main menu and redraws the screen. If we did not do this, the shell program would stop and the user would be at the Unix prompt - something we do not want to happen.

Another advantage of writing the program this way is that we can return to the main menu from any prompt by depressing the 'del' key. This saves us some programming time since we do not have to give the user the option to 'exit' every time we prompt for more information. Even if we are in another program that branches off the main menu, the 'del' key can always be used to return to the main menu.

Line 3 executes our "kmenu" program which we will be presenting in just a moment.

When the user selects "Exit Menu" from our main menu, control is passed back to line 4 in the .profile file. The screen will clear and the "kill -9 0" in line 5 will log the user out.

For those of you that want to use these menus but do not want to create a login account, do the following:

1. Using a Unix editor, create a shell program in the "/usr/bin" directory and call it "kstart".

2. Put the command lines above in the "kstart" program.

3. Login in as root. You can execute the "kmenu" program by entering "kstart" (be sure you have given it execute permissions - chmod +x kstart). This would be the same as executing the program from the .profile file. If you want to remain at the Unix prompt and not be logged out of the system, remove line 5 from the script.

Now that we have the inital startup script written, we can look at the "kmenu" program:

Script name: kmenu

```
 1    ext=n
 2    while test "$ex" = ""
 3    do clear
 4    echo "                                  Uni-Komp Menu
 5
 6    Your current working library is: `pwd`
 7
 8    Depress <del> to exit at any prompt.
 9
10      1.  System Administration Menu.
11      2.  Daily Backup (/usr directory only - tar).
12      3.  Weekly (Friday) Backup (full system backup - dump).
13      4.  Shutdown system.
14      5.  Link New User Into Menu System.
15      6.  Callout on the modem.
16      7.  Games Menu.
17      8.  Format Floppy Disk.
18      9.  Backup /usr Directory to Floppy Disk.
19     10.  'tar' Backup of Individual Files/Directories.
20     11.  Exit Menu.
21
22    ENTER YOUR SELECTION: __^H^H\c"
23    read sel
24    case $sel in
25            1) clear
26               /usr/bin/sysadmsh;;
27            2) clear
28               echo "
29    Backup of /usr files ...
30
31    Insert tape and depress <RETURN> to continue: \c"
32               read bunk
33               if tar cvfbk /dev/rctmini 20 37000 /usr
34                  then echo "
35    Backup successful."
36               else echo "
37    Backup not successful."
38               fi
39               echo "^GDepress <RETURN> to continue: \c"
40               read bunk;;
41            3) clear
42               echo "
43    Backup of entire system ...
44
45    Insert tape and depress <RETURN> to continue: \c"
46               read bunk
47               if backup 0ufk /dev/rctmini 39000 /dev/root
48                  then echo "
49    Backup successful."
50               else echo "
51    Backup not successful."
52               fi
53               echo "^GDepress <RETURN> to continue: \c"
54               read bunk;;
55            4) clear
56               /etc/shutdown;;
57            5) /usr/bin/kprofile;;
58            6) clear
59               /usr/bin/kcallout;;
60            7) /usr/bin/kgames;;
61            8) clear
62               /usr/bin/kdisk;;
63            9) clear
64               echo "Insert first FORMATTED floppy disk.
65    Depress <del> to exit or <RETURN> to continue: \c"
66               read bunk
67               if tar cvfbk /dev/fd096 1200 20 /usr;;
68                  then echo "
69    Backup successful."
70               else echo "
71    Backup not successful."
72               fi
73           10) /usr/bin/kback;;
74           11) exit;;
75            *) echo "^GNot a valid response.
76    Depress <RETURN> to continue: \c"
77               read bunk;;
78    esac
79    done
```

Before we get into the program and dissect it a little, let's look at the menu and a brief description of each of the items on it:

```
**********************************************************************
                          Uni-Komp Menu

Your current working library is: /usr/admin

Depress <del> to exit at any prompt.

  1.  System Administration Menu.
  2.  Daily Backup (/usr directory only - tar).
  3.  Weekly (Friday) Backup (full system backup - dump).
  4.  Shutdown system.
  5.  Link New User Into Menu System.
  6.  Callout on the Modem.
  7.  Games Menu.
  8.  Format Floppy Disk.
  9.  Backup /usr Directory to Floppy Disk.
 10.  'tar' Backup of Individual Files/Directories.
 11.  Exit Menu.

ENTER YOUR SELECTION: __

**********************************************************************
```

We will discuss each of these items in more detail as we go through the program but for now, here is a brief description of what each of them will do:

1. For those that feel more comfortable in the user friendly Xenix administration menus, this selection will execute the "sysadmsh" command.

2. We will discuss backup philosphy in detail later on but for now, this selection would be executed as our backup program on Monday through Thursday. It does a "tar" backup of the user data files.

3. Selection 3 does a complete system backup of the "root" partition. It uses the "dump" command which is filesystem

specific. Note that you may have to add more selections to your menu - one for each filesystem or partition on your hard disk.

4. Selection 4 runs the "/etc/shutdown" command. We will have to modify the "shutdown" shell program to allow our "admin" account to use it since it checks to see if you are logged in as "root" or "sysadm" before it will actually shut the system down.

5. We will be creating a "prototype" file that will be added to a user's ".profile" file to link them into the user's version of "kmenu". Obviously, the "kmenu" we are discussing now is not intended for the everyday user. The user menu will allow the user to run application programs such as word processing, spreadsheet and database. We will discuss this in more detail when we present the main menu for other users on the system. After we create the prototype file, this selection will be used to automatically update the .profile file for any user on the system. When the time comes, we will show you the "kprofile" script that goes along with this selection.

6. This selection will be used to callout on a modem that has been set up on one of the serial ports (if you have one). We will be using "kcallout" as a follow-on program to this menu selection. Our system has a more extensive menu system here that will actually use a modem port for dialin and/or dialout at different baud rates. It sends all necessary sequences to the modem and enables or disables the port, as required. Obviously, all of that is a topic for another issue ...

7. We find that our clients usually end up taking the Games Menu off of the menu system. Either people end up spending too much time "playing games" or they get into one and can't get out which is one hassle the System Administrator does not need. In any case, we will present a simple menu and script (kgames) for the games.

8. Here we will give you another menu that will present selections for formatting floppy disks (kdisk). It will include different media types such as high density, 360K floppy, and the 3 1/2" floppy. It will also allow these selections if you have a second floppy drive installed.

9. Selections number 2 and 3 assume you are using a tape device on your system. Selection 10 allows the daily backup to be done on floppy disk. It is only provided for use as a failsafe in case something happens to your tape drive.

10. This selection is provided for occasions where you want to back up single files or directories on floppy disk. Another script will be presented that prompts for the files and/or directories to backup (kback).

11. Obviously, this selection is used to exit the menu.

Speaking of exiting, our space has run out for this issue. Next time we will talk about the script itself. Knowing how some of you type, it may take you that long to key it in. For those of you that are not typists and want to cheat, you can contact The Cambridge Consortium for a diskette that has these programs on it.

If you do decide to enter the programs and use them, remember that we have not discussed the individual commands in the scripts yet. The configuration of your particular system may dictate that some of the command lines be changed. For example, the tape backup selections assume you have a mini-cartridge tape drive. The particular backup commands for "tar" and "dump" would have to be changed in order for the script to work properly. We will be discussing some of these topics in the issues that follow.❑

Postscript printing from a Fortune -- Can it be done???

A reader recently wrote to ask us if it was possible to print on a Postscript printer with a Fortune computer. Postscript is a "page description language" that is very convenient for desktop publishing. It is widely used with programs like Pagemaker or Ventura Publisher, and in fact it's how we create /u/fortune news.. We use a NEC LC-890 laser which can emulate an HP Laserjet or print with Postscript, so it's compatible with most of our work. Here's the answer to the reader's question.

Answer: We have given this one some thought and the answer is yes, and no, and we have some ideas for solving this problem completely. But, let's give our readers some information now.

We have been able to make a very simple printcap entry which allows the printing of text files on a postscript printer. Note that this does not work with Fortune:Word - only with the lpr command. The printcap entry is reproduced here:

```
PSCR|POSTSCRIPT:\
        :is=/Courier findfont 12 scalefont setfont\n\
        20 720 moveto\n/vpos 720 def\n/hpos 20 def\n(:\
        :ff=)showpage\n20 720 moveto\n/vpos 720 def\n(:\
        :cr=) show\n/vpos vpos 15 sub def\nhpos vpos moveto\n(:\
        :ej=showpage\n:
```

One should copy this into /etc/printcap exactly as shown. Then, set up a new printer in the /etc/devtype file with the PSCR type. Then, you can use the lpr command to send text files to your printer. Once again, this will not work with Fortune:Word files. The solution for sending Fortune:Word files to a Postscript printer is a bit more complicated but we have some ideas as to how it might be done. We will discuss one idea later in this article.

How the Printcap Entry Works

The above printcap is very simple and really only uses the most basic capabilities of the printcap system. However, it does allow one to send text files to a printer which uses the postscript language. Let's decompose the printcap entry so that our more ambitious readers can modify it, if they wish.

First is the "is" variable which stands for "initialization string" and we use this to set up some basic information that a postscript printer will need. The following lines are part of the initialization string:

```
/Courier findfont 12 scalefont setfont\n
20 720 moveto\n
/vpos 720 def\n
/hpos 20 def\n(
```

The first line specifies the type of the **font** we will use (i.e., Courier) and the size of the font (i.e., 12 points). Naturally, these parameters can be modified as long as the particular printer being used has the desired font. So, for example, if you wanted to have the output be in Times-Roman at 11 points, you would use:

```
/Times-Roman findfont 11 scalefont setfont\n
```

The second line, 20 720 moveto\n, positions the printer at the top of the page. The numbers 20 and 720 are in "point" units where there are 72 points to an inch. Also, the **moveto** command in postscript is used to move the "print head" to a particular location on the current page. Thus, 20 and 720 refer to a coordinate on a 8.5 by 11 inch page relative to the bottom left-hand corner of the printable surface. There is nothing magic about the 20 and the 720 except that it does position the "print head" at the "top of the page." These parameters could be changed to leave more room on the left margin (e.g., change the 20 to a 36 for 1/2 inch margin) or more room on the top (e.g., change the 720, which leaves about an inch on the top, to 684 to leave about an inch and a half).

The third and fourth lines are used to set up two variables. One is called **vpos** which stands for "vertical position" and we set this to 720 to record the current vertical position of the print head. This parameter, **vpos**, will be decremented each time we put out a new line. The next parameter, **hpos**, stands for "horizontal position" and records the location of the left margin. Both of these parameters are used below in controlling where each line of the output will be located on the page.

One final thing to note about the fourth line is that it ends in an open parenthesis. This is critical because all character output that is sent to a postscript printer is enclosed in parentheses followed by the word "**show**". For example, if we wanted the sentence "This is a sentence." to appear on a postscript printer, we would need to send the following:

```
(This is a sentence.) show
```

The open parenthesis is setting us up to handle the incoming text! You will see that some of the following commands will close the text and output it.

Looking back at our postscript printcap entry we see several more cryptic lines. One begins with "**ff=**", another with "**cr=**" and the last with "**ej=**". The way the printcap system works, at least on a simplified level, is to key on certain special characters that are assumed to be in the information that is sent to the printer. One special character is the "formfeed" (i.e., ff) character which tells the printer to end the current page and start a new one. Another special character is the "Carriage Return" (i.e., cr) character which tells the printer to end the current line and begin a new line. Finally, a third special character is the "eject" (i.e., ej) character which generally signals the complete end of a document.

In order to build a printcap entry, we must specify what the postscript printer should do for each of these special characters. For the formfeed character we specify the following:

```
:ff=)showpage\n20 720 moveto\n/vpos 720 def\n(
```

In essence, when we see a formfeed character we want to do three operations. First, we want to close off the text. That's why the command begins with a close parenthesis. Second, we want to print that page, which is what the "**showpage**" postscript command does. Finally, we want to reposition the print head on the top of the next page and reset the **vpos** variable, which is what the "20 720 moveto\n/vpos 720 def\n" does.

The next character to take care of, and the most complicated of this printcap entry, is the carriage return character. For this character, we specify the following:

```
cr=) show\n/vpos vpos 15 sub def\nhpos vpos moveto\n(:\
```

This entry does three things as well as follows:

```
) show\n
/vpos vpos 15 sub def\n
hpos vpos moveto\n(
```

First, it closes off any text with a close parenthesis and then puts this text onto the page (i.e., the "show" command). Second, we reset the **vpos** variable which you will remember represents the "vertical position" on the page. The second command in the above display simply subtracts 15 from the value of **vpos** and then resets **vpos** to this new lower number. For example, when we begin a new page, **vpos** is set to 720. The first line of the output will be at this vertical position. The second line of the output will be at a vertical position of 705. The third line at 690 and so on until a formfeed character is encountered which resets everything.

The final line of the display (hpos vpos moveto) does the repositioning to the left margin of the new line where the new text will begin and finally it leaves an open parenthesis in anticipation of the new text.

Lastly, we must deal with the "eject" character which we do simply with:

```
ej=showpage\n:
```

This just issues the postscript "showpage" command which ejects the page which is being processed.

Public Domain Postscript Printing Program

The above postscript entry is very basic and although it would allow one to print a text file on a postscript printer it is not very flexible. We have found a public domain program which is designed to print text files on a postscript printer and this program provides the user with more control over the printer. The program is called **lwf** which stands for "Laser Writer Filter" and we have compiled it so that it works on the Fortune 32:16, Formula and the 5386.

In general, this program give you the ability to control several aspects of the printing. For example, lwf provides an easy mechanism to control the font size with the -s flag. The lwf command to print the file called file.txt at 10 points would be:

lwf -s 10 file.txt

Several other attributes can be controlled as listed below:

-c	Multiple Copies are printed
-i	Control of indented lines
-l	Print in landscape mode
-o	Print only certain pages
-p	pr is used to paginate and print page headers
-t	controls the tabstops

If you are interested in this program, it is available on disk and costs $15.

Fortune:Word Files On A Postscript Printer?

We are not able to design a printcap entry that will drive a postscript printer. The process is very complicated and involves, to the best of our knowledge, the interaction of several built-in programs that are a part of the Fortune:Word system.

However, given our capabilities of translating Fortune:Word files to Word Perfect, we feel that the way to solve this problem would be design a program which would "translate" a Fortune:Word file to a "postscript" file. We feel that we could produce a program which would allow Fortune:Word files to be output on a postscript printer but it would not be trivial. At the current time, we don't feel there is enough interest in a product of this sort.

However, we would welcome the comments of our readers with regard to this capability and if there is sufficient interest, we would consider programming a "Fortune:Word to Postscript" converter.❑

Mark Palmerino

News From SCI/Fortune

Fortune 5386 25 MHz System Available

The Fortune 5386 is now available in a 25 Megahertz (MHz) version. Operating at a 25% faster clock rate than 20 MHz systems, the Fortune 5000 System Model 5386/25 incorporates the following standard features:

- Vertical cabinet supporting a maximum of 5 half-height devices
- 1.2 MB 5.25 inch flexible diskette drive
- 150 MB cartridge tape drive
- Enhanced 101-key keyboard
- 250 watt power supply
- Clock/Calendar with battery backup
- Monochrome/CGA Video Adapter

The system motherboard features the Intel 80386-25 microprocessor. The design of the motherboard includes the following features:

- Flexible diskette drive controller
- 7 16-bit expansion slots
- Socket for optional 80387 math coprocessor
- 25/8 MHz keyboard selectable operation
- 2 serial ports
- 1 parallel port
- Phoenix ROM BIOS PLUS

The 32-bit memory board provides 4 MB of dynamic random access memory, expandable to 8,10, or 16 MB. Hard disk drives are available in 70 MB, 150 MB, 322 MB, and 660 MB (formatted) sizes. A 3.5 inch microfloppy is also available.

Maxpeed Intelligent 8-port Controller Available for Fortune 5000

SCI/Fortune is pleased to announce the availability of the Maxpeed SS-8 Intelligent Serial Controller. The Maxpeed SS-8 Intelligent Multi-Serial Controller is an intelligent eight-port serial I/O controller for Fortune 5386 and Fortune 5286 systems running INTERACTIVE 386/ix (2.0.2) or SCO XENIX/UNIX operating system software.

Product Features

The Maxpeed SS-8 Intelligent Multi-Serial Controller has the following features:

- Provides eight high speed 38,400 baud serial communication chanels per SS-8 board.
- Provides aggregate communication capacity of 300,000 bps across eight channels per board.
- Significantly off-loads the main CPU from serial communication tasks.
- Fully supports terminals, modems, serial printers, and other serial devices.
- Operates with inexpensive telephone jacks and cords.
- Works with 16, 20, and 25 MHz 80286 and 80386 computers running the high speed 16/12.5/10 MHz bus.
- Fully supports INTERACTIVE 386/ix (2.0.2), SCO XENIX/UNIX, and VP/ix software.
- Supports 32 channels with four SS-8 boards.
- Supports 32 transparent printers from terminals.

The controller includes one 25' four wire cable withi a terminal/printer adapter, as well as user's manuals ans software drivers for both INTERACTIVE 386/ix and SCO XENIX/UNIX. Additional cords and adapters can be ordered separately.

Maxpeed recommends using four wire cords for terminals and printers and six wire cords for modems. Using four wire phone cord for terminals and printers eliminates the potential noises coupled through the unconnected DCD and DTR liens. The Maxpeed SS-8 Controller provides a lower cost, higher performance solution for adding additional serial ports to the SCI/Fortune 5000 series of computers.

The following questions/answers were provided to us from the technical support staff at SCI/Fortune:

Question: I have an ST-506 style hard disk drive with 1170 cylinders and 7 heads. I want to use it on my SCI/Fortune 5286 system, but I don't know what drive type number I should use. Is it possible to use this drive on my 5286 system. (editor's note: This issue may also apply to the Fortune 5386 systems as well.)

Answer: Yes, with these considerations: When a drive is installed in a 5286 system, the CMOS Setup parameter for the drive must be changed to reflect the drive's type. Determining the drive type requires some knowledge of the drive's geometry (number of cylinders, number of heads, number of sectors per track, and landing zone). Normally the drive type

See **News**, page 14

System V vs. For:Pro

What are the differences between UNIX on a Fortune 5000 and a 32:16?

With the advent of Fortune's new line of Fortune 5000 computers, new and different operating systems became available. The current line of 32:16 and Formula computers use the same variant of the Unix operating system developed by Fortune and called For:Pro. The new 5000 line of computers allow the user to select between two other variants of Unix (XENIX and System V) and DOS.

In this article, we wish to discuss some of the basic differences between For:Pro and System V Unix as implemented by Interactive Corp. We have had access to one of Fortune's 5386 machines courtesy of SCI/Fortune for over a year and have had the opportunity to explore the System V Unix in both the 1.06 and 2.02 releases.

I often think of the variants of Unix available today as being like dialects of a language - say English. Of course, the analogy isn't complete and I should say at this point that I hope that our readers in England will forgive me if I am not entirely accurate in some details. However, the difference between For:Pro and System V might be a little like the difference between "American" English and "British" English.

Perhaps the main point I want to make is two-fold. First, even though there are differences between the two dialects, speakers from either dialect can understand one another almost perfectly. That is, most of the words in either dialect mean the same thing, are spelled the same way and connect up with other words in the same grammatical fashion. Second, where there are differences (e.g., in spelling color vs. colour), individuals from either side can quickly become accustomed to these differences and, with practice, become as adept at using the other dialect as their own.

What I would like to do now is to list a couple of the differences in dialect between For:Pro and System V Unix. The first few examples will be some of the first differences that we noted when we first began using the 5386 computer.

Differences in ls

Perhaps the first thing we noticed was how differently the basic form of **ls** worked on the 5386 as compared to For:Pro. In Unix, the **ls** command is used to print the names of the files in the current directory. In For:Pro, when one uses the ls command without any arguments, one receives a listing of filenames in several columns on the screen. This allows many files to be displayed on the screen at one time. However, in System V Unix, the default format is to list the files in one column.

Of course, it is very easy to produce a multi-column output in System V by simply typing **ls -C**. Further, it would be easy to set up a shellscript that would do this for you. And, in fact, in the 2.02 version of System V from Interactive, there is a command, **lc**, which does just this. Thus, I've become quite accustomed to issuing the **lc** command when I use System V and **ls** when I use For:Pro.

Another difference with the listing of directories was the absence of the **ll** command in the 1.06 version of System V. The **ll** command in For:Pro lists a "long" version of the directory entries which gives more information about each file (e.g., file size, date of last modification, file permissions, etc.). Again, one could simply make a shellscript called ll with the following line in it:

```
ls -l $@
```

After the file was created then it would be made executable by issuing the following chmod command:

```
chmod +x ll
```

Finally, you would put this command in one of the directories in your path.

Pg vs. More - Viewing the Contents of Files

One common activity is viewing the contents of files. In For:Pro, the command of choice for this is called **more**. **More** displays the contents of file on the screen one screen-full at a time and then waits for the user to hit the space bar for the screen-full. On the original 1.06 version of System V, the **more** command was not available. An alternative, called **pg**, was available and it worked in almost the same fashion. One difference in the way that **pg** works is that it uses the RETURN key to signal that the next screen-full of information is desired. With **more**, the RETURN is used when you wanted to see just one more line of information.

Since I was used to using **more** I created a shellscript called **more** which simply called the **pg** command. However, the difference became less problematic when 2.02 was released since it included the **more** program which is in all relevant points identical to the **more** command under For:Pro.

Differences With cp

The **cp** command on the two versions of Unix are actually quite different. Basically, **cp** in For:Pro is a richer command. For example, the usage statement for **cp** in For:Pro is shown below:

Usage: cp [-abBcCdEilIMnNoprRsStTuvVxX] file1 file2
or: cp [-abBcCdEilIMnNoprRsStTuvVxX] file ... directory

This indicates that there are 26 different flags that modify what **cp** does. Contrasted with this is the fact that there are no flags for the **cp** command under System V.

The differences here very clearly highlight two different philosophies with regard to how commands should be constructed in Unix. The **cp** command in For:Pro is a prime example of the philosophy that one command should do many related things whereas the **cp** command under System V is an example of the philosophy that one command should do one simple thing.

For example, **cp** under For:Pro will "link" files one to another (e.g., cp -l). It will also perform complete system backups (cp -B) and system restores (cp -R). Under System V, linking is reserved for the **ln** command and system backup and restores can be performed by several other commands including **tar**, **cpio**, **backup** and **restore**.

Conclusion

These are some of the differences between For:Pro and System V Unix. Of course, there are many others and we will continue to describe some of the major differences and how users can best cope with them as they move from one "dialect" of Unix to the other.❑ **Mark Palmerino**

News, continued from page 13

determination is made by comparing the drive's geometry with drive types listed in the Setup drive type table and choosing a matching type. This works fine if you have a drive that matches a table entry. If the Setup routine allows you to select drive type 48 or 49, you can enter the drive geometry parameters under those types.

If the Setup routine does not allow selection of drive type 48 or 49, then you must select a drive type that closely matches the geometry of your drive. When selecting a drive type using this method, be careful not to choose a type that has more heads or cylinders than the drive you are installing. In your particular case, with this particular drive's geometry, you should choose drive type 19 which is 1024 cylinders by 7 heads. Because the drive's geometry cannot be accurately defined, you will have to accept some decrease in the drive's data storage capacity. The loss of capacity for this particular drive is approximately 8.9 megabytes.

A kit is now available for users who require the newer ROM BIOS version that allows selection of drive types 48 and 49, as discussed above. The new BIOS should be available from your dealer.❑

Using fxfer to Transfer Multiplan Spreadsheets to DOS

In order to transfer a **Multiplan** spreadsheet to a computer other than a Fortune, it is necessary to save it in the **SYLK** or **Symbolic** format. To do this, select **Transfer Options**, and then **Symbolic**. When you then save your spreadsheet it will be saved in this special format.

SYLK files are plain ASCII files which describe each cell of the spreadsheet. Many other programs can read SYLK files, including **Multiplan** on the PC, **Excel** on a MAC or PC, etc. Suppose your original spreadsheet was called **Quarter1**. In the native Multiplan mode, the spreadsheet file has a the name **Quarter1.MP**. All Multiplan spreadsheets have the .MP extension. When it is saved in the SYLK format, it is called **Quarter1.SL**. The .SL extension indicates that it is a SYLK file.

The SYLK file can easily be transferred to a DOS machine using our **fxfer, floppy transfer**, program. Using the menu version, just choose the Quarter1.SL file to write on the Fortune. On the PC, just read the disk with the DOS conversion turned on, and your file will be copied. If you have BASIC fxfer, on the Fortune you'd type **fxfer cv Quarter1.SL**. (The cv stands for create verbose.) On the PC, you'll type **fxfer xvYF 0**. (The xv stands for extract verbose. The Y instructs fxfer to convert the text file to a DOS file, and the F 0 says you're reading a Fortune floppy.)

Once the file is on the PC, you can read it with Multiplan or Excel. With Multiplan, you need to use the **Transfer Options Symbolic** instruction again to specify that the file you will load is a SYLK file. Excel will automatically know it's SYLK when it begins to load it.

If you want to transfer to **LOTUS 123**, both Multiplan and Excel on the PC can save a file in Lotus format. However, since Lotus does not support external links to spreadsheets, they will not be preserved. Multiplan is even economically priced, so it could be justifiably purchased simply as a conversion program.

If you want to transfer files back to the Fortune Multiplan, the reverse process should work fine. However, Fortune's Multiplan is limited in terms of spreadsheet size, so you won't be able to transfer large spreadsheets or macros.❑

/u/help

Diagnosing and Repairing Hard Disk Problems

Question: I have a question for your /u/ help! column. A 32:16 we have often reports hard disk errors, even when the system is sitting idle (except for its self-maintenance). When the hard disk is tested by a Field Engineering disk, moreover, after a certain point is reached in the scan, hard and soft errors are reported literally by the hundreds. Since the errors first occurred when we received the system back from a dealer (for an unrelated floppy controller repair), we wondered whether the dealer had accidentally forgotten to perform the head-parking shutdown procedure. (Or, since this system is sensitive to power irregularities, perhaps the dealer did not notice that the shutdown procedure had possibly been interrupted by a power fluctuation.) After we returned the system to the dealer because of the hard disk errors, the dealer stated that the problem was really an electrical short due, I believe he said, to the cpu cover. (The dealer also said that we should never open the cpu cover unless we wanted to assume full responsibility for any repair.) Since the system did have a brief period without any hard disk errors, we wrote the dealer thanking him for correcting the problem -- only to have the problem reappear very shortly, with full vehemence.

Do you have any thoughts on this matter? Is it possible that reported hard disk errors (reported during normal usage as well as under the control of the Field Engineering program) could really be evidence of an electrical short? And could the cpu cover really be responsible for such an electrical short? If it is, what part is shorting out and what should we do to prevent the short (we have yet to open the cover after the dealer's warning)? Or should we contact one of your mail-order dealers for a new or used C20 disk?

We have not been overwhelmed with Fortune dealers. When I exchanged a practically brand new AT&T UNIX PC for the well-used System 20 described above, the System 20 had a COMM A board without needed chips and a floppy disk or disk controller that would not allow complete reading/writing of floppies (though the repairman insisted the floppy was bad -- until I read and wrote it fully on another Fortune while he was taking a phone call). In addition, we were told that the motherboard had been replaced after one of these repairs when there had been no replacement (I had marked the microprocessor chip and mentioned the fact to the repairman, who then said he had really meant he had replaced only the floppy controller on the motherboard), and then had this problem with the hard disk (had the disk been scarred by faulty shutdown or, even worse, had the disk been swapped with a faulty disk after copying the files?). Perhaps all of the above may be ethically explained -- for the dealer's sake I really hope so -- but I am uneasy leaving a system with any dealer for "repair". I am more inclined, instead, to order the parts by mail and install them myself, perhaps with phone support -- or have the system repaired by a dealer only when I may personally watch. After a while, one becomes weary with the Fortune and its limited dealerships (though some of the ads in the News seem to appeal on the basis of thorough and ethical service).

It is perfectly normal in the life of a hard disk to develop "bad blocks". ..Generally, the problem can be corrected by sparing out the offending block, and either reentering the data, or reloading software.

Thank you for being one source I feel we can fully trust.

Sincerely,

Harry K. Tressel

Answer: Well, to begin at the end, thank you for your flattery and vote of confidence. Hopefully we can live up to your trust.

Your letter addresses several issues which we shall deal with one at a time.

Hard disk problems:

As you may know from following our columns in the past, we've had our share of hardware problems ourselves. Sometimes they're easy to find, and sometimes they're very tricky to track down. Clearly a problem that is consistently there is simpler to attack than an intermittent problem here and there.

It is perfectly normal in the life of a hard disk to develop "bad blocks". The hard disk is coated with a magnetic media which is read by a head which floats across the disk without actually contacting the surface. This is one of the differences between a floppy disk vs a hard disk. Because there isn't actually contact, the heads can move much more quickly without creating wear on the disk. When a disk is formatted, tracks are laid down in concentric rings which determine where your data will end up. The heads will physically move to a specific cylinder and read the requested data.

Bad blocks can emerge for many reasons, but two of the more common reasons would either be an actual physical marring of the disk surface which might have been caused by actual head

contact to the disk (or a trapped piece of dust), or a loss or misalignment of the formatting information which has been put on the disk when it was originally formatted. As these bad blocks occur, a message like "hd00 gone bad, error trying to move block 52322, no more spares, flushing que", suddenly appears. Generally, the problem can be corrected by sparing out the offending block, and either reentering the data, or reloading software. Older Fortunes had room for 46 bad blocks to be spared out, For:Pro 2.1 can keep track of over 150. This is adequate for normal aging.

Mr. Tressel says that his hard disk has hundreds of bad blocks. This is shown by the Field Engineering Diagnostic disk which can run a hard disk test which tries to read every block on the disk. It is not possible to spare out hundreds of blocks, and even if it were, it would probably not fix his problem. There is something definitely wrong with his system -- the trick is to discover which part is causing the problem, which requires some detective work, some experience, some luck, and hopefully not too much money.

Let's explore some of the options:

Physical damage to the disk

It is possible that the heads might have contacted the drive surface while the computer was being moved. During the shutdown procedure, the heads should be moved to what's called a "landing zone" -- to the edge of the disk where there shouldn't be any data to damage should the heads accidently crash. Usually if there is a minor head crash, the bad blocks will be localized across the damaged area. An analogy could be drawn to a skip in a vinyl record, although since the heads typically are moving around the circle, the bad blocks are all contiguous. If the crash produced some particles by scraping the disk, those particles could move around the disk and cause future head crashes. This would usually be demonstrated by more and more hard errors on the drive. Reformatting the drive would not produce any change.

Alignment problems or "weakening" of format information

The PC press has been filled with information lately about loss of formatting information over time. Electronic and physical misalignment can occur, as well as a reduction in the magnetic charge of the format. Special products have been developed that actually "reformat" the drive on the fly, and maintain the user data that has been placed on the disk. Certain vendors recommend doing this once a month to keep your drive healthy and trouble free. In the Fortune world, we do not have any such products, so the only alternative is to backup the entire hard disk and do a cold boot which will completely reformat the hard disk.

With a tape backup, this is not too painful, although it needs to be done carefully and conservatively. We like to always have two backups of our information before we knowingly erase it all. It is also possible to backup one hard disk onto a second drive, or onto floppies. Usually, unless this option is incredibly difficult, it is the second corrective action pursued because it doesn't require any expense for hardware. The first is the hard disk controller because it doesn't require any time.

If formatting cures the problem and it stays cured, than that is all that was needed.

Bad hard disk controller

The hard disk controller tells the drive what information it wants and where to get it. Controllers can malfunction, causing erratic hard disk operation. A frequent symptom of this problem is lots of hard disk errors that seem to be random. The hard disk diagnostic program would reveal different bad blocks each time it runs.

Bad power supply

Mr. Tressel indicates that his system is very sensitive to power fluctuations. That could be because he has an older Zenith power supply that produces the POWER FAILURE message if the electricity is interrupted for more than 1/250th of a second. This is at least twice as sensitive as the newer Western Electric power supplies. On the other hand, it's possible that the power supply is weak or unreliable, in which case all sorts of other error messages might be produced. The power supply converts the 100-130 volts AC that's coming in from the wall to 5 and 12 volts DC that every component in the computer needs. Bad power supplies can cause erratic operation of any part of the computer. Again, the way to test this is by measuring the voltage that the supply is producing with a voltmeter (never test the supply without some "load" on the circuit), or by swapping the supply out and replacing it with a known good one. If the problem disappears, it was the supply. If the power supply just seems very sensitive and produces POWER FAILURE messages, etc., the problem may not be with the computer at all, but with the AC power supply coming from the wall. Typically, there isn't 120 volts of clean AC power consistently supplied to the computer. In particularly bad environments the power supplied can be extremely inconsistent. Computers should also be plugged into surge protectors to eliminate dangerous spikes. If necessary, an Uninterruptable Power Supply (UPS) should be plugged in between the wall and the computer to insure that the computer is getting a clean source of 120 volts.

Electrical Shorts

The inside of the Fortune case is coated with a conductive surface which reduces noise, but can in fact short out the computer. When a Western Electric power supply is installed, it should be accompanied by a strip of copper conductors which helps ground it to the case, and also some teflon insulating sheets which are attached to the supply and to the top of the case. There are also some "ribs" on the underside of the top of the case that should be scraped so that the conductive coating is removed. Although it's possible that Mr. Tressel's problem is caused by a short, it doesn't seem terribly likely, especially if the above-mentioned steps were taken. If the hard disk problems go away with the cover off, than that would indicate it is the cover. If not, the problem is elsewhere.

Mother board

Although it's unlikely that the mother board is causing the problem, it is conceivable. All other boards on the computer plug into the mother board, and the CPU and other circuitry on it have an effect on other components.

Shielding, Cabling, Capacitors

As Fortune's were placed out in the field, engineering problems were discovered and corrected. Fortune recommended putting metal shielding around drives, particularly the 20-meg drives to prevent interference. These shields are just pieces of sheet metal that encompass the drive. Information gets to and from the drive via 2 flat ribbon cables. These cables may be making faulty connections at either the drive or controller, an may have been damaged by scraping from the sharp points on the memory boards. They should be physically inspected for damage. Fortune suggested that a capacitor be placed on the power cable to the drive to try to filter out any noise in the DC voltage.

Where does that leave you

Repairing anything is an art. It is based on experience, knowledge, skill and luck. As the years pass and more Fortunes have broken and been repaired, the knowledge of what things cause what problems grows. Hopefully the person repairing your computer possesses some of that knowledge and experience. Typically repairs are done by swapping (exchanging a suspect part for a known good one) components until the problem is fixed. In your situation, a repair person might start by swapping the disk controller, followed by the power supply. They might then try a different drive to see if it exhibits the same kinds of problems. If it is the drive, then they might try reformatting and reloading the drive. If none of that works, they could replace the mother board. The trick is to use logic and the process of elimination.

Then there is the issue of who's paying for the repair. If you're on a maintenance contract (and we assume Mr. Tressel is based on his dealer's warning), then the dealer will do the repair in the way that is least costly for him/her. If they've got a pile of hard disk controllers, they'll swap that out first, because the labor is 5 minutes and the controller is readily available. If you don't have any spare parts, reformatting the drive, which is labor intensive might be the way to go.

Many users are buying used systems at a fraction of the cost of their original systems to use as spares. Each user has to evaluate if this is a good solution for their situation and skill. We often recall a book called "Anybody's Bike Book" that explained that a spoke wrench (which is used to straighten a bent or wobbly wheel) only costs 45 cents and any bike store will gladly sell it to at that price -- and then charge 15 dollars to repair the wheel that an unskilled rider/repairer "fixed". There isn't that much inside of a computer to break, but it is important that cables be connected properly, that fans continue to run, that components are smashed by other components, etc. As you may gather, over the years we've made all of these mistakes. Sometimes they aren't serious, and sometimes they require expensive replacement of parts. If you're handy and mechanically adept, you can probably swap parts in as easily as a repairperson. Keep in mind that since most software is coded to a specific Fortune computer, you will not be able to move programs from one computer to a spare without changing the PAL "identity" chip from the old computer to the new one.

Replacing the 20 meg drive

Hard disks have come down greatly in cost. If you choose to replace the 20 meg drive, you'd probably be better off putting in a larger, more reliable drive. The 30 meg (which is actually about 25) CDC drive or 70 meg Micropolis drive could be substituted. It's suggested that a Western Electric power supply be used with either drive, although an upgraded 28-amp Zenith is probably OK. (The Western Electric has a hollow white dot on the power switch.) The 70 meg drive should have WD-04 or 06 level hard disk controller. You may be able to determine the rev level by looking for the 06 marking on the controller, or by looking for "daughter boards", add on printed circuit boards which are attached to the controller near the rear of the card. The hard disk controller runs from the front of the computer to the rear, and is right next to the floppy drive.

Hopefully, that addresses most of your questions. The floppy drive could be dirty, or out of alignment, or there could be a problem with the motherboard/floppy controller.

As to the competence/incompetence and trustworthiness of Fortune dealers, we can't really comment. Presumably, they span the same range on each of those scales as people in other fields. One thing that is for sure is that the ones that are still out there are patient and dedicated. Unfortunately, most users don't have much in the way of choice in terms of a local dealer. If your experience with your dealer is unsatisfactory, shop around and get references. Mail order can be a viable option, although nothing beats seeing a machine in your environment. We strongly encourage everyone to have a modem, especially as their prices fall. They make remote diagnostics and repair much more feasible.

Repairing anything is an art. It is based on experience, knowledge, skill and luck. As the years pass and more Fortunes have broken and been repaired, the knowledge of what things cause what problems grows.

Good luck to you and all of the rest of our readers. Overall, we've found that most Fortune users have been pretty satisfied with the durability of their machines, and with the lack of down time.❑

Josh Lobel

Classified

This Classified section of **/u/fortune** *news* is designed to serve our subscribers and Fortune computer users in general. In order to help everyone use their Fortune hardware and software to its utmost, we are providing a forum for the selling and the buying of used equipment. We will be including this Classified section in each issue of **/u/fortune** *news* as long as there is sufficient interest. So, if you would like to find something for your Fortune or locate a buyer for your equipment, you should not overlook this valuable resource. If you are interested in a listing in this section of **/u/fortune** *news*, please call us at (617) 894-6900.

FOR SALE: 32:16 Package

Updated system with 70 meg Hard disk and tape backup. Includes 4 terminals plus console. Expanded memory and Fortune:Windows. Software includes Fortune:Word, Multplan and Guardian Calender program. Available in 30 days. Price: best offer. Call Judy at 319 234-5701. 10/89

Fortune 32:16 System

30MB hard disk with 256KB, 800K flexible drive, 4 ports, console terminal plus 2 Fortune Intelligent Work Stations (12" monochrome display). Software: Fortune:Word, Multiplan and Sequitur, plus Hayes Smart Modem, 300 baud. Form more information, call Laurie at (602) 795-8870. 11/89

Fortune XP 32:16 with 4 terminals

1 Meg of RAM, 70 MB hard disk, with Fortune:Word, Multiplan, Business Basic, all documentation.

Call Don at (617) 267-6500 for more information. 11/89

WANTED -- 32:16 Software for Job Shop Management

of production of single or lot quantities to customer's specifications, estimating based on past history, scheduling of labor and machines, tracking production and promised "Due Dates", material purchasing, accurate job costing and accounting features.
Please call Dave at 419 294-1987 2/90

Fortune 32:16 XP 70

1 Meg Memory
13-70MB Hard Disks
Ports: 12 serial & 1 Parallel
60 MB Tape Expansion
Fortune Basic Workstation

Will take Best Offer
Call Rena at (301) 907-8900 10/89

Fortune 32:16 – 1 meg RAM, 50 meg

Console + 1 Terminal
1 Meg Memory (4x256)
50 meg Maxtor Drive
C and Pascal compilers
Fortune:Word 2.0
Extra WE Power supplies

Call Chuck at 203 937-5063 2/90

32:16, 6 terminals, 2 printers!!!

Hardware:70 MB with 30 meg/20 meg tape exp cab 2 MB RAM, 2 4-port CommA's. **NEC 7710 & 3500 printers.**

Software: For:Pro 2.0, Fort:Wrd 3.0, Document Conversion, Handshake ITE & COLT, Kermit
Contact: Susan (203) 972-3400 2/90

Fortunes For Sale - Priced Right!

Two - 32:16's (20 & 30 MB HD)
One - SX70T (2MB memory)

For:Pro 2.1, Multiplan, Ext. Fortune:Word BASIC 6.5.12, IDOL, BAS, Dev. Utils., Communications, other software. Call **Russ** or leave message at (708) 510-0088. 1/90

Fully Loaded Fortune XP-45

If you have been looking for the perfect expansion system, this is it.

Start with a beefy 45Mb hard drive, and 1 Mb of RAM, expandable to 2 Mb. Of course, there are 2 four-port Comm A boards, plus **Fortune:Link LAN with the companion board for your other system.** System console is an amber FIS 1000 terminal and keyboard.

For:Pro level is 2.0. Fortune:Windows, Fortune:Word 3.0, Multiplan, ITE, SST's Handshake Colt ,Informix and Development Utilities are all included.

This is one whale of a computer.

Call Mike Eisen at (619) 323-4555. 2/90

Used Fortune Parts For Sale

WD06
CRT Electronics and Tube
256KB Memory Board
800KB Floppy Drive
Comm A 4 Port Board
Video Controller
1 MB XP Mother Board
10 MB Hard Disk
20 MB Hard Disk (C20)
and more...

All components are fully functional, supplies are limited.

For more information, call **Jim** at **(203) 828-4811** 12/89

Fortune 32:16 XP

Hardware: 45 MB Hard Disk, 2 MB RAM, Terminal and keyboard, 4 ports

Software: For:Pro V2.1, Rubix

All Cables and Books included
Price: Best Offer
Call: Brad Hunter for information at (313) 296-7130 1/90

Fortune 32:16

1 MB Memory
30MB Hard DIsk
Fortune:Word
Will take best offer.
Call David at
(718) 965-9857 9/89

20MB 32:16 - Includes For:Pro 1.7.4.1, 768K RAM, console, 2 terminals, 4-port Comm A and a **20MB Tape Expansion Box.** Software includes: SMC BAS modules, Extended Fortune:Word, Developtment Utilities, and a C compiler. Purchased direct from Fortune, Hardly used, will take best offer. Call Gerry Armel at (508) 285-3040. 9/89

ON A WEEKEND YOU COULD:

A. HAVE A BARBECUE WITH FRIENDS

B. VISIT WITH RELATIVES

C. REPLACE YOUR **32:16** WITH A **386**

Select C and Monday morning won't look too much different. You will have your same accounting menus, your same data files, your same word processing documents. But your processing speed will be up to five times faster. Your new platform will be state-of-the-art. You will be able to modernize and expand gradually without excessive down time while you reload files and learn new operations. Whether you choose Fortune or some other 80386-based computer, life could be so much better come Monday morning. Let's talk about it.

What are you doing this weekend?

BASIC
SOFTWARE CORPORATION

5201 POWHATAN STREET
BALTIMORE, MD 21207
301-448-9460

Authorized Dealer for:
SCI/FORTUNE • SCO
INTERACTIVE • THOROUGHBRED®

Now your Fortune can talk to a PC!!!

With our fxfer, floppy transfer program, you can read and write a Fortune floppy disk with an IBM PC or compatible.

With Fortune:Word<-->Word Perfect, Two of the best word processors can exchange documents preserving margins, tabs, indents, bold, underline, etc.

They said it couldn't be done, but we did it, and you can benefit.

Call the folks at /u/fortune news for more information: (617) 894-6900.

/u/fortune news
225 Crescent Street
Waltham, MA 02154

If you're reading a xeroxed copy of **/u/fortune *news***, don't. Copying this publication is prohibited. We depend on subscription revenue to continue publishing **/u/fortune *news***. Please play fair. There aren't many of us out there and we have to stick together.

/u/fortune* news

The Newsletter for Users of SCI/Fortune Computers April 1990 / Volume 7 Number 3

Wedding Bells Ring as Fortune invites guests

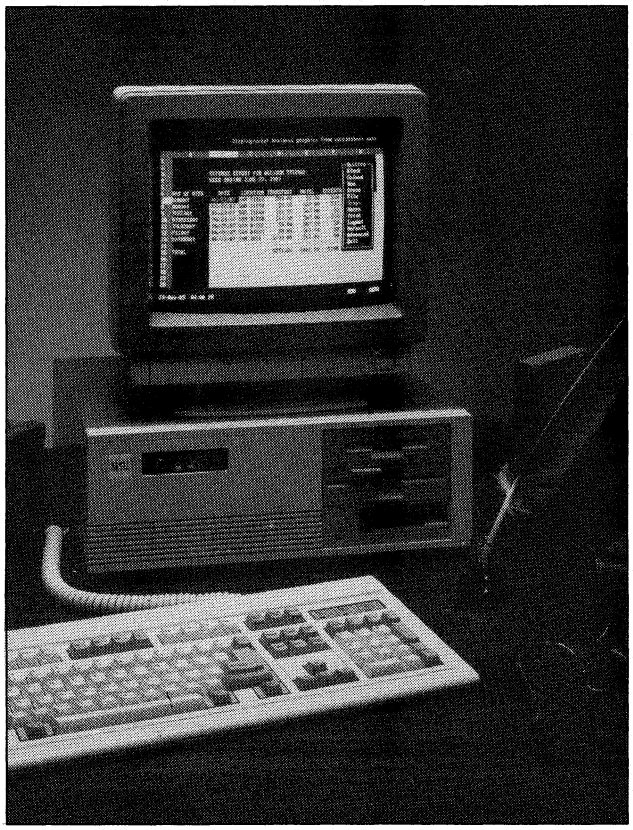
Using Postscript to address the envelopes -- Page 15

Comparing IDOL-IV to other 4 GL's

The BASIC Advisor -- Page 4

More Friendly Menu s on Fortune 5000's -- Page 6

Newest Arrivals To Our Fortune 5000 Product Line



5386SX 20

*386 power at 286 prices

* State of the Art SX-20 -- 20Mhz speed
* Price/Perfomance Leader

5386 33 MHz Cache

* Additional power for more users, or
* Faster Response in existing applications
* Improved Price/Performance

Fortune 5000, a family of products ranging from single user workstations and standalone systems to low end and medium scale multi-user UNIX systems.

Call your dealer for new pricing and more information.

SCI FORTUNE TECHNOLOGY

/u/fortune *news*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz

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, Inc. are affiliated with SCI/Fortune Systems.

Subscriptions cost $60 for one year within the USA. For foreign subscriptions, add $36 each year for added postage and handling costs.

Back issues are offered at:

Volume 1, Numbers 1 - 8	$2/each - $14/set
Volume 2, Numbers 1 - 7	$3/each - $18/set
Volume 3, Numbers 1 - 12	$4/each - $44/set
Volume 4, Numbers 1 - 12	$4/each - $44/set
Volume 5, Numbers 1 - 12	$5/each - $55/set
Volume 6, Numbers 1 -	$5/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $15 each, quantity discounts available. 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 the contents of **/u/fortune *news***.

For more information, call or write:

The Cambridge Consortium, Inc.
225 Crescent Street
Waltham, MA 02154
(617) 894-6900

The text for this newsletter was prepared using an SCI/Fortune 5000 for both text input and typesetting. The typesetting was accomplished by using the Aldus Pagemaker program under VP/ix, which is a DOS emulation.

CONTENTS

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.

March 25, 1990

Question: We have BASIC and Thoroughbred accounting running on a unix system (non-Fortune). Why would one of our printers all of a sudden stop working in BASIC?

Answer: The first thing we always ask is if the printer is plugged in, turned on and on line. Believe it or not, this solves 90 percent of all printer problems. It is also the third most frequent cause of operator embarrassment (the first two being unprintable).

Assuming hardware is not your problem, the second thing we examine is the permissions on all unix files which have anything to do with the operation of printers in BASIC. This includes the tty's. Use the **chown**, **chgrp** and/or **chmod** commands to reset the permissions and flags if you find that any of these files have been altered (e.g., accidently reassigned to a superuser account).

Other possible problems are more involved. For example, if you have one of the newer systems, the printer tty's must be put to sleep before serial printers will work in BASIC. The possibility exists that damage has been done to the scripts which manage these tty's. There are many ways to put printer ports to sleep, and we have no idea what technique was used on your computer. If you do not know either, we recommend that you find technical support.

IN BALTIMORE WE CALL IT "PREAKNESS"

Ah, another mailer from Bill Clarke, Product Manager for Concept Omega. Let's see, new products, yes . . . 4GL VAR Council News, yes . . . product brochures, yes . . . what's this? "Fourth Generation Language Comparative Evaluation Project Report" from Advance Ware in Tustin, CA. Gee, it's nicely bound. I guess it has a bunch of good stuff to say about IDOL-IV. "Prepared by Jeffrey L. Zickler, Advance Ware, Inc., (c) November, 1989." Isn't he the one who wrote the first "How To" book for IDOL-IV? He sure is. According to Page 1, "[Mr. Zickler] is the author of Developing Business Applications in the 90's." Well, that was a pretty good book: a little textbookish, but informative just the same.

According to the introduction, "This report summarizes the results of a comparative evaluation project in which six Fourth Generation Language products (4GLs) were examined and rated based on a number of application software development criteria."

The six products evaluated in this publication were ACCELL from Unify Corporation, dBASE IV from Ashton-Tate, IDOL-IV from Concept Omega Corp., INFORMIX from Informix Software Corp., ORACLE from ORACLE Corporation and PROGRESS from Progress Software Corp. There were ten evaluation criteria under which each product was examined. The first, "General Issues" was subdivided into nine subcategories: required components, "nice to have" components, integration with other application development tools, pricing and upgrades, hardware and operating system availability, hardware and operating system portability, other hardware/software requirements, training and learning issues, and technical support available. The remaining nine major categories were user interface design and development issues, end user facilities, developer interface issues, 4GL structure issues, application development issues, application maintenance issues, model development issues, language semantics and syntax, and overall evaluation.

"This report summarizes the results of a comparative evaluation project in which six Fourth Generation Language products (4GLs) were examined and rated based on a number of application software development criteria."

Each of these categories and subcategories was assigned an importance factor with a "three tiered weighting scheme". The three tiers in order of importance were Most Critical, Very Significant and Expected. The ratings in the various catego-

ries, subcategories and weighting tiers were presented in text as well as in easily readable bar graph charts.

We admit we expected this report to be slanted in the direction of IDOL-IV (after all, Thoroughbred mailed it to us). But Mr. Zickler made every effort to keep an open mind during his study, acknowledging areas in which personal bias may have been an influence. As a matter of fact, IDOL-IV did not place first in the study, which did surprise us.

We cannot say that we totally agree with Mr. Zickler's weighting tier assignments, nevertheless the report was structured in such a way that the reader could define his own priorities and judge each of the products accordingly. It would have been nice, too, if the report had contained more detail. This was a 22-page report, but 100 or more pages would not have been unreasonable considering the number of evaluation criteria used. The lack of feature-to-feature comparisons left us wondering at times whether Mr. Zickler was comparing apples to apples or apples to oranges. Also, one of the issues we feel is important is the ability of a 4GL to interpret, upgrade and run 3GL applications. If this issue was covered in one of Mr. Zickler's categories, we could not find it.

In general, though, we appreciate Mr. Zickler's efforts. We believe he selected the right products to evaluate and was honest in his attempts to be fair with each. We strongly recommend this report to anyone who is shopping for a 4GL product, be they a dealer, a VAR or an end user. Advance Ware is located at 1371-C Warner Avenue, Tustin, CA 92680. The phone number is (714) 259-1761 and the fax number is (714) 259-1751. Or you can probably obtain a copy by contacting your Thoroughbred dealer or Concept Omega Corporation.

Because software technology progresses at such a rapid pace, this report is already a little outdated. We hope Advance Ware and Mr. Zickler continue their evaluations of 4GL products and provide us with occasional updates. We, for one, would be happy to be included on their mailing list for future publications.

Those of you who want to read the publication for yourselves (or who want to wait until the movie is released) should stop reading now. We are going to give away the ending. Keep in mind that what we present here is a synopsis. Mr. Zickler's conclusions are intelligently drawn, and he provides insights into the 4GL product market which must be read in full to be appreciated. Based upon the evaluation criteria and priorities defined in the report, the six products finished in the following order.

The crown went to PROGRESS, "a mature, solid 4GL with a broad range of capabilities and operating environments". Mr. Zickler rated this product very high in most of the categories. He warns Progress, however, that it must continue moving forward if it wishes to keep the lead. He goes on to say this may be difficult in some categories, maybe impossible in others: "The developers . . made some poor, short-sighted decisions in their ORACLE integration approach." There is a good chance Progress will be overtaken by it's closest competitor (you guessed it).

First runner-up and heir to the crown was Miss New Jersey, IDOL-IV, "the 4GL easiest to improve dramatically" and "the best overall 4GL in this group". The report pointed out three major problems with IDOL-IV. The first, a lack in the Training/ Learning subcategory, is something Concept Omega will be hard pressed to address in the immediate future. They simply do not have the resources to provide the expensive tutorials, videos and "how to" manuals which are available from other 4GL suppliers such as Progress Software. But since the publishing of this report Concept Omega has made some changes which address the two other major problems. In response to the second problem, a lack of technical support, they have created a technical support division to help developers of applications in 4GL. The third major problem, lack of end user facility, has been addressed in part with the release of Query-IV. We will discuss this product more in depth in a future article.

The rest of the 4GL's ranked significantly lower than PROGRESS and IDOL-IV. Placing third, just a hair above fourth, was ACCELL which when "once mastered, can be made to do some really remarkable things". Mr. Zickler says the major problems with this product are not easily solved by the supplier as was the case with IDOL-IV. The report finds ACCELL difficult to understand, restrictive (operating only in the Unix/Xenix environments) and expensive. Furthermore, it "uses a lot of hardware resources".

Fourth place went to INFORMIX, boasting "a broad list of operating environments . . and . . the ability to use any screen for QBE, which neither Progress nor IDOL-IV can offer". Problems included language obscurity, procedural language limitations and poor user interface. This, too, was found to be an overly expensive product.

Ranked just below INFORMIX, in fifth place, was ORACLE, a 4GL which "still does not contain all the components required by the accepted definition of a Fourth Generation Language". In all fairness, the report admits that Oracle Corporation "never positioned its product as a 4GL".

Although it finished far in last place, "nevertheless dBASE IV is a real improvement over previous Ashton-Tate products (provided all the bugs get cleaned up), and it did evaluate fairly well against some stiff competition". Other than these kind words, the report found little good to say about dBASE IV. Problems included "fractured and overlapping" language syntax and semantics, "a bewildering number of commands" and difficulty in making major changes and enhancements. At the time of the report, dBASE IV operated only in the DOS environment.❑

System Administration: Part 35

No Longer requires a Degree in UNIX Wizardry

This series is a must for all owners and users of the SCI/Fortune computer. Dave is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the SCI/Fortune computer, and is past president of the Houston UNIX User's Group. He contributes independently to **/u/fortune *news***.

Creating Friendly Menus on a 5000 *by Dave Kloes*

In the last issue, we left you with a shell program called "kmenu" that can be used a front end user-friendly menu system. In this issue, we will discuss the script in detail. Our intent is not to present everything there is to know about shell programming, but to present some of the basics as they relate to this particular script. Please refer to the copy of the script that was presented in the last issue. Learning about shell programming will help you administer your system regardless of which SCI/Fortune system you are using.

NOTE: Please make the following two corrections to the "kmenu" shell program shown in the last issue:

1. Remove the double semicolon (;;) from the end of line 67.
2. Put the double semicolon at the end of line 72.

```
1     ext=n
```

The first line of the program sets the variable "ext" to the value "n". This variable is used to keep the program in a "loop" that basically continues to present the menu after each selection. Variables are used just like in math where we did things like "x=0". Here the name of the variable is "x" and its value is "0". Note there are no spaces on either side of the equal sign.

```
2     while test "$ext" = "n"
3     do ........
         ........
79    done
```

In line 2, we have a "while loop". A loop is a continuous process that is done as long as a certain condition exists. Each "while" loop must have an associated "do" and "done" statement. The instruction we are giving here says that **while** this condition exists, **do** the things listed until you see the **done** statement. The "do" statement in our script is on line 3 and the associated "done" is on line 79. Here we have enclosed our whole menu system in the loop. The commands for the actual menu and its selections begin with the "clear" in line 3 and end with the "esac" on line 79.

Basically, line 2 says that while the value of the variable "ext" is equal to "n" to continue to display (or redisplay) the menu. Let's break this line down and look at each of its components.

The "test" command in Unix is used to perform various tests. For example, we can use it to see if two strings are equal (=) or not equal (!=). We can evaluate numeric expressions to see if one result is equal (-eq); not equal (-ne); greater than (-gt); less than (-lt); greater than or equal to (-ge); or less than or equal to (-le) another result. We can also test to see to see for things such as whether a name entered is a valid file name or directory name. See the "test" command in your Unix documentation for more information about this command.

Whenever we want to refer to the value of a variable in a shell script, we put a "$" in front of it. Our variable name is "ext" and when we want to refer to the value of "ext", we would use "$ext". Let's look at line 2 again:

while test "$ext" = "n"

In actuality, "test" returns a successful or not successful result (0 or 1). Here the test is performed to see if the value of the variable "ext" is equal to "n". The "while" will continue as long as this test is successful (as long as "ext" is equal to "n"). Note the required spaces on either side of the equal sign when we evaluate expressions. We also enclose both sides of the expression in quotes when evaluating strings. While this is not required, we find that in some cases, it is a good rule

to follow. When we do numeric evaluations, we DO NOT enclose either side in quotes:

while test $var -eq 3

Note also that "=" and "!=" are reserved for string evaluations and -eq, -lt, -le, -ne, -gt, -ge are reserved for numeric evaluations.

```
3    do clear
4    echo "                                Uni-Komp Menu
5
6    Your current working library is: `pwd`
7
8    Depress <del> to exit at any prompt.
9
10     1.  System Administration Menu.
11     2.  Daily Backup (/usr directory only - tar).
12     3.  Weekly (Friday) Backup (full system backup - dump).
13     4.  Shutdown system.
14     5.  Link New User Into Menu System.
15     6.  Callout on the modem.
16     7.  Games Menu.
17     8.  Format Floppy Disk.
18     9.  Backup /usr Directory to Floppy Disk.
19    10.  'tar' Backup of Individual Files/Directories.
20    11.  Exit Menu.
21
22   ENTER YOUR SELECTION: __^H^H\c"
```

Remember that the "do" in line 3 starts the list of commands that are to be run as part of the "while" loop. The "clear" command clears the terminal screen and is the first action taken each time the menu is displayed.

Next, the menu is displayed on the terminal screen with the "echo" command. Note that what is echoed to the screen is everything between the quote on line 4 and the ending quote on line 22. You need to include spaces where necessary to center what is displayed such as we did on line 4.

Anytime we use the backquote (`), the shell expects that we are going to use a Unix command. In line 6, for example, the `pwd` means to display "the results of the pwd command". In this case, since the "pwd" command tells us our current working directory, this will be displayed on the screen where it appears within the "echo" command. If we wanted to display the date, we would put `date`. We could also set a variable equal to the result of a Unix command. For example, our menu would function the same if we did the following:

```
directory=`pwd`
echo "Your current working library is: $directory"
```

In line 8, we are instructing the user that they can exit at any prompt by depressing the CANCEL/DEL key. Remember that we are able to do this because of the "trap" we set in the "kstart" program we showed you in the last issue.

The actual menu selections are displayed in lines 10 thru 20. The action that is taken for each selection depends on the logic we use in our script.

Finally, in line 22, we prompt the user to enter a selection from the menu. Here we use a technique that controls where the

cursor will be on the screen. If we do an "echo" by itself, it does an automatic line feed and the cursor is positioned on the next line. When we are prompting a user for input, however, we normally want the cursor to remain on the same line as the prompt. In these cases, one of two techniques can be used depending on what version of Unix you are using. If you look at the "echo" command in your reference manual, it will show which of the two methods you can use.

One method is to put a `\c` at the end of the echo line and before the ending quote as we have done in our example. Another method is to use the "`-n`" flag argument to the "echo" command. Some Unix versions such as For:Pro and SCO Xenix allow you to use either method.

Notice that there are two control H's at the end of line 22. This represents a backspace on many terminals. In effect, `\c` at the end of line 22 keeps the cursor on the prompt line and the two control H's back up the cursor two spaces so that the cursor is over the first underscore on the screen. Other control characters such as reverse video and blinking can be used. Remember, however, that not all terminals have the same control sequences and our screen displays would be effected if the program was run on another type of terminal.

```
23    read sel
```

Anytime we want the program to stop and wait for user input, we use the "read" command. The value of what is entered is put into the variable that follows. In this case, we have used "sel" as our variable name. In this particular shell program, we would expect a number for the selection the user picks from the menu to be the value of this variable.

```
24    case $sel in
25            1) clear
26               /usr/bin/sysadmsh;;
27            2) ......
                 ......;;
78    esac
```

The "case" command is a glorified "if" statement. Basically, it tests the value of a variable and takes an action depending on the value. In our menu example, we said that whatever the user enters at the "ENTER SELECTION" prompt is put into the variable "sel". The "case" sequence evaluates the value of the variable and takes the action based on the number of the selection that was entered. In essence, the "case" statement says that if the value of the variable "sel" is "1" then do all of the things listed after "1)" on line 25 until it sees a double semicolon (;;). The double semicolon tells the shell that all of the items associated with that particular selection or "case" are finished. In our script, all of the commands from line 25 to line 26 would be executed if the user selects "1"; all of the commands from line 27 to 40 would be done if the user selects "2", etc.

In our shell programming, however, we must anticipate all of the things a user may enter - including invalid selections. This is handled in the "case" sequence as we did in line 75. The only way we can get to the "*)" case is if none of the cases above it apply. Once a case is met and the actions taken, none of the other cases are evaluated. In other words, if the user enters "2", the actions associated with case 2) are done and control of the program goes to the "esac" and none of the other cases are evaluated. The *) case, therefore is normally used to display an error message because the only way the program could get this far is if none of the other cases matched. If the user enters a valid number from 1 to 11, an action is taken otherwise the error message on line 75 is displayed.

Let's look at the logic of how each of the lines in the program works from the beginning of the program:

1. First of all, since the name of our shell progam is "kmenu", it is started by entering "kmenu" at the Unix prompt.

2. In line one, the value of the variable "ext" is set to "n".

3. All of the actions between the "do" on line 3 and the "done" on line 79 are taken as long as the value of the variable "ext" is equal to "n". We know all of these actions are taken when the program is first run because we set the value of "ext" to "n" when the script is first run. Whether the program continues to to run depends on what the user enters.

4. The screen is cleared with the "clear" command in line 3 and everything between the starting quote on line 4 and the ending quote on line 22 are displayed to the terminal screen using the "echo" command.

5. The program then stops and waits for user input with the "read" command on line 23.

6. The user enters "1" and the "case" statement causes the program to take the actions on lines 25 and 26. The two actions are to clear the screen and then run the "/usr/bin/sysadmsh" command which is the SCO Xenix system administration menu.

7. Once the user exits the SCO Xenix menu, control comes back to our program. No other cases are evaluated and the program goes back to line 2. If the value of "ext" is still set to "n", then our menu is displayed again. The only way the user can exit our program is to select item 11 on the menu which executes the "exit" command on line 74. The program would also exit at this point if we set the value of "ext" to anything other than "n":

```
74        11) ext=y;;
```

If we had chosen this method, the program would go back to line 2 and exit since the value of "ext" is no longer "n".

In the next issue, we will continue by discussing items 2 and 3 on the menu. We will also include some general backup guidelines for Unix systems and will discuss some of the commands used for backup. Before we depart, however, here are some of the more common errors that beginning shell programmers make:

1. Using a starting quote without an ending quote.

2. Leaving out required keywords in command sequences:

   while, do, done
   case <variable name> in, esac

3. Putting spaces on either side of the equal sign when assigning values to variables.

4. Not putting spaces on either side of operators when using the "test" command.

5. Forgetting to put the double semicolon after each case when using the "case" command.

6. Forgetting the word "test" when you are evaluating expressions.

When these kinds of errors are encountered, you will get a "syntax error" and your program will not run. Other error messages such as "end of file unexpected" may be generated. The key is not to panic and to look at your script for some of the more common errors listed above.❑

System V vs. For:Pro

What are the differences between UNIX on a Fortune 5000 and a 32:16?

In last month's issue of /u/fortune news, we began a series which will explore the differences between For:Pro and System V Unix as implemented by Interactive. In that article, we finished by noting the large differences between the cp commands under these operating systems.

In this article, I would like to begin by talking about the copy command that exists under Interactive because it performs some of the useful portions of the cp command as we know it under For:Pro.

Under For:Pro the cp command will "recursively" copy files if you use the -r flag. For example, if you wanted to copy all the files in some directory, and all the subdirectories of the directory, to some other location, you would type:

```
cp -r * /newdir
```

The cp command under Interactive does not support a -r flag but the copy command does. One difference, however, is that if you want to copy all the files in a directory including all the subdirectories, then you would use the name of the directory instead of an asterisk. For example, suppose I wanted to copy all of the files in /usr/mbp to /tmp/fred, then you would type:

```
copy -r /usr/mbp /tmp/fred
```

There are several other functions of the copy command under Interactive that are similar to the cp command under For:Pro. The -a flag will cause copy to "ask" the user before attempting a copy and this flag is equivalent to the -a flag on cp under For:Pro. The -l flag will cause copy to link files whenever possible and this flag is also equivalent to the -l flag on the cp command under For:Pro. The -n flag is used to make sure that the file being copied doesn't already exist in the destination directory. If it does exist, then the file is not copied. This flag is also available on the cp command under For:Pro. The -o flag will cause copy to set the owner and group of the newly created file the same as the original file. If not set, the owner and group of the person who invoked the copy command is used. This flag is also identical to the -o flag for the cp command under For:Pro. The -m flag will cause copy to keep the same modification and access time on the new file as on the original file. This flag is similar to the -t flag on cp under For:Pro. Finally, the -v flag, which stands for verbose, causes copy to print on the screen a description of every action it is taking. This option is similar to the -V option on the cp command under For:Pro.

In general, then, the combination of cp and copy under Interactive perform many of the same functions of the single cp command under For:Pro. A couple of the functions that are not performed under Interactive but are under For:Pro are:

1) Backup and Restore (-B and -R)
2) Read from a list of files (-S)
3) Use full path names of files (-p)

Text Processing Filters

There are some relatively common commands on the Fortune 32:16 series of computers running For:Pro that either work differently or do not exist on the Fortune 5386 computer running Interactive Unix. The first commands that I will discuss deal with processing text oriented files. In general, these commands are used to make changes to one file and then write out the new file.

Tr - Translating or Deleting Characters

The tr command is used most often to either make one-to-one changes or to delete characters from a file. For example, suppose you want to change all occurrences of i to I - that is lower case to upper case. The command tr would be the perfect command for this purpose. Under For:Pro and under Interactive Unix you would type:

```
tr i I < infile > outfile
```

Now suppose you wanted to do something a little more complicated which is to say, change all lower case letters to upper case letters. Under For:Pro you would type:

```
tr a-z A-Z < infile > outfile
```

However, under Interactive Unix, you would need to type:

```
tr "[a-z]" "[A-Z]" < infile > outfile
```

Here the obvious difference is that brackets and double quotes are needed before the tr command will recognize a range of characters. In general, you will need use quotes more often under Interactive Unix so that the Unix shell will not interpret the characters first. In other words, if the tr command blows up on you, try using quotes.

The standard flags for tr work the same under For:Pro as they do under Interactive Unix. With tr, you can delete characters by using the -d flag. For example, if you wanted to delete all

the occurrences of the letter a from a file, you would type:

```
tr -d a < infile > outfile
```

The -c flag is used to "complement" the desired action. So, for example, if you wanted to delete everything but the a from a file, you would type:

```
tr -cd a < infile > outfile
```

As an aside, the above example may seem stupid but supposed you wanted to know how many times a particular character occurred in a file. One way to solve this would be to do a tr command like the one above and then do an "ls -l" on outfile to see how many characters it had in it and that would be your answer! Another way to see the number of characters would be to use the wc command as follows:

```
tr -cd a < infile | wc -c
```

The -c tells wc to just print out the number of characters it has counted rather than the other standard information which includes a "word" count and a line count.

The final tr flag is -s which has the effect of "squeezing" multiple occurrences of a character into just one of them. Suppose you had a file which had many spaces on the end of a line and you wanted to reduce the number of spaces to just one. You could type:

```
tr -s ' ' < infile > outfile
```

Changing Tabs into Spaces and Back Again

There are two commands that are available as part of the Development Utilities that are used to change the way tabs and spaces are represented in a file. These commands are called expand and unexpand. Expand is used to "expand" the tab character into the proper number of spaces so that text will line up. Unexpand does the opposite - which is to say, unexpand changes spaces into tabs whenever this will result in fewer characters. I have often used expand and unexpand under For:Pro for a variety of reasons. One reason is that sometimes I will make a file and I will use tabs but later I will find that an application that I apply to that file doesn't like to see spaces. Therefore, I will used expand to change tabs to spaces.

Expand and Unexpand work very simply under For:Pro. If I have a file called infile and I want to change all the tabs to spaces, I would type:

```
expand infile > outfile
```

And unexpand would work the same way.

Interactive Unix does not have the commands expand and unexpand but it does have a command that does exactly what expand and unexpand does under For:Pro. The name of this command is called newform. To expand the tabs in the file called infile to spaces under Interactive Unix, type:

```
newform -i infile > outfile
```

and to unexpand tabs, simply type

```
newform -o infile > outfile
```

Of course, newform is designed to do a number of other things besides expand or unexpand tabs. For example, the -s flag allows you to "shear" off leading character on each line up to the first tab and then place up to 8 of the sheared characters at the end of the line. One interesting use of this flag would be to take line numbers, at the beginning of a line, and put them on the end of the line. Another flag, -b, allows you to simple delete characters from the beginning of a line and the -e flag is equivalent but lets you delete characters from the end of the line.

Since I am used to using expand or unexpand to manipulate tabs, I set up a little shellscript on the 5386 which will take care of both the expand and unexpand functions. The following is this shellscript:

```
:# expand and unexpand equivalents
# using newform command
#
case $0 in
      expand) arg="-i";;
      unexpand) arg="-o";;
esac

newform $arg $@
```

To set this command up you would put the above 10 lines into a file. At first, you should call this file expand and you should create it in one of the directories that are a part of your path, for example, /bin or better yet, /usr/public. Once it is created, you should make it executable by typing:

```
chmod +x expand
```

After this, you should link this file, called expand, to unexpand in the following fashion:

```
link expand unexpand
```

The expand and unexpand functions are all taken care of in the same shellscript. The way the shellscript knows that you want to expand or unexpand is by the name that you use when you call the command! This information is included in the `$0` argument which simply holds the name of the command. If we use expand to call the script then we set the argument for the newform command to -i. On the other hand, if we call the script with unexpand, then the argument gets set to -o.

Counting Lines, Words and Characters

We mentioned the wc command in one of our examples above and wc is truly a useful tool in the Unix user's arsenal. I personally use wc in a wide range of situations. However, there are a couple of differences between the way wc is implemented under For:Pro and how it is implemented under Interactive Unix.

Its main uses are the same - count words, lines and characters in a file. Under For:Pro, however, the wc command is able to do quite a bit more including count the number of pages that a file would take if printed (the -p option) and specify the amount of time a file would take if transferred at a specified baud rate. In addition, the -v flag under For:Pro will print a verbose output that tells you all of these things at once. Thus, the wc command under Interactive is simply a subset of the wc command under For:Pro. If you are interested in more details of how wc works, please see the October 1989 issue of /u/fortune news.

Conclusion

In this article, we discussed several differences between Interactive Unix and For:Pro. First, we continued from last month's article and a discussion of the cp command. Next, we discussed some text processing filters including tr and newform. Finally, we discussed the differences between For:Pro and Interactive Unix relevant to the wc command.❑

Mark Palmerino

News From SCI/Fortune

(Editor's note: In addition to many new product announcements, SCI/Fortune has made many changes in their price list which should be pleasing to end users. Contact your dealer for the latest news.)

Memory Expansion Board Available for Fortune 5286

SCI/Fortune announces the availability of an Extended Memory Expansion Board for the Fortune 5286. The board is populated with 1 MB of RAM, and has an additional capacity for a total of 2.5 MB. The board is addressable on 64 KB boundaries, making it possible to add multiple Extended Memory Expansion Boards to the Fortune 5286 without wasting necessary memory space. The maximum number of boards that can be installed in a Fortune 5286 is six, bringing the maximum amount of RAM up to 16 MB (including the motherboard's 1 MB).

Accessing memory on the Extended Memory Expansion Board is as fast as accessing main system memory. There is no penalty as a result of installing the memory board in an I/O slot. Addition of this board does not make the system run at a reduced clock rate when accessing the extended memory on the board. Installation and configuration are quick and easy.

Additional memory chips can be added to the Extended Memory Expansion Board in increments of 512 KB. Installation of the additional memory can be done very easily, following instructions supplied with the board.

The Extended Memory Expansion Board provides the memory necessary to run multi-user SCO XENIX or more memory intensive DOS applications.

Hard Disk Drive Upgrades Now Available for Fortune 5000 Systems

SCI/Fortune is pleased to announce that hard disk drives and controllers can now be ordered separately for upgrades to both the Fortune 5386 and Fortune 5286. This provides the means to increase hard disk storage capacity beyond the standard configuration by replacing the existing hard disk with a higher capacity drive, or by adding a second hard disk, either at the time the system is originally ordered or later, when the need arises at the customer site.

Product Description

The following hard disk drives are available 71 MB, 161 MB, 338 MB, and 676 MB (formatted). The 71 MB drive is the ST-506 type; the 161 MB and 338 MB drives are supported by a 10 MBPS Western Digital ESDI controller; and the 676 MB drive is operated through a 15 MBPS ESDI controller. Installation instructions are included.

Depending on the existing configuration and the type of hard disk being replaced or added, a controller may also be required. Two Western Digital controllers are available: the WD1006V-MM1 ST-506 controller and the WD1007V-SE1 ESDI controller. Each controller includes vendor authorized documentation as well as **SCI/Fortune** Installation Instructions.

SCI/Fortune is also releasing "spares" of the Utilities Diskettes and Chassis Rails. A utilities diskette is required in order to complete the hard disk formatting after physical installation. If the original Utilities Diskette, supplied with the system, has bee misplaced or is otherwise unavailable, a new one can be ordered.

Spare drive mounting rails are also offered for those users who do not have a set of rails available to support a second disk. The spare drive mounting rails include attaching hardware (four screws).

Additional Technical Information

The hard disks supplied as original equipment in **SCI/Fortune** systems are supported by one of the following controller types: ST-506, ESDI 10 MBPS, or ESDI 15 MBPS. Therefore, second disk choices are limited, and may require that a controller be purchased.

The following general rules apply:

- First, determine if the system chassis can accommodate the installation of the drive (half-height or full-

height) under consideration.

- If a second drive is added to a system, you must purchase a second 20-conductor data cable and a set of drive mounting rails.

- If an ESDI-controlled hard disk is added to an ST-506 system, the ST-506 controller must be disabled. The original ST-506 disk will not function with the new ESDI controller.

- If a 676 MB (formatted) hard disk is added to a 150 or 322 MB ESDI system, the controller must be changed to a 15 MBPS controller, which will run both 10 MBPS and 15 MBPS drives.

- An ST-506 drive cannot be added to an ESDI-controlled system without losing the functionality of the ESDI drive.

Choosing the best possible upgrade requires that you know which of the following controllers is currently in the system. The possibilities are listed below with other technical information to help you determine the best solution for your customer:

- In Fortune 5386 model 300/301 system boards, the on-board controller is an MFM encoded ST-506 supporting 17 sectors per track at a 2:1 interleave factor. If this controller is not used, it must be disabled.

- The Seagate ST11M 8-bit controller supports two ST-506 MFM drives. Although the ST11M does not support 1:1 interleave, it can support drives with up to 1024 cylinders and 16 heads.

- The Western Digital WD1006V-MM1 16-bit controller supports two ST-506 MFM drives with up to 2048 cylinders and 16 heads. 1:1 interleave is supported.

- The Western Digital WD1007A-WAH 16-bit controller supports two 10 MBPS ESDI drives at a 1:1 interleave factor.

- The Western Digital WD1007V-SE1 16-bit controller supports two 15 MBPS ESDI drives or two 10 MBPS ESDI drives at a 1:1 Interleave factor.

- The Adaptec ACB2322B-8 16-bit controller supports two 15 MBPS ESDI drives at a 1:1 interleave factor. It also includes a flexible diskette drive controller which is usually disabled because all Fortune 5000 system boards include a flexible diskette drive controller on board.

Other considerations:

- The interleave factor is determined by the hard disk controller.

- 16-bit controllers are more efficient that 8-bit controllers.

- Data (20-conductor) and control (34-conductor) ribbon cables work identically, whether used for ST-506 or ESDI style drives.

- In any two ST-506 or ESDI disk system, both drives' drive select jumpers are set to 1 if the manufacturer uses a 0,1,2,3 system; 2 if the manufacturer uses a 1,2,3,4 system. The drive connected to the end of the control cable is terminated (the terminating resistor on the inside drive should be removed).

Questions/Answers from Technical Support

How can I disable the on-board video and hard disk controllers on my Fortune 5386?

To disable the on-board video controller:

Locate the three EGA video chips between the power supply and the expansion slots on the model 300 or 301 motherboard. These chips are labeled:

Paradise PBI 383060 8817
EGA PROM U59
L1A22861 Paradise PEGA2A.

Removal of these chips will disable the video controller.

To disable the on-board hard disk controller:

Locate jumpers JP8 and JP27. Move both jumpers so that they connect pin 1 to pin 2 (instead of pin 2 to 3).

NOTE: **SCI/Fortune** recommends that these procedures should only be performed by a qualified technician. These modifications would be necessary to add external video/hard disk controllers.

Technical Bulletin

Description: Fortune 5386 systems with Revision Level D (1.10.12) BIOS PROMs will not boot the NOVELL Operating Systems 2.12 and 2.15.

Symptom: When trying to boot, the system will hang with the following error message:

ABBEND:
General Protection Interrupt
Running Process TIProc

There is not way to recover from this problem except to cold boot the system.

Resolution: Upgrade to level "J" (1.10.20) PROMs.

Before attempting to install the Novell Network, verify what level CMOS is installed in the Fortune 5386. The PROMs are labeled on the motherboard. You can also determine the CMOS level by the ROM BIOS version level displayed when the system is booted. ROM BIOS Version 1.10.12 indicates "D" level PROMs. When upgrading to level "J", a PAL must also be upgraded. This PAL is:

Model 300/301: U81-Part number TT5210178-001 from Rev A to Rev D

Model 302/303: U22-Part number TT5210178-001 from Rev A to Rev D

Model 304: U22-Part number TT5210178-002 from Rev A to Rev D❑

/u/help

Fortune:Word Tips and Questions

Question: How do you get the Fortune:Word Exception Dicionary to save U.S. or P.M.?

Answer: I guess your question is generally how do you save words with punctuation. We experimented with your problem and found that by simply using the COPY (F11) key during a spelling check, the words you mention were not saved. When we looked in the exception dictionary (by simply editing it like any other document) we noticed that U.S. got saved as u.s with no period at the end. Perhaps this is because the dictionary assumes that a word with a period at the end is at the end of the sentence and therefore discards the final period. (What is the correct punctuation for the sentence: **It happened at 10 P.M.** ? Should there be two periods at the end?)

In any case, by simply editing the exception dictionaryand adding the second period, it seemed that the spelling check then ignored the U.S. which was in our document, so presumably this fix worked. The exception dictionary can be edited in the same way that any other document is editied.

Question: I operate a Fortune 32:16 XP system with Fortune:Word 3.1 software. One of the printers available to our system is a Hewlett Packard HP 2686A LaserJet printer. I want to replace the caret symbol (shift 6 on my keyboard) with a degree symbol in my output documents.

I know that my printer is capable of making the degree sysmbol as it has printed before as part of "garbage" that has printed because of some error made in printer assignment.

Is there an easy way to change this? Would I change it on my system through Unix or on my printer through Unix. Please let me know where to start to replace a character with another character or symbol. Any help would be appreciated.

Julia Prince, Nashville, TN

Answer: The answer to this one involves altering the /usr/lib/wheels/hp.whl file. As we've mentioned in the past, the wheel files act as filters for various printers. They specify the characters to print and/or their widths. In theory, it should be possible to alter the hp.whl file (the wheel file for an HP LASERJET printer) to print out a degree symbol instead of a ^, however we had some difficulty doing it. Fortunately, there is another solution.

The wheels file looks like this:

```
# Standard ascii character set
# wp print wheel 1 portrait letter size
>1
+init "\E(8U\E(s0p10h12v0s0b5T"
-          "\263"       #caret goes to degree
^Yj        "\273"          #extended English pound sign
^YAA       "\241"          #Grave A
^YAE       "\243"          #Grave E
...
A          (I*36/300)
B          (I*30/300)
C          (I*32/300)
D          (I*36/300)
```

The first column is always the character that is typed in and shows on the screen. The column in double quotes, which is optional, indicates what character should be printed when the screen character is entered. The column in parentheses, which is also optional, indicates the width of the character (for proportionally spaced fonts). For instance, the "-" near the top of the file will print character "\263". "^Yj" prints character "273", which is the English pound sign. We'll explain more about this in a moment. Please note that >1 indicates the number of the wheel file that you will specify on the **Fortune:Word Print Menu**. If you wanted to send your files in a secret code, you could have the printer put out a "b" every time there was an "a" in your file. To do that the wheels file would have this line in it:

```
Screen      Printer
a           "b"
```

With this information, there are just two other steps. The first is to find the special code which will produce a degree sign on the HP Laserjet. If you can find an ASCII chart in your Laserjet manual, you may find the Hex, Decimal, and perhaps Octal code for the degree sign (along with other symbols). We found that the decimal code for the degree sign is 179. In order to get this to work with the wheels file, it's necessary to convert that number to base 8, or octal. (As Tom Leher says, "That's just like base 10 if you're missing two fingers.")

The answer is 263. If you want to know how we got that, read on for a moment. In base 8, instead of the columns being 100's, 10's and 1's (so 179 is **1**x100, **7**x10 and **9**x1), the columns are 64's 8's and 1's. To achieve decimal 179, we'll need **2** 64's (128) plus **6** 8's (48) + **3** 1's (3). 128 + 48 + 3 = 179.

As we said above, to have the ^ print the degree sign (\263), we should be able to insert this line into the /usr/lib/wheels/hp.whl file:

```
^       "\263"
```

This didn't work for us for some reason. However, there is a keyboard sequence that will acutually show a degree sign on the Fortune screen. It is `CTRL-Y 0`. Hold down the grey CTRL key and then press the letter Y. Release both keys and hit the number 0 to the left of the backspace. A degree sign should appear. To make this print a degree sign, insert this line into the /usr/lib/wheels/hp.whl file:

```
^Y0    "\263"
```

Note that the up-caret Y is actually a CTRL-Y. To enter it using the **sc** or **screen** editor, hit a CTRL-C first, then the CTRL-Y. If you are using **ed** or **vi**, enter a CTRL-V first. The spaces between the two columns are created with the TAB key.

To actually make the change, follow these steps:

```
Login as root

Type cd /usr/lib/wheels
Type cp -t hp.whl hp.whl.rl8
Type chmod 666 hp.whl
```

These commands will copy your hp.whl file just in case you make a mistake while editing. The hp.whl file is copied to hp.whl.rl8, which is just an arbitrary name that presumably doesn't already exist on your computer.

If you have the **screen** editor, type

```
screen hp.whl
```

If you have **sc** editor, type

```
sc hp.whl
```

Hit the letter p to page through the file, until you see:

```
# wp print wheel 1 portrait letter size
>1
+init "\E(8U\E(s0p10h12v0s0b5T"
```

Put the cursor on the line following the init line, and press the letter i for insert.
Type

`CTRL-C CTRL-Y 0` (with no spaces) `TAB "\263" RETURN`
`Hit the ESC key`
Type q for quit and u for update.

After that change is made, the `CTRL-Y0` sequence should print as a degree symbol.❑

What's been happening in Boston? *or:*

Postscript brings guests to wedding

What's been happening around Boston?

The big news around Boston lately is that yours truly, Josh Lobel, is about to get married. Probably the big day will have come and gone by the time you read this, and things will be back to normal, or back to the future, or whatever. In any case, as you may have learned elsewhere, this is at least part of the reason that this newsletter has been delayed for such a long time. It seems that all of the details of performing a marriage take some time.

One of the things that absorbed some of my time was our invitations. We knew that we wanted something different than the standard wedding invitation we could order at the printer down the street. I had this idea that it would be fun to try to address the envelopes with the computer, because try as I might, my caligraphy is terrible, and my fiance's isn't much better. Since 1971 when I bought my first Rapidograph caligraphy pen, I've actually spent much more time mastering the computer, so it didn't seem to tacky to do what I do best.

This decision posed some problems though. The first was how could I feed an envelope through my NEC LC890 laser printer. The second was how best to do a "mail merge" to get the addresses onto the letters. I decided that the best approach to the envelope problem was to not use a traditional envelope. Instead, we used a piece of "cover stock", heavy paper that could be folded up to make an envelope. That way we could print it while it was flat, and then fold it as needed.

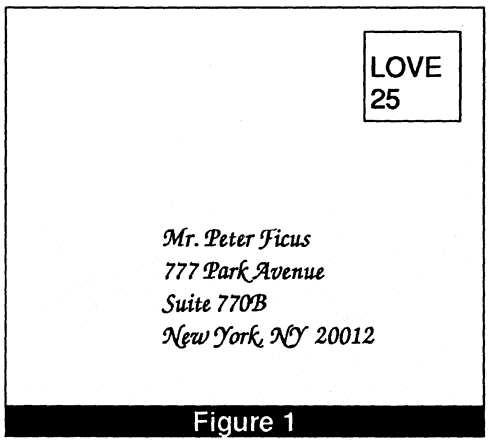


Figure 1

We ended up printing on a standard 8 1/2x11 piece of paper, then cutting that to a square size, and folding in the four corners as shown in Figure 1.

This solved the mechanical problem of printing an envelope, but introduced a new problem. If I wanted the address to end up parallel to the top of the envelope, I would actually have to print it on a 45 degree angle. Figure 2 shows how the address is printed.

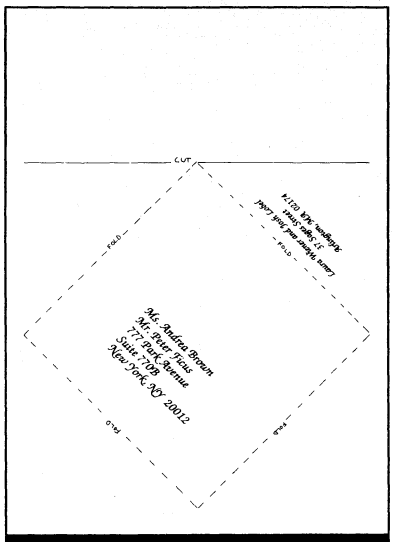


Figure 2

I didn't know of any software programs that could do a mail merge and rotate the type, so I decided to experiment with the Postscript information that Mark wrote about in the last issue. Postscript is a page description language that is used mostly for desktop publishing. Usually, our programs (like Pagemaker) take care of generating the postscript "programs", but in this case I decided it might be easiest to create mine from scratch. The NEC LC890 is a laser printer that supports Postscript as well as HP laserjet emulation.

I started expermenting with just choosing a font, rotating it, and displaying it. Here's the program I started with:

```
/ZapfChancery-MediumItalic findfont 18 scalefont setfont
45 rotate
370 320 moveto
(This is a test) show
showpage
```

The first line chooses the font -- ZapfChancery bears a striking resemblence to hand caligraphy, and the 18 scalefont sets the font size in points. 45 rotate rotates the axis 45 degrees. 370 320 moveto moves to those coordinates on the page (more on this shortly). Any text that you want to display is enclosed by parentheses and followed by the word show. Finally showpage ejects the page from the printer. This produced the page in Figure 3.

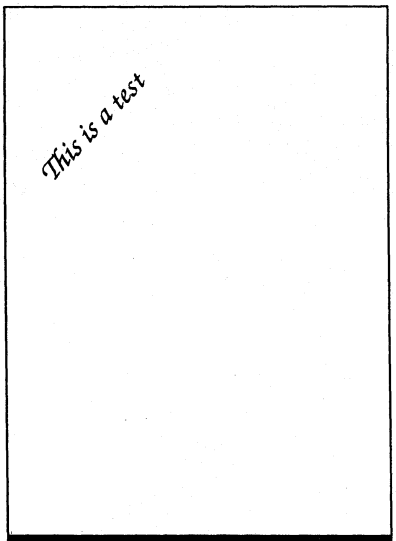


Figure 3

I must confess here that I am not a very experienced, or very good Postscript programmer. As Mark explained last time, the moveto command assumes the 0,0 point to be the bottom left corner of the page. The numbers inserted in the moveto command move from there up and to the right in points. There are 72 points to an inch. The problem is that the rotate command actually rotates the axis, so moveto 370 320 after the 45 rotate command actually moves about 5 inches (370/72) in a NorthWest direction, and 4 inches in a NorthEast direction. It takes some triganometry, or in my case trial and error to determine where the type will end up. In my case, it was alot of trial and error.

I finally ended up with the program in Figure 4 which would print our return address and the invitees address in the suitable position. As mentioned above, I'm sure there are better ways to deal with the rotation and moveto commands, but I found something that worked. To test it, I just catted the file shown in Figure 5 out to the printer, and out came Figure 5A. The command was simply:

`cat invtest > /dev/tty11` (tty11 is the port my NEC LC890 printer is attached to.)

The final step was to create the mail merge. This could be done in many ways, including a Fortune:Word Records Processing document, since all Postscript needs is an ASCII document. I chose to just use a simple shell script which would echo out the commands I needed directly to the port that our NEC printer is attached to.

The first shellscript reads my list of names. The shell script is shown in Figure 6. The input list is simply a list of names and addresses with 6 lines per name. I didn't worry about blank lines, because the blank lines in the address were always at the end of the list, e.g. after the city/state line, rather than within the address, like you might have if you always left two lines for the street address.

The first thing I do is set the IFS (Internal Field Separator) to a RETURN. Then the script begins reading in the names with the read N1 command. If N1 equals the word 'Funky' (which is the very last line in my name list file), the program ends. Otherwise it reads in lines 2-6 into the variables $N2 - $N6. Finally it calls my postscript program with the variables $N1-$N6 (which are the address lines for a single address. In order to get it to work, I cat my name list into the readnames procedure:

`cat namelist | readnames`

The prtnames shellscript is the completed postscript program shown in Figure 5 with the echo command added at the beginning, and the variables $1-$6 substituted in. $1 refers to the first argument passed on the command line. If I gave the command prtnames Josh then $1 is equal to Josh. Normally the arguments are separated by spaces, but since I set the IFS to a RETURN, each line gets treated as a single variable, so Josh Lobel is $1.

As mentioned above, if you'd rather, you could simply use Records Processing to create the Postscript file to send to the printer. When you create your final output document, choose a printer type of PLAIN1 and a left margin of 0. Also, you'll probably want to delete any page breaks in the final docu-

ment since the Postscript showpage command ejects a page.

Hopefully this may be helpful to some of our readers who have Postscript pritners available to them, or at least stimulate some curiosity. To use a Postscript printer with the Fortune, you need to connect it with an RS-232 (serial connection) rather than appletalk. Next time we'll look at a more practical Postscript task of printing cassette labels for a video store from a database.❑

Josh Lobel

```
/ZapfChancery-MediumItalic findfont 18 scale-
font setfont
45 rotate
715 -072 moveto
90 rotate
(Arlington, MA  02174) show
270 rotate
695 -46 moveto
90 rotate
(37 Sages Street) show
270 rotate
675 -094 moveto
90 rotate
(Laura Wiener and Josh Lobel) show
225 rotate
-0 0 moveto
315 rotate
/ZapfChancery-MediumItalic findfont 24 scale-
font setfont
-100 390 moveto
(Ms. Andrea Brown) show
-100 367 moveto
(Mr. Peter Ficus) show
-100 344 moveto
(777 Park Avenue) show
-100 321 moveto
(Suite 770B) show
-100 298 moveto
(New York, NY  20012) show

showpage
```

Figure 4 invtest
Program to print entire return address and address with postscript. Produces Figure 2

```
:
echo "/ZapfChancery-MediumItalic findfont 18
scalefont setfont
45 rotate
785 -02 moveto
90 rotate
(Arlington, MA  02174) show
270 rotate
765 24 moveto
90 rotate
(37 Sages Street) show
270 rotate
745 -24 moveto
90 rotate
(Laura Wiener and Josh Lobel) show
225 rotate
-0 0 moveto
315 rotate
/ZapfChancery-MediumItalic findfont 24 scale-
font setfont
-145 440 moveto
($1) show
-145 417 moveto
($2) show
-145 394 moveto
($3) show
-145 371 moveto
($4) show
-145 348 moveto
($5) show
-145 325 moveto
($6) show

showpage" > /dev/tty11
```

Figure 5 prtnames
Like Figure 4, but feeds variables $1-$6 to program and redirects output directly to Laserprinter on /dev/tty11.

```
:
IFS='
'
while :
do
read N1
if [ ''$N1'' = ''Funky'' ]
then
      exit
fi
read N2
read N3
read N4
read N5
read N6
prtnames $N1 $N2 $N3 $N4 $N5 $N6
done
```

Figure 6 -- readnames
feeds lines 1-6 from address list as variables to prtnames

Classified

This Classified section of **/u/fortune** *news* is designed to serve our subscribers and Fortune computer users in general. In order to help everyone use their Fortune hardware and software to its utmost, we are providing a forum for the selling and the buying of used equipment. We will be including this Classified section in each issue of **/u/fortune** *news* as long as there is sufficient interest. So, if you would like to find something for your Fortune or locate a buyer for your equipment, you should not overlook this valuable resource. If you are interested in a listing in this section of **/u/fortune** *news*, please call us at (617) 894-6900.

FOR SALE: 32:16 w/5 Terms

30 MB hard disk, QUME letter quality printer, 12 serial ports, 1 console plus 5 terminals, Fortune:Word, BASIC

No Reasonable offer refused
Call Ted at 203 725-4000 4/90

Fortune 32:16 XP30

1 MB Ram
4 port controller
CRT and keybaord
For:Pro 2.0
Extended Fortune:Word 3.0
Fortune-to-Fortune Copy 1.2
Call **Larry Zoeller** (312) 558-4575 4/90

Fortune XP 32:16 with 4 terminals

1 Meg of RAM, 70 MB hard disk, with Fortune:Word, Multiplan, Business Basic, all documentation.

Call Don at (617) 267-6500 for more information. 11/89

WANTED -- 32:16 Software for Job Shop Management

of production of single or lot quantities to customer's specifications, estimating based on past history, scheduling of labor and machines, tracking production and promised "Due Dates", material purchasing, accurate job costing and accounting features.
Please call Dave at 419 294-1987 2/90

Fortune 32:16 XP 70

1 Meg Memory
13-70MB Hard Disks
Ports: 12 serial & 1 Parallel
60 MB Tape Expansion
Fortune Basic Workstation

Will take Best Offer
Call Rena at (301) 907-8900 10/89

Fortune 32:16 -- 1 meg RAM, 50 meg

Console + 1 Terminal
1 Meg Memory (4x256)
50 meg Maxtor Drive
C and Pascal compilers
Fortune:Word 2.0
Extra WE Power supplies

Call Chuck at 203 937-5063 2/90

32:16, 6 terminals, 2 printers!!!

Hardware: 70 MB with 30 meg/20 meg tape exp cab 2 MB RAM, 2 4-port CommA's. **NEC 7710 & 3500 printers.**

Software: For:Pro 2.0, Fort:Wrd 3.0, Document Conversion, Handshake ITE & COLT, Kermit
Contact: Susan (203) 972-3400 2/90

Fortunes For Sale - Priced Right!

Two - 32:16's (20 & 30 MB HD)
One - SX70T (2MB memory)

For:Pro 2.1, Multiplan, Ext. Fortune:Word BASIC 6.5.12, IDOL, BAS, Dev. Utils., Communications, other software. Call **Russ** or leave message at (708) 510-0088. 1/90

Fully Loaded Fortune XP-45

If you have been looking for the perfect expansion system, this is it.

Start with a beefy 45Mb hard drive, and 1 Mb of RAM, expandable to 2 Mb. Of course, there are 2 four-port Comm A boards, plus **Fortune:Link LAN with the companion board for your other system.** System console is an amber FIS 1000 terminal and keyboard.

For:Pro level is 2.0. Fortune:Windows, Fortune:Word 3.0, Multiplan, ITE, SST's Handshake Colt ,Informix and Development Utilities are all included.

This is one whale of a computer.

Call Mike Eisen at (619) 323-4555. 2/90

Used Fortune Parts For Sale

WD06
CRT Electronics and Tube
256KB Memory Board
800KB Floppy Drive
Comm A 4 Port Board
Video Controller
1 MB XP Mother Board
10 MB Hard Disk
20 MB Hard Disk (C20)
and more...

All components are fully functional, supplies are limited.

For more information, call **Jim** at **(203) 828-4811** 12/89

Fortune 32:16 XP

Hardware: 45 MB Hard Disk, 2 MB RAM, Terminal and keyboard, 4 ports

Software: For:Pro V2.1, Rubix

All Cables and Books included
Price: Best Offer
Call: Brad Hunter at (313) 296-7130 1/90

Fortune 32:16 System

30MB hard disk with 256KB, 800K flexible drive, 4 ports, console terminal plus 2 Fortune Intelligent Work Stations (12" monochrome display). Software: Fortune:Word, Multiplan and Sequitur, plus Hayes Smart Modem, 300 baud. Form more information, call Laurie at (602) 795-8870. 11/89

20MB 32:16 - Includes For:Pro 1.7.4.1, 768K RAM, console, 2 terminals, 4-port Comm A and a **20MB Tape Expansion Box.** Software includes: SMC BAS modules, Extended Fortune:Word, Developtment Utilities, and a C compiler. Purchased direct from Fortune, Hardly used, will take best offer. Call Gerry Armel at (508) 285-3040. 9/89

Now your Fortune can talk to a PC!!!

With our fxfer, floppy transfer program, you can read and write a Fortune floppy disk with an IBM PC or compatible.

With Fortune:Word<-->Word Perfect, Two of the best word processors can exchange documents preserving margins, tabs, indents, bold, underline, etc. (Fortune:Word to Microsoft Word on its way next.)

They said it couldn't be done, but we did it, and you can benefit.

Call the folks at /u/fortune news for more information:
(617) 894-6900.

A message to our readers:

If you have been missing **/u/fortune *news*** for the last couple of months, it's not your postperson's fault. We've been very tardy in getting out this issue. The last one was volume 7 number 2. We have renamed this issue April -- there won't be a March, but we will be catching up with much more frequent issues in the coming months. We're sorry for the confusion.

Coming soon:

Information on a printcap entry for the new HP Laserjet 3 that uses scalable fonts for even better looking documents.
The latest scoop on SCI/Fortune's newest entries into the Fortune 5000 line.
Some thoughts on the pros and cons of upgrading your system
Using a database program to print to a postscript printer
Faster ways to backup with tape
Using a modem on 32:16's and 5000's
Exporting data from a BASIC file

/u/fortune news
225 Crescent Street
Waltham, MA 02154

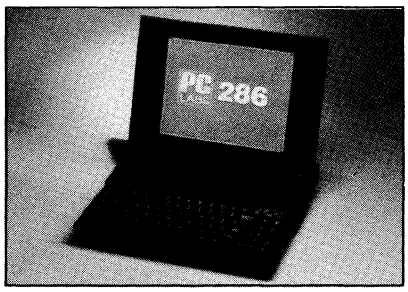
If you're reading a xeroxed copy of **/u/fortune *news***, don't. Copying this publication is prohibited. We depend on subscription revenue to continue publishing **/u/fortune *news***. Please play fair. There aren't many of us out there and we have to stick together.

/u/fortune* *news*

The Newsletter for Users of SCI/Fortune Computers May 1990 / Volume 7 Number 4

SCI/Fortune Debuts a "Top of the Line" 88000 RISC Computer and a 286 Laptop

See "News from Fortune" -- Page 16



Ray's experience with IBM -- A play in Two Acts

The BASIC Advisor -- Page 4

- Dave Kloes continues his discussion of menus
- Using the latest Hewlett Packard Series III in /u/help
- UNIX Basics -- a useful shellscript
- Classified ads

/u/fortune *news*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz

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, Inc. are affiliated with SCI/Fortune Systems.

Subscriptions cost $60 for one year within the USA. For foreign subscriptions, add $36 each year for added postage and handling costs.

Back issues are offered at:

Volume 1, Numbers 1 - 8	$2/each - $14/set
Volume 2, Numbers 1 - 7	$3/each - $18/set
Volume 3, Numbers 1 - 12	$4/each - $44/set
Volume 4, Numbers 1 - 12	$4/each - $44/set
Volume 5, Numbers 1 - 12	$5/each - $55/set
Volume 6, Numbers 1 -	$5/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $15 each, quantity discounts available. 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 the contents of **/u/fortune *news***.

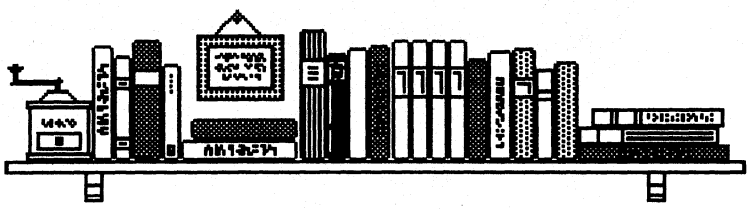
For more information, call or write:

The Cambridge Consortium, Inc.
225 Crescent Street
Waltham, MA 02154
(617) 894-6900

The text for this newsletter was prepared using an SCI/Fortune 5000 for both text input and typesetting. The typesetting was accomplished by using the Aldus Pagemaker program under VP/ix, which is a DOS emulation.

CONTENTS

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.

July 18, 1990

We seem to be at a crossroads with Thoroughbred Software. It may be just the slow months of summer, but the BASIC Advisor is receiving fewer and fewer calls for assistance with the BAS applications, and Solution IV is too new to confuse anyone yet. Although there are still a few 32:16 users around, most of our original readers have moved on to newer machines and, in some cases, different accounting software. Many of those who have stayed with third generation Thoroughbred are now using a Level F or Level G version, both of which are very similar to the Fortune BAS software. We will be taking a short break from our question and answer format, and when we return we will field inquiries from all Thoroughbred Software users, not just users of BAS.

In the meantime we will be featuring a few odds and ends which I guess we can call collectively "How I Spent My Summer Vacation". We begin with a play in two acts called "Who Did You Say You Were?" Some names have been changed or mangled to protect my backside.

ACT I

Scene 1: an office in Baltimore. The temperature outside is approaching 100, and a window air conditioner is straining to hold the occupants in its keep at a level just below the clothing saturation point. A pensive, shirt-sleeved man with loosened tie sits at his computer terminal busily tapping in the final program lines of a week-long project.

Ray: Tap! Tap! Tap!

Telephone: Ring, ring, ring, ring!

Ray: It's your nickel.

Telephone: Is this Mr. Ray Murn'l of BASIC Software?

Ray: This is Ray Wannall, pronounced the way it's spelled, accent on the last syllable.

Telephone: Hello, Mr. Murn'l, my name is Isaac Moneymaker.

Ray: Why don't you just call me Ray.

Isaac: OK, Mr. Murn'l. As I was saying, this is Isaac Byron Moneymaker and I'm from IBM. Did you receive the literature I sent you about becoming an IBM Business Partner?

Ray: Yes. I believe it went out with yesterday's trash.

Isaac: Well, I'm sure you read it. After all, WE ARE IBM. The reason I'm calling is to see if we can set up an appointment at your offices. We are very much interested in exploring the possibility of a BASIC Software/IBM Business Partnership. When would you be free?

Ray: I get it! This is Frank, right? Ha ha! You old kidder! So you finally got me back for that crushed spider in your secretary's income statement folder! You dog you! Ha ha!

Isaac: I assure you, Mr. Murn'l, IBM takes its business partnerships very seriously. I feel certain that if we can meet you will find it very profitable for BASIC Software.

Ray: Sure, Frank. How about 9:00 AM Thursday?

Isaac: That will be fine. I'll see you then.

Scene 2: several days later in a conference room in the same office building. Seated at the table are Ray, Isaac and Lynne, co-owner of BASIC Software.

Isaac: (Displaying a color brochure) We have recently released a new computer called the RISC System/6000 running under our proprietary AIX operating system, and we are very pleased with the early sales results. This is a feature-rich machine onto which any applications running in the Unix environment can be easily translated.

Ray: Is this a 386-based box?

Isaac: It has no correlation to either Intel or Motorola. It is our own 32-bit architecture.

Ray: And AIX? Isn't that just IBM's way of spelling Unix?

Isaac: It is similar to Unix, but it is our own product. As I said, your programs must be translated from Unix into AIX.

Lynne: Time Out! This whole meeting

is a waste of time if Thoroughbred Software doesn't run on this machine. Everyone just cool it while I call Thoroughbred.

Isaac: Mr. Murn'l, I feel certain that if this so-called Thoroughbred runs under Unix it will run on the RS/6000.

Lynne: Dial, dial.

Ray: Gee, I love your blue suit. Do you buy off the rack?

Lynne: (On the phone) I need to know if you all are up and running on the IBM RS/6000. Uh huh. Uh huh.

Ray: So, have you picked up any good "business partners" lately?

Isaac: We are in the initial stages of this project in the Baltimore area, but the program has been very successful in other parts of the country.

Lynne: Uh huh. Uh huh.

Isaac: Do you currently sell computer systems?

Lynne: Uh huh. Thanx. Bye. (Click) It's a go. Thoroughbred says the system is selling like hot cakes, and they have been on it for several months. Sorry to interrupt.

Isaac: I was just asking Mr. Murn'l if you sold hardware.

Lynne: Yes, we sell several systems. On the high end we recommend Fortune and SCI systems.

Isaac: Good box! Good box! And what software do you carry other than Thoroughbred?

Ray: Other than Thoroughbred?

Lynne: I like your blue suit. Is it custom tailored?

Isaac: I see. Well, I believe we can be very helpful to each other. Your verticals coupled with a world wide IBM sales force can be very profitable for all. Now, we have several business partner plans, and I'm sure BASIC Software will fit nicely into at least one of them.

Ray: (Aside to Lynne) It's just awe-inspiring how quickly he sized up our business! Maybe that's how they came to own two thirds of the globe.

Lynne: Quiet!

Isaac: Here's a little blue brochure which explains the various business partner plans.

Lynne: Blue! How quaint!

Ray: Quiet!

Isaac: As you can see, business partners fall into one of six categories. There are IBM Authorized Agents, IBM Authorized Application Specialists, IBM Authorized Industry Application Specialists . . .

Ray: (Under his breath) IBM Authorized Lead Generators . . .

Lynne: Sshhh!

Isaac: . . . IBM Authorized Dealers, IBM Authorized Industry Remarketers and Cooperative Software Program Suppliers. Perhaps you would like to be a dealer. Our inventory requirements are very fair. How many computers do you normally purchase in a year?

Lynne: Perhaps it would help if I explain what I expected when I called you.

Ray: YOU CALLED HIM??? HE'S NOT WORKING FOR FRANK???

Lynne: Yes, now be quiet. What we need, Mr. Moneymaker, is a local outlet for a new vertical we are putting on the market. It was my hope that we could work hand in hand with the IBM sales force in moving IBM systems and Thoroughbred software as a turnkey package. We certainly don't want to fill up our offices with boxes of IBM's. Nor do we want to jeopardize our other vertical systems and software sales.

Isaac: I see, so you would be interested in becoming an Applications Specialist.

Ray: Now I am finding all of this hard to swallow. For over ten years we have been selling against IBM because IBM would never do business with anyone who didn't know all of the words in the second verse of their company song. Why, all of a sudden, have you decided to romance low-life such as ourselves?

Isaac: This is a new machine with, we feel, ideal solutions for the business community. We want to take advantage of the vast supply of marvelous software running under Unix and help the top local VAR's become successful in their business ventures.

Ray: How noble of you.

Lynne: So what is the next step?

Isaac: We will set up an appointment for you to come to our downtown offices and see the RS/6000. At that time we can more deeply discuss whichever business partner plan you feel most comfortable with. I will leave these brochures with you. Thank you for your time, Mr. Murn'l. Ms. Crawford. (Exits stage left).

Lynne: So what do you think?

Ray: First of all, I think we should research this new IBM marketing approach and see just what kind of a deal we can expect to write. Second, I think I'd better call Frank and warn him not to turn on his printer today.

ACT II

Scene 1: the lobby of a ten story office building located across the street from Baltimore's Inner Harbor. A large, enclosed rectangular reception counter sits in the middle of a barren, marble room with a fifteen-foot high ceiling. Ray and Lynne enter from stage right, their footsteps echoing. As they try to cross to the elevator a very obese woman in a security uniform appears from behind the reception counter.

Very Obese Woman: Hold it right there, bucko! You've got to sign in!

Lynne: Oh, I'm sorry. I didn't see you wallowing there.

Ray: (Picks up a pen and starts to write in one of two notebooks lying on the counter.)

Very Obese Woman: Who are you here to see?

Lynne: Mr. Isaac Byron Moneymaker at IBM.

Very Obese Woman: Then you are signing the wrong book. For IBM you have to sign the blue notebook.

Ray: I should have known. (Begins writing in the other book.)

Lynne: We don't know where to find Mr. Moneymaker. Can you tell us which floor he is on?

Very Obese Woman: (Disgusted) Just a minute. (Looks through papers on a clipboard.) He's on the fourth floor.

Lynne: Thank you. (Turns and exits with Ray through elevator door upstage.)

Scene 2: Another reception desk in an upstairs lobby. This room is decorated in a more modern fashion. A slim, middle aged receptionist in a blue business suit is seated behind a counter, stage right. Lynne and Ray enter from an elevator door upstage and cross to the counter.

Lynne: We have an appointment with Mr. Isaac Byron Moneymaker.

Receptionist: Sign in, please.

Ray: Where do I check my weapon?

Lynne: Here, let me do it this time. (Begins writing in the only notebook on the counter.)

Ray: I believe Mr. Moneymaker is expecting us.

Receptionist: Let me call him. (Dials a phone.) Mr. Moneymaker? Your appointment is here. Okay. (To Ray) Mr. Moneymaker will see you now. He's on the sixth floor.

Ray: Is that where they keep the guard dogs? (Exits with Lynne upstage.)

Scene 3: another lobby and reception desk exactly as before. Ray and Lynne enter from an elevator door upstage and cross to the desk where they are met by Isaac, who escorts them to an adjoining conference room. The conference room contains a table, a computer system with a 19 inch color console screen, a slide projector, and a large sign on an easel which says "THE IBM/BASIC SOFTWARE BUSINESS PARTNERSHIP". Ray and Lynne sit at one side of the conference table, and Isaac stands opposite them and picks up the slide projector control. As he speaks, Isaac randomly clicks through slides which display on the far wall.

Isaac: The history of the AIX family of computer solutions is over five years old, beginning with the PS-2 and its associated applications solutions. Operating in top MIP capacity with megaflop benchmarks . . .

Ray: Yawn.

Isaac: . . . dedicated to the proposition that . . .

Ray: (Aside to Lynne) Have you ever seen so many blue suits in one place?

Isaac: . . . of the people, for the people, to the people . . .

Lynne: (Aside to Ray) How about those Orioles?

Isaac: . . . with liberty and justice for all. Any questions so far?

Ray: Yes. Can we dispense with the rest of the slides and talk about business? I'm sure you have a great product, but I'm becoming a little over MIPped and Megaflopped.

Isaac: Sure! That's why we are here. (Sits at table opposite Ray and Lynne.) It seems we have settled on the IBM Authorized Applications Specialist business partner agreement. Do you have any questions about it?

Lynne: What do you expect from us in this deal?

Isaac: You agree to sell six to ten systems per year for us. You will work with various members of our specialized sales force and develop relationships with them. With any luck, one or more of them will agree to market your software. If at the end of the first year you have exceeded your quota by at least ten percent IBM will buy you a lunch. We do, of course, expect you to turn over all of your leads to our sales people and agree not to sell any competing product to any prospect. After all, that's only fair to IBM. Would you like to examine the RS/6000 now?

Ray: Gee, I can't wait. You certainly have motivated me to abandon Fortune.

(A six-foot, eight-inch tall man in a blue suit enters the conference room.)

Isaac: This is Irving Bob Mipburner, our technical specialist. He can answer any questions you may have regarding AIX.

(The four of them exit.)

Scene 4: a room full of computers. Isaac and Irving escort Ray and Lynne into the room and up to one of the computers. The computer display shows several windows in a rainbow of colors and gaudy graphics. Irving fingers the keyboard.

Irving: As you can see, this window has a menu which allows any user to conduct all system housekeeping: account maintenance, backup and restore, set date and time, etc. With one simple keystroke I can bring the system into the shell. With another, I'm back to the window menu. See? Shell, window menu, shell, window menu, shell . . .

Ray: Stop it! You're making me dizzy! May I touch it?

Irving: Sure! Help yourself! (Steps aside as Ray approaches the keyboard.)

Ray: (Entering on the keyboard) cd /

<CR> ls -l <CR>. Hey, look at this! Tell me, Isaac, what is that word on the screen?

Isaac: (Under his breath) Unimmphh.

Lynne: What was that?

Isaac: Uh, unix.

Lynne: Oh.

Ray: Does it also have uucp?

Irving: I'll check on that for you. (Starts to exit.)

Ray: Wait a minute, does it feature tar or cpio?

Irving: I'm not sure, but if you will let me check with someone I can answer that for you.

Ray: Never mind. (Types into the computer: find / -name uucp -print.) Here it is, uucp. It won't be necessary to check with anyone. It appears our software will be fine on this box.

Isaac: Great! Let's talk contract!

Lynne: With all due respect, we have found a dealer who will sell us these machines at a considerable savings. We must pursue that avenue before we can agree to sign a contract. Perhaps we can get back to you?

Isaac: I understand, but you must consider just how much you are sacrificing for the thousands of dollars you save when you buy from a dealer.

Ray: Such as?

Isaac: Our sales people may not be quite as cooperative. You will have to spend a half a day putting together the machines, and you will have to deliver them to the end user yourself.

Lynne: Don't forget about that lunch!

Ray: I guess you're right. We certainly will give your offer our deepest consideration, Mr. Moneyman. Thank you and Good-bye.

(Dim lights. Applause.)

System Administration: Part 36

No Longer requires a Degree in UNIX Wizardry

This series is a must for all owners and users of the SCI/Fortune computer. Dave is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the SCI/Fortune computer, and is past president of the Houston UNIX User's Group. He contributes independently to **/u/fortune *news***.

Creating Friendly Menus on a 5000 *by Dave Kloes*

In the last issue, we began discussing a simple "user-friendly" shell program that can be used as a front end menu for your users. If you remember, we left off at line 26 of the shell script. In this issue, we pick up beginning at line 27.

If the user enters "2" at the prompt on line 22, the "case" statement on line 27 is executed. Remember in the last issue that we said that all of the commands for a particular case are executed until a double semi-colon (;;) is seen. The actions if "2" is selected from the menu then are lines 27 thru 40:

```
-----------------------------------------------------------------
    27            2) clear
    28               echo "
    29    Backup of /usr files ...
    30
    31    Insert tape and depress <RETURN> to continue: \c"
    32               read bunk
    33               if tar cvfbk /dev/rctmini 20 37000 /usr
    34                  then echo "
    35    Backup successful."
    36               else echo "
    37    Backup not successful."
    38               fi
    39               echo "^GDepress <RETURN> to continue: \c"
    40               read bunk;;
-----------------------------------------------------------------
```

```
    27            2) clear
```

First of all, line 27 clears the screen.

At the top of the screen, you would see what is echoed in lines 28 thru 31. Remember that putting multiple lines to be displayed in the same echo statement is much more efficent than echoing a line at a time.

```
    28               echo "
    29    Backup of /usr files ...
    30
    31    Insert tape and depress <RETURN> to continue: \c"
```

The following would give us the same display but it would take longer to "paint" the screen:

```
    28    echo ""
    29    echo "Backup of /usr files ..."
    30    echo ""
    31    echo "Insert tape and depress <RETURN> to continue: \c"
```

Here we have four "echo" commands to run instead of one.

```
    32            read bunk
```

The "read" statement in line 32 would wait for user input and put the value of whatever they enter into the variable "bunk". We use this variable name whenever we use a variable that has no useful purpose in life except to give the user a chance to read what is on the screen and take the appropriate action. Even though we told the user to depress <RETURN>, the script will continue regardless of what they enter.

```
    33               if tar cvfbk /dev/rctmini 20 37000 /usr
    34                  then echo "
    35    Backup successful."
    36               else echo "
    37    Backup not successful."
    38               fi
```

The "if" statement is used whenever we want to test for some result and take an action based on that result. One basic use of the "if" is to simply test to see if a command is successful or not as we have done here. If the "tar" backup command is successful then "Backup successful." is displayed on the screen. If the command fails for any reason, then "Backup not successful." is displayed.

Whenever we use the "if", it must have a corresponding "fi" (if spelled backwards for those of you that are dozing). The "then" on line 34 is also a required keyword and is associated with a successful result. The "else" means the same as "otherwise" but is not a required keyword. In other words, we can use the "if" to test for a successful result without testing for a an unsuccessful result.

The last element of the "if" statement is the "elif" (stands for else if) which is used to test for other conditions. This all may sound confusing so let's look at a simple menu example to try to clarify things:

1. Display today's date

2. Tell me who is on the system
3. Display a three month calendar
4. Exit Menu

If our script displays the above menu and we want to use the "if" statement to determine what action to take, the construct would be as follows:

```
if the user selects 1
        then execute the date command
elif the user selects 2
        then execute the who command
elif the user selects 3
        then execute the cal command
elif the user selects 4
        then execute the exit command
else echo an error message
fi
```

It would be nice if this is what the program actually looked like, however, we will leave the "test" portion of the "if" statement alone for now and concentrate on the construct fundamentals. As you can see, the program checks for the various selections a user can make from the menu and takes a corresponding action. The "else" condition is executed if none of the other conditions are met.

Remember we said that the "case" statement was a glorified "if" statement. In fact, it is much easier to use the "case" statement for a job like this:

```
case $answer in
        1) execute the date command;;
        2) execute the who command;;
        3) execute the cal command;;
        4) execute the exit command;;
        *) echo an error message;;
esac
```

In general, we use "case" instead of "if" when we have a choice.

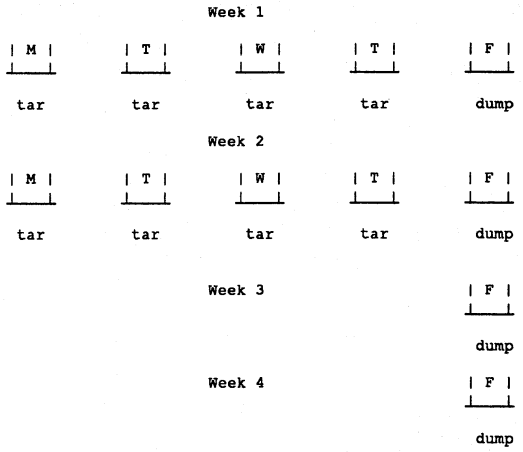
We have been mainly talking about shell programming so far. Let's change the subject and talk about the "tar" command we are using for the "daily" backup.

Backups are probably the most important job that a System Administrator has. Here are some backup objectives that we strive for:

1. Minimize down time.
2. Be able to restore the system with minimal data loss.
3. Keep the amount of time required to backup the system to a minimum.

Keep in mind that these objectives do conflict with each other. For example, the backup that is done the quickest may not minimize our down time.

It is unfortunate that most people learn the backup lesson the hard way. Here are some strategies that we use with our clients to achieve the above objectives:



First of all, notice that in this backup scheme, 12 tapes are needed (NOBODY is still doing diskette backups - are they???). We realize that cartridge or mini-cartridge tapes cost anywhere from $18-35 each. What would it cost you if your people had to re-enter a couple of weeks worth of data??? How much would it cost your company if your computer was down for a few days because all of the software had to be reinstalled from scratch??? It still amazes us how much it hurts clients to spend this money. In any case, the idea here is that the system is backed up daily in one way or another. Obviously, the frequency of backups can be adjusted based on the frequency of system use. Basically, it works this way:

1. The Monday thru Thursday tapes are rotated every other week. For example, the Monday tape from week 1 will be used as the Monday tape for week three.

2. The Friday tapes are rotated every four weeks. For example, the Friday tape for week 1 will not be used until week 1 of the following month.

With this scheme, we can go back as far a one month to recover data. Remember, that a file that is damaged may not be noticed right away. For example, if you have a file that is only accessed when you do a month end closing, you are in trouble if your backups only go back as far as one week. Some of our clients go as far as to keep a month end tape for each month of the year. How about a file that is accessed only once when you close the books at the end of the year?

Notice also that we are using two different types of backups - "tar" and "dump". There are advantages and disadvantages to both methods, however, this scheme gives us some flexibility in recovering files, directories and the entire system.

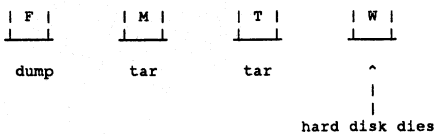
"tar" or "cpio"

A major advantage of a "tar" or "cpio" backup is that you can extract individual files or whole directories. A major disadvantage is that these backups are slower than "dump". The "daily" backup uses these commands to backup only the day-to-day user directories.

"dump"

The "dump" provides us with a faster backup that includes a particular filesystem or partition. If your disk is "partitioned" into separate sections, an individual "dump" must be performed on EACH partition. Since system files change less frequenly than user files, this type of backup is overkill if done on a daily basis. If your system files are altered very little from one week to another (adding or deleting users, printers, modems, terminals) then you may opt to do the system backup less frequently. A major disadvantage of the "dump" is that directories cannot be extracted. Either the whole system or individual files are restored.

Let's look at a disaster scenario and see how the tapes created using this scheme might be used:



We faithfully do our system backup on Friday and our daily backups on Monday and Tuesday. On Wednesday, our hard disk gives in to fatigue and has to be replaced. First of all, if you have made your "emergency boot floppy" (discussed in a previous article), we would minimally load one diskette of the SCO Xenix operating system and then boot from our "emergency" floppy. Using this floppy, we would restore the entire system from our Friday backup. We would then boot the system from the hard disk and restore Tuesday's daily backup. In less than a couple of hours, we are back up and running. Assuming that our backups are done at the end of each day, the only data loss would be what was entered on Wednesday.

Another helpful hint would be to keep some of the tapes on-site and some off-site. Doing your backups religiously every day does not do any good if all of the tapes are destroyed in a fire.

Since the system should be "quiescent" (users should be off of the system) while a backup is being done, the typical excuse for not doing a backup is that the users are too busy using the system. For those with this "excuse", remember that backups can be performed automatically in the evening when users are not on the system.

Tapes should also be replaced occasionally. Over time, they do wear down. In general, if you get "read" or "write" errors consistently on the same tape, it is a candidate for the garbage can. There is nothing more frustrating than to do your backups as prescribed only to find out the tape cannot be used to restore the system because it is bad.

Let's look at the "tar" command we used in line 33:

```
tar cvfbk /dev/rctmini 20 37000 /usr
    ^^^
    |||-------------------^
    ||-----------------^
    |--------^
```

c tells us to "create" the backup starting at the beginning of the tape

v "verbose" - displays the files as they are backed up

f expect an argument to follow (in this case /dev/rctmini) which specifies the device to be used. This device name refers to a mini-cartridge device. If the device you want to use is the default device for the system (as specified in the /etc/default directory) then the "f" and the device name are not needed on the command line.

b expect an argument to follow (in this case 20) which specifies the blocking factor to be used. In other words, write the data 20 blocks at a time. Since 20 is the default on most systems, the "b" option and the 20 in the command line are not needed. The higher the blocking factor, the faster the backup but the more possibility of error. The lower the blocking factor the slower the backup but more accurate.

We realize that cartridge or mini-cartridge tapes cost anywhere from $18-35 each. What would it cost you if your people had to re-enter a couple of weeks worth of data??? How much would it cost your company if your computer was down for a few days because all of the software had to be reinstalled from scratch??? It still amazes us how much it hurts clients to spend this money.

k expect an argument to follow (in this case 37000) which specified the capacity of the tape in Kbytes. The actual capacity of the tape we are using is 40MB - give or take some depending on the number of bad blocks on the tape. We probably could specify as high as 39000K and be safe. By specifying this option, the "tar" knows when to prompt for second and subsequent tapes.

/usr specifies the files and/or directories to be backed up. Separate each name with a space. This parameter should be the directory or directories where your daily work is done (i.e. accounting, word processing, spreadsheet, etc.).

```
39              echo "^GDepress <RETURN> to continue: \c"
40              read bunk;;
```

The last two lines for selection 2 on the menu display the message and wait for the user to depress the <RETURN> key. This allows the user the opportunity to see the various messages that are displayed for either a successful or not successful backup. The ^G (control-G) in line 39 is the "ring bell" control sequence for many terminals. Other control characters can be used for things like reverse video, blinking or highlighted characters. Remember that these sequences are terminal dependent.

In the next issue, we will continue our discussion of backups, shell scripts and the other selections on "kmenu".❑

UNIX Basics

A shellscript that makes transferring files faster and easier

In this article, we would like to explain a shellscript that we often use. By doing so, we hope to accomplish several purposes.

First, we want to introduce the reader to the power of shellscripts. Shellscripts can be defined as collections of common Unix commands that are put together so that the collection will perform some bit of work. As such, shellscripts are great ways to reduce the amount of drudgery and tedium that can sometimes confront the computer user. In addition to being composed of common Unix commands, shellscripts also utilize some special "shell" commands that make certain activities possible. An example of this would be looping commands such as the "for" command or the "while" command.

Second, we want to walk through an example to give you the flavor of solving a "computer" problem and implementing its solution in the "shell" language. Generally, the two biggest obstacles in solving a problem is 1) being able to break the problem down into sufficiently small steps and 2) translating those small steps into available Unix commands.

Third, we want to introduce you to some useful Unix and shell commands. The shell commands include basename, mv, and uncompress. The shell command that we want to illustrate is the for loop.

Problem To Solve

First, we want to introduce the reader to the power of shellscripts. Shellscripts can be defined as collections of common Unix commands that are put together so that the collection will perform some bit of work. As such, shellscripts are great ways to reduce the amount of drudgery and tedium that can sometimes confront the computer user.

We often use kermit to transfer files between other remote Unix computers and our Fortune. We have written in the past how to use kermit so we won't do that here. However, since kermit is a file transfer program and because typical phone connections to remote computers run at 1200 or 2400 baud and because some of the files we wish to transfer are rather large, we often use the compress program to reduce the amount of space that a file takes up before we transfer it. In this way, we are able to reduce the amount of time it takes to transfer a file - sometimes we can reduce this amount of time by 50 to 75 percent!

As most of our readers know, when one compress compresses a file, it takes the file, reduces the amount of space it takes up, and then leaves a new file on the disk with a .Z appended to its name. For example, suppose we wanted to compress a file called bigfile, we would type the following:

```
compress bigfile
```

after compress was done, we would no longer have a file called bigfile in my directory. However, we would have a file called bigfile.Z in my directory. The .Z extension is compress's way of telling us that a particular file has been compressed.

Now, if we use compress on some remote computer to compress a file before we transfer it to our computer, we will end up with a file with a .Z extension. After the file is compressed, we use kermit to send the file to our computer and after the transfer we notice that the .Z extension has become a .z extension - the upper-case Z has become a lower-case z! That is a real problem because when we want to uncompress the file, the uncompress program looks for a file with a .Z extension and does NOT recognize a .z extension. To remedy this situation, we can use the mv command to rename my file. For example, the following two commands will rename the file and then uncompress it:

```
mv bigfile.z bigfile.Z uncompress bigfile.Z
```

You might say, at this point, "so, what's the big deal - why do we need a shellscript for this?" Well, more often than not, we transfer not one file, but many files. Just think of the work that would be involved in renaming and uncompressing 20, 30 or even 100 files! You see, that's where a shellscript can be a real work-saver.

Problem Specification

So, here's the problem. We have a directory with many files in them with .z extensions and what we want to do is rename everyone of the them in such a way that we change the lower-case z to an upper-case Z. Furthermore, after the case change, we want to uncompress the file. The problem specification involves three major steps:

1) Find the files with a .z extension
2) Change the z to a Z
3) Uncompress the file

We've already seen how to accomplish the third step by using the uncompress command. It is really the first and second steps that prove to be a challenge. So let's break them down into some easier-to-solve components.

Solving Step One

Beginning with first step, we can note that one way to find the files in a directory that end with a .z is to use the ls command like:

```
ls *.z
```

This will list out all the files in the current directory that have a .z extension. Since we want to take each file in turn and do something with it, we need something more than the ls command. In fact, to fully solve step one we need the "for" shell command which provides us with "looping" capability. Let's take a look at the following simple example of the for command:

```
1. for file in `ls *.z`
2. do
3.          echo "The File is called $file"
4. done
```

The above is an example of the for loop which has the following structure:

```
for variable_name in list
do
        body of loop which contains Unix commands
done
```

where

1) variable_name is a shell variable which takes, in turn, whatever "words" are in "list"
2) list is a list of "words" where word is defined, generally, as any sequence of alphanumeric characters separated by "white" space (i.e., spaces and tabs)
3) Body of loop are a series of one or more Unix commands that are executed once for each "word" that is in the list.

So, let's go back to our little example above and dissect it. In line 1 we saw:

```
for file in `ls *.z`
```

First, let's look at `ls *.z` and note that we have used "back-quotes"! When back-quotes are used it means that the command in between the back-quotes is run and its output is used. In our example, then, what ends up between the back-quotes is a list of all the files in our directory that have a .z extension and it is a list just like this that the for command needs. Now, let's suppose that we have three files in our directory that end in a .z and let's suppose that they are called file1.z file2.z and file3.z. What we end up with is a command that looks something like this:

```
for file in "file1.z file2.z file3.z"
```

And because we have three files, the commands in the body of the for loop will be executed three times. Since line 3 in our example is:

```
echo "The File is called $file"
```

We would see echoed on our screen the following:

```
The File is called file1.z
The File is called file2.z
The File is called file3.z
```

Please note that the shell variable, which we called file, first held the value "file1.z" on the first pass through the loop, then it held "file2.z" on the second pass and finally held "file3.z" on the third and final pass through the loop.

We know we've spent some time on explaining the syntax and the workings of the **for** loop, but we did it because it is an extremely useful command in solving many computer problems and because it is central in solving the problem we have set for ourselves in this article.

Thus, up to this point we have solved step 1 in our three step problem specification.

Solving Step Two

Step two was stated as "Change the z to a Z" - for example, if our file name is file1.z we want to end up with file1.Z. Basically, what we want to do is:

1) Strip off the .z extension
2) Add on a new .Z extension
3) Change the old name to the new name

We know we've spent some time on explaining the syntax and the workings of the for loop, but we did it because it is an extremely useful command.

There are a couple ways of doing this but the one we have chosen uses the **basename** command. The basename command, in its general form, is used to strip any suffixes off of some word just leaving the "basename". For example, the output from the following:

```
basename file1.z .z
```

is the word file1. This is what is left after we strip off the .z extension. So, we can accomplish the stripping function with the basename command.

Next we need to add on the new .Z extension. To see how this works, let's start building up our solution using the components we have already figured out:

```
for file in `ls *.z`
do
      BN=`basename $file .z`
done
```

The variable BN, which stands for basename, will hold the leftover file name after the .z has been stripped off. To add a .Z to BN, we simply say use a construction like: $BN.Z - this would map into something like "file1.Z". To change file1.z to file1.Z therefore would require the **mv** command like this:

```
mv $BN.z $BN.Z
```

and if we want to then uncompress our new file, $BN.Z, we would use a command like:

```
uncompress $BN.Z
```

Thus, putting it all together, we have:

```
for file in `ls *.z`
do
      BN=`basename $file .z`
      mv $BN.z $BN.Z
      uncompress $BN.Z
done
```

Problem Solved!

Thus, our problem is solved. However, if you want to actually build a command like the one we have illustrated, there are still a couple of steps you must perform. First, you must type the shellscript into a file using your favorite editor (e.g., screen, ed, vi, etc.). Be careful to use backquotes on the first and third lines of the script.

After the file has been put into a file, you will need to make it executable. Let's say you have named the file with the shellscript in it "upz", then you would use the chmod command to make it executable like:

```
chmod +x upz
```

Once it is executable, you can use it like any other Unix command by typing it to your Unix prompt.

Two More Comments

First, the command as we have written it works quite silently. Generally, we like to see what a command is doing so that if it starts to behave unseemly, we can take some sort of action. In our version of this command we use an echo command to tell us what file it is working on and we put the following command before the basename command:

```
echo "Working on $file"
```

Second, we have solved step two of our overall problem by using the basename command. However, we could have used the tr command which is built to translate one character into another. So, we could have translated the lower-case z to an upper-case Z. The following command would do this:

```
BN=`echo $file | tr z Z`
```

However, there is one major drawback to this method. The tr command will indiscriminately change all lower-case z's to upper-case Z's and what we want to do is just change the extension. So the above command would change the file called zoo.z to Zoo.Z and pizza.z to piZZa.Z, which as you might guess, is definitely not what we want.

Conclusion

In this article, we explained how we solved a problem by writing a shellscript using Unix commands. In the process, we hope we have helped our readers understand how to take a problem, break it down into solvable parts and then write a shellscript to solve that problem. In addition, we discussed the basename command and the for loop. ❑

Mark Palmerino

News From SCI/Fortune

In the past month or so, **SCI/Fortune** has announced a slew of new products which are keeping pace with the latest technological developments. Among these are a 20Mhz 386SX computer, a 33Mhz 386 with cache, 88000 based RISC computer, and a 16 Mhz 286 based laptop. This fills out the product line rather nicely, providing appropriate hardware and software for single user 286 requirements, all the way through settings that require 20-50+ users running multi-user UNIX based applications. **Fortune:Word** and **Fortune:Windows** are available on all of the UNIX based machines.

As is the case with most of the computer world of late, the prices are still coming down. We all groan when faced with the news that the thousands we spent back in 1983 for a 20 megabyte 32:16 will now buy us much more powerful machines with huge drives, capable of running over 50 users. Of course, that's good news when it comes to upgrading. It would be a lot worse if things had gone the other way and it now cost us hundreds of thousands to upgrade. Besides, we find that most users have gotten a lot of use out of their 32:16's. (We certainly have.) In this issue, we'll present the specs on the new equipment from **SCI/Fortune**, and next month, we'll share some of our thoughts about upgrading, now that there are so many choices.

Fortune 5000 -- Model 5386/33 Cache Base System

The Fortune 5386/33 cache base system include the items listed below. The cabinet is a tower design with five half-height peripheral bays, three of which are accessible from the front panel. The dual floppy drive controller on the motherboard supports both 1.2 MB 5.25" and 1.44 MB 3.5" drives. The specifications for the Fortune 5386/33 cache system are listed below. Additional options can be obtained from your dealer. A *5386 User's Guide* is included with the base system.

Intel 80386 32-bit microprocessor
33 Mhz operation
8 Mhz operation keyboard selectable
Motherboard Features
- Capacity for up to 8 MB Memory
- Phoenix ROM BIOS PLUS
- 32 KB cache memory
- 8 16-bit industry standard expansion slots
- Dual floppy drive controller
- Socket for optional 80387 math coprocessor
- 1 console keyboard port,
- 1 bi-directional parallel port, and 2 serial ports
- Clock/calendar with battery backup

Tower cabinet supporting a maximum of 5 half-height devices
Enhanced 101-key keyboard
250 watt power supply

Fortune 5000 -- Model 5386SX/20 Base System

The Fortune 5386SX/20 base system includes the items listed below. The cabinet is a 17" wide desktop Baby-AT style which supports a maximum of three half-height devices, each accessible from the front panel. The dual floppy drive controller on the motherboard supports both 1.2 MB 5.25" and 1.44 MB 3.5" drives. The specifications for the Fortune 5386SX/20 are listed below. Additional hardware options (disk drives, video adapters, etc.) can be obtained from your dealer. A *5386 User's Guide* is included with the base system.

Intel 80386SX microprocessor
20 Mhz operation with 0 wait states
8 Mhz operation keyboard selectable
Motherboard Features
- Capacity for up to 4 MB Memory
- Phoenix ROM BIOS PLUS
- 8 16-bit industry standard expansion slots
- IDE hard disk controller
- Dual floppy drive controller
- Socket for optional 80387SX/20 math coprocessor
- 1 console keyboard port,
- 1 bi-directional parallel port, and 2 serial ports
- Clock/calendar with battery backup

Desktop cabinet supporting a maximum of 3 half-height devices
Enhanced 101-key keyboard
200 watt power supply

RISC/UNIX Fortune 5880 System

SCI/Fortune is pleased to announce the most powerful system we've ever offered: the Fortune 5880 RISC-based (Reduced Instruction Set Computer) UNIX system. This product line brings mainframe technology to the multi-user environment with its unique design. The base system provides support for 27 devices, and the addition of a single add-in board offers expandability to over 50. The Fortune 5880 features both the power of the Motorola 88000 RISC architecture and the usability of the Intel 80386 CISC (Complex Instruction Set Computer) architecture. The M88000 performs all UNIX system functions, while the 80386 performs all I/O operations. Two processing speeds and four RAM packages are offered, as well as three sizes of high-performance ESDI hard disk drives and a number of other options, permitting a level of configuration flexibility which can provide solutions to many business needs.

HARDWARE

The Fortune 5880 platform includes a computational subsystem around the Motorola 88000 processor, which consists of

three chips: the 88100 CPU, containing integer and floating point circuitry, and two 88200 memory management unit chips which handle memory management and the dual caches (16 KB for instructions and 16 KB for data). Both 16 and 20 Mhz versions of the 88000 are available. The I/O subsystem features an Intel 20 Mhz 80386 and 1 MB of RAM, with seven 16-bit industry standard expansion slots.

32-bit parity protected memory is offered in 8, 12, 16, or 20 MB configurations, and must be ordered in conjunction wit the processor desired.

The communication subsystem includes one bi-directional parallel port, two serial ports, one console keyboard port, and a 24-port intelligent asynchronous serial port board as standard equipment, so that the Fortune 5880 will handle 27 users at the low end. An additional 24-port board can be added to the computer to expand system access to more than 50 users.

The cabinet design features our familiar tower packaging, which offers space for up to five half-height devices. Standard on the Fortune 5880 are a Mono/CGA/Hercules video controller for the console, a 1.2 MB 5.25" floppy disk drive and a 150 MB 5.25" QIC-02 streaming tape cartridge drive. The power supply is rated at 250 watts, providing plenty of power for both standard and optional devices and ad-in boards. An on-board floppy disk controller can handle two floppy disk drives, and the high-performance ESDI controller supports one or two optional hard disk drives. Drives in unformatted sizes of 182 MB, 382 MB, and 767 MB are available.

SOFTWARE

The operating system provided is a multi-user version of AT&T's UNIX V.3.2 certified 88000 Binary Compatibility Standard (BCS). Included are all commands, utilities, and other programs that are a part of the standard AT&T release, as well as the portable C compiler and an ANSI-standard Fortran 77. MS-DOS 4.01 completes the standard package. The operating system support all software which meets 88open Consortium Ltd. standards.

A wide range of 88open compliant software applications are available in the marketplace today, such as products from AcuCobol, Affinity, Progress, Thoroughbred, and Unify. In all, well over 60 ISVs are in the certification process at this time, with over 125 products pending.

IN addition, **SCI/Fortune's** highly respected word processing package, **Fortune:Word**, as well as the **Fortune:Word Thesaurus Option** and **Fortune:Windows**, are available

Performance and Limitations

In dhrystone testing, the Fortune 5880 benchmarked at greater than two times 486/25 Mhz machines and greater than two times the Sparcstation 1s at 20 Mhz (i.e. approximately 48,000 dhrystones). The 5880 turned in the fastest AIM multi-tasking performance, edging out the MIPS R300/3010.

Operations performed in the MS-DOS operating system environment are limited to the console. MS-DOS use tends to degrade performance for other users on the system. DRAM is integrated into the computational subsystem, and is not upgradeable.

SUPPORT

SCI/Fortune's Fortune 5880 product line is provided with a one year warranty.

Dynabook 286 Portable Computer

SCI/Fortune is pleased to announce a brand new product -- the Dynabook Technologies portable microcomputer. The Dynabook is one of the smallest, fastest, most powerful full-function portables ever. Manufactured by SCI in Huntsville, Alabama, the Dynabook microcomputer product line adds significant breadth to our current system offerings.

Product Features

SCI/Fortune's Dynabook microcomputer features an 80C286 microprocessor with clock speeds of 8 and 16 Mhz. Mass storage is accomplished with a 40 MB internal hard disk drive with an average access time of 28 ms, and with a 3.5" 1.44 MB floppy disk drive. 1 MB of memory is standard, and can be upgraded in 1 MB increments to 4 MB. A socket is provided for a 12 Mhz 80C287 math coprocessor.

The integrated 85-key keyboard has 12 function keys and an embedded numeric keypad, and can duplicate all functions of a standard 101-key IBM keyboard. The detachable VGA display has an 11" (diagonal) viewing area, and features an electroluminescent backlit, supertwist, blue mode LCD with 640x480 resolution.

Standard interfaces include two RS-232 serial ports, and one each of the following ports: parallel, external VGA video display, external enhanced keyboard, and external mouse. Also provided are one 104-pin PC/AT expansion bus port, one RJ-11C telephone line jack, and one RJ-11C telephone set jack.

The strikingly distinctive modular design boasts a high-grade black magnesium case and one of the thinnest displays on the market. In fact, its profile with the display (closed) is only 2.2 inches in height. The power efficient disk drives are shock-mounted to withstand the rigors of travel. The internal design continues the high-tech theme with its VLSI and surface mount technology.

Included as standard with the **Dynabook** are the MS-DOS 4.01 operating system, SmartStart system setup software, Dynabook utilities, and on-line documentation.

Options

Battery: The thin-line rechargeable battery is a lead-acid dry type providing two to four hours of life. Standard charge time is 12 hours, and trickle charge time is 24 hours. Its profile is 0.4" x 13.2" x 11" and its weight is 5.4 pounds.

Also available are a nylon or leather carrying case, a second universal PC adapter, and a 2400 bps modem.

Performance and Limitations

The Dynabook microcomputer delivers desktop power which can rival even 386 desktop machines. Performance testing, representing computer performance relative to an IBM XT, resulted in a Norton rating (Peter Norton's Computing Index) of 15.4, which compares favorably with such portables as the Compaq SLT/286 (9.8), Zenith SupersPort 286 (11.7), the Compaq SLT/286 (9.8), and the Toshiba T1600 and T3200 (11.7). Even the Compaq Deskpro 386s Desktop falls short at 13.2.

The Dynabook's optional thin-line battery provides two to four hours of desktop computing power, after which time a recharge is required. Recharge rate is 12 hours (standard charge) or 24 hours (trickle charge). An automatic energy saver program reserves battery power when the system is not in use, and an indicator light signals when power is running low. Data communications require only an ordinary phone line.

Support

SCI/Fortune's Dynabook product line is provided with a one year warranty.

Tips from Technical Support

The following are some questions and answers from SCI/Fortune's technical support team:

Question: We can automatically attach a glossary to a document to be edited by using wp2 documentname glossaryname. Can a glossary be automatically attached to a **Fortune:Word** *session in some way?*

Answer: Not in a simple straightforward way, but the following procedure will work. First create a keystroke file (for example, /usr/wp2/gl-in) with the following contents:

^Am^MaglXXXXX^M

where XXXXX is the name of the glossary you wish to attach. The ^A is a CTRL-A and the ^M is a CTRL-M. The easiest way to create this file is just by using the **cat** command.

cat > /usr/bin/wp2/gl-in
^Am^MaglXXXXX^M
CTRL-D

Because the CTRL characters are special characters, you will have to enter a CTRL-V before typing them. (If you use the screen editor to make the file, enter a CTRL-C before entering the ^A and ^M.)
Then make a copy of /bin/wp2 on FOR:PRO or /usr/bin/wp2 on System V, calling it /usr/bin/wpgl. Edit the copy replacing the line that "exec"s "wp" with the following two lines:

stty raw -echo
exec sh -c "cat /usr/wp2/gl-in /dev/tty | wp"

When executing wpgl, the glossary will be attached before control is passed to the keyboard.

Note: When exiting **Fortune:Word** invoked in this way under FOR:PRO, a prompt will not display until a key is pressed and a message about a broken pipe has been displayed. When exiting **Fortune:Word** invoked this way under 386/ix, the keyboard will seem locked -- press the LF/GL key to clear anything you have already typed, and then, though nothing displays, type stty sane and press the LF/GL key again. If your keyboard doesn't have a line feed key (LF/GL), use ^J instead -- that is, hold down the CTRL key and press j.

/u/help

Fortune:Word Tips and Questions

Our question: Will the new Hewlett Packard Laserjet III work with Fortune systems, and specifically with Fortune:Word?

Answer: Robert Dick, of Toronto Canada, wrote to let us know that in fact the latest printer from HP works very well with his Fortune systems. The new Series III printer has been hailed across the industry as a major improvement over the Series II, sporting better print quality and scalable (many sized) fonts. The following is the text of his letter to us:

As promised, here's the disk containing all of the needed files to use either a Laserjet III or IIP (HP's personal laser printer at 4 ppm) with Fortune:Word. The disk contains a modified printcap file as well as two new wheels: hpIII.whl and hpIIp.whl.

The wheel for the Laserjet II (hpIII.whl) allows any **Fortune:Word** user to take advantage of the built-in CG Times (Times Roman) and Univers (Helvetica fonts through the Laser Interface. The wheels 1b, 2b, 3b, 4b and 5b gain access to the CG Times and Univers fonts. The CTRL-y videotext codes work with the 2b wheel in the same fashion as they do with Fortune's 1p wheel. The fonts used are slightly large than those on HP's "B" cartridge (11.5 points versus 10 points) so documents formatted for the "B" cartridge will need to be reformatted. However, I think the CG Times font is more pleasing to the eye in the 11.5 point size.

The wheel for the IIP (hpIIp.whl) works with HP's "WordPerfect" font cartridge and the IIP or the Series II. As with the wheel file for the Laserjet III, CG Times and Univers fonts are accessed through wheels 1b, 2b, 3b, 4b, and 5b. The CG Times fonts in this case are sized at 12 points and, again, documents formatted for the "B" cartridge will have to be reformatted.

Installation is simple. Rename the /etc/printcap file to /etc/printcap.old then copy the printcap file from /f to /etc. The wheels files should be copied to /usr/lib/wheels. Once the files are installed you need to go to S2, 39 on the Global Menu and change the TYPE of printer to HPIII for Laserjet III or HPIIP for a Laserjet IIP or Series II using the "WordPerfect" font cartridge.

I've also included a wheel for the HP DeskJet which support the built-in Courier fonts in four sizes as well as Times Roman fonts on the "22706P" cartridge. It's printer type is DESKJET. We have used it here and it seems to work properly.

Regards,

Robert J. Dick

I've had some further conversations with Robert Dick about how the wheels files work with the HP Series III. As we've discussed previously in these pages, with the proportionately spaced fonts, Times and Univers, each character takes up a different amount of space, e.g. an "i" is much thinner than a "W". **Fortune:Word** needs to know how wide each character is, and it gets this information from the appropriate wheels file. HP has a utility program which will print out the character widths for any font. Robert Dick used this program to determine what information should be put in the wheels file he created. In theory, this could be done for a great variety of font sizes. If you are interested in his wheels files, they are available from us on a floppy disk. Please send us $10 for shipping and handling, and we will gladly send you a copy. The HP program should be available from Hewlett Packard or your dealer.

Question: When one of our users goes in to display a short index in Fortune:Word, the file names are sometimes single spaced instead of double spaced, and it's impossible to move the cursor onto the desired file. Do you have any idea what could cause this?

Answer: Well, we had never seen that, but when we checked with Dave Vantine of Vantine Business Systems, he asked if your user was using an FIS1000 (new type) of terminal. Apparently sometimes these terminals get confused and exhibit the problem you're having. To fix the problem, enter the Terminal Setup Mode (CTRL-ALT-HELP), choose the default setup, and save the changes. If you need to, you can go back in and change the setup back to Softset. Even though everything looks okay in the setup, the contents of the setup chip sometimes get scrambled, and need to be re-written.

We have occasionally seen the same problem with the EAROM in a Fortune 32:16.

Classified

This Classified section of **/u/fortune** *news* is designed to serve our subscribers and Fortune computer users in general. In order to help everyone use their Fortune hardware and software to its utmost, we are providing a forum for the selling and the buying of used equipment. We will be including this Classified section in each issue of **/u/fortune** *news* as long as there is sufficient interest. So, if you would like to find something for your Fortune or locate a buyer for your equipment, you should not overlook this valuable resource. If you are interested in a listing in this section of **/u/fortune** *news*, please call us at (617) 894-6900.

FOR SALE: 32:16 w/5 Terms

30 MB hard disk, QUME letter quality printer, 12 serial ports, 1 console plus 5 terminals, Fortune:Word, BASIC

No Reasonable offer refused
Call Ted at 203 725-4000 4/90

Fortune 32:16 XP30

1 MB Ram
4 port controller
CRT and keybaord
For:Pro 2.0
Extended Fortune:Word 3.0
Fortune-to-Fortune Copy 1.2
Call **Larry Zoeller** (312) 558-4575 4/90

Fortune XP 32:16 with 4 terminals

1 Meg of RAM, 70 MB hard disk, with Fortune:Word, Multiplan, Business Basic, all documentation.

Call Don at (617) 267-6500 for more information. 11/89

WANTED -- 32:16 Software for Job Shop Management

of production of single or lot quantities to customer's specifications, estimating based on past history, scheduling of labor and machines, tracking production and promised "Due Dates", material purchasing, accurate job costing and accounting features.
Please call Dave at 419 294-1987 2/90

Fortune 32:16 XP 70

1 Meg Memory
13-70MB Hard Disks
Ports: 12 serial & 1 Parallel
60 MB Tape Expansion
Fortune Basic Workstation

Will take Best Offer
Call Rena at (301) 907-8900 10/89

Fortune 32:16 -- 1 meg RAM, 50 meg

Console + 1 Terminal
1 Meg Memory (4x256)
50 meg Maxtor Drive
C and Pascal compilers
Fortune:Word 2.0
Extra WE Power supplies

Call Chuck at 203 937-5063 2/90

32:16, 6 terminals, 2 printers!!!

Hardware: 70 MB with 30 meg/20 meg tape exp cab 2 MB RAM, 2 4-port CommA's. **NEC 7710 & 3500 printers.**

Software: For:Pro 2.0, Fort:Wrd 3.0, Document Conversion, Handshake ITE & COLT, Kermit
Contact: Susan (203) 972-3400 2/90

Fortunes For Sale - Priced Right!

Two - 32:16's (20 & 30 MB HD)
One - SX70T (2MB memory)

For:Pro 2.1, Multiplan, Ext. Fortune:Word BASIC 6.5.12, IDOL, BAS, Dev. Utils., Communications, other software. Call **Russ** or leave message at (708) 510-0088. 1/90

Fully Loaded Fortune XP-45

If you have been looking for the perfect expansion system, this is it.

Start with a beefy 45Mb hard drive, and 1 Mb of RAM, expandable to 2 Mb. Of course, there are 2 four-port Comm A boards, plus **Fortune:Link LAN with the companion board for your other system.** System console is an amber FIS 1000 terminal and keyboard.

For:Pro level is 2.0. Fortune:Windows, Fortune:Word 3.0, Multiplan, ITE, SST's Handshake Colt ,Informix and Development Utilities are all included.

This is one whale of a computer.

Call Mike Eisen at (619) 323-4555. 2/90

Used Fortune Parts For Sale

WD06
CRT Electronics and Tube
256KB Memory Board
800KB Floppy Drive
Comm A 4 Port Board
Video Controller
1 MB XP Mother Board
10 MB Hard Disk
20 MB Hard Disk (C20)
and more...

All components are fully functional, supplies are limited.

For more information, call **Jim** at **(203) 828-4811** 12/89

Fortune 32:16 XP

Hardware: 45 MB Hard Disk, 2 MB RAM, Terminal and keyboard, 4 ports

Software: For:Pro V2.1, Rubix

All Cables and Books included
Price: Best Offer
Call: Brad Hunter at (313) 296-7130 1/90

Fortune 32:16 System

30MB hard disk with 256KB, 800K flexible drive, 4 ports, console terminal plus 2 Fortune Intelligent Work Stations (12" monochrome display). Software: Fortune:Word, Multiplan and Sequitur, plus Hayes Smart Modem, 300 baud. Form more information, call Laurie at (602) 795-8870. 11/89

20MB 32:16 - Includes For:Pro 1.7.4.1, 768K RAM, console, 2 terminals, 4-port Comm A and a **20MB Tape Expansion Box.** Software includes: SMC BAS modules, Extended Fortune:Word, Developtment Utilities, and a C compiler. Purchased direct from Fortune, Hardly used, will take best offer. Call Gerry Armel at (508) 285-3040. 9/89

Now your Fortune can talk to a PC!!!

With our fxfer, floppy transfer program, you can read and write a Fortune floppy disk with an IBM PC or compatible.

With Fortune:Word<-->Word Perfect, Two of the best word processors can exchange documents preserving margins, tabs, indents, bold, underline, etc. (Fortune:Word to Microsoft Word on its way next.)

They said it couldn't be done, but we did it, and you can benefit.

Call the folks at /u/fortune news for more information:
(617) 894-6900.

A message to our readers:

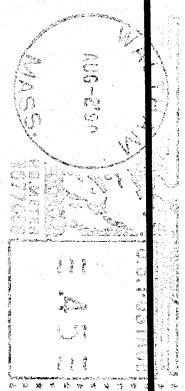
SCI/Fortune has announced two new products this month -- the 88000 based RISC computer that can run over 50 users, and a 80286 based laptop that will keep you going on the road. Details of the systems are in the "News from Fortune" article. We're excited about the new products and many things that we have to share with you. We've brought you some exciting news this month, and look forward to some exciting articles next month.

Coming soon:

ISome thoughts on the pros and cons of upgrading your system
Using a database program to print to a postscript printer
Faster ways to backup with tape
Using a modem on 32:16's and 5000's
Exporting data from a BASIC file

/u/fortune news
225 Crescent Street
Waltham, MA 02154

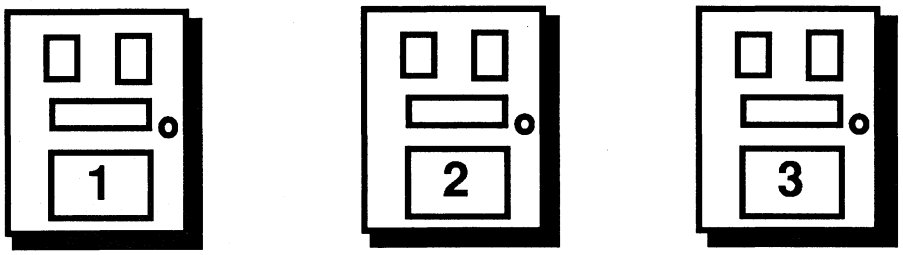
If you're reading a xeroxed copy of **/u/fortune *news***, don't. Copying this publication is prohibited. We depend subscription revenue to continue publishing **/u/fortune *news***. Pleas play fair. There aren't many of us out there and we have to stick together.



/u/fortune* *news*

The Newsletter for Users of SCI/Fortune Computers June 1990 / Volume 7 Number 5

Time to Upgrade?



"You can have what's behind Door #1, Door #2, or Door #3, or maybe you'd just prefer what I have here in this black box?"

/u/fortune *news*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz

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, Inc. are affiliated with SCI/Fortune Systems.

Subscriptions cost $60 for one year within the USA. For foreign subscriptions, add $36 each year for added postage and handling costs.

Back issues are offered at:

Volume 1, Numbers 1 - 8	$2/each - $14/set
Volume 2, Numbers 1 - 7	$3/each - $18/set
Volume 3, Numbers 1 - 12	$4/each - $44/set
Volume 4, Numbers 1 - 12	$4/each - $44/set
Volume 5, Numbers 1 - 12	$5/each - $55/set
Volume 6, Numbers 1 -	$5/each

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $15 each, quantity discounts available. 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 the contents of **/u/fortune *news***.

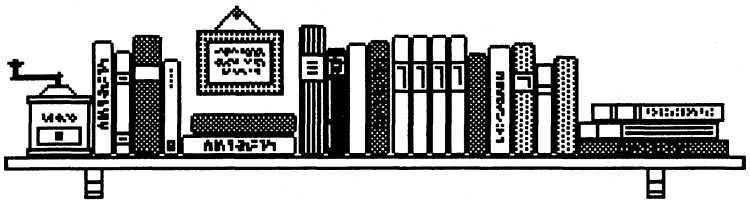
For more information, call or write:

The Cambridge Consortium, Inc.
225 Crescent Street
Waltham, MA 02154
(617) 894-6900

The text for this newsletter was prepared using an SCI/Fortune 5000 for both text input and typesetting. The typesetting was accomplished by using the Aldus Pagemaker program under VP/ix, which is a DOS emulation.

CONTENTS

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.

August 13, 1990

This month we continue with our collection of odds and ends called "How I Spent My Summer Vacation". Our standard format will pick up again once the dog days have passed.

Summertime is a time to reflect on the many spring projects, work on the left over, less critical projects, relax in the sun, and, my favorite activity, read the mail.

Although mail arrives daily from the post office, from various postal services and by fax, we tend to open immediately only those envelopes which appear to have checks enclosed. The rest we save up for a few days and read at our leisure. Here is a typical sampling of our weekly mail.

Dear Thoroughbred Business Partner:

Version 4.0.1 of SOLUTION-IV Accounting Software for General Ledger, Accounts Payable, and Accounts Receivable is now available. This latest release includes such additional features as: faster program startup after menu selection; file information report showing critical files; and the ability to disable sales/purchase analysis summary changes on an existing customer/vendor; date ranges on AP Monthly Check Register allowing you to show checks for multiple months, vendor ranges on AP Monthly Disbursements Journal and AP Monthly Purchases Journal, and many other enhancements.

This new version requires IDOL-IV Version 4.3.2. For more information, please contact your Thoroughbred Account Representative.

Sincerely, Concept Omega Corporation

* * * * * *

Dear Mr. Murn'l:

I checked with my local manager and have received authorization to offer you and Ms. Crawford a full dinner each time you exceed your yearly systems sales quota. Please let me know when you are ready to become an IBM Business Partner.

Sincerely, Isaac Byron Moneymaker

* * * * * *

Dear Ray:

I thought you might be interested in a strange error 2 I have encountered twice in the last couple of weeks. It happens when entering the BAS applications on the 32:16 multiuser systems, apparently after some kind of a system crash.

I found that the control file CCNVZ has been damaged in such a way that the "number of records" indicator in the file ID has been corrupted. Even though there are plenty of free records, when the new operator is logged into CCNVZ the error 2 (full file) appears.

The fix is very simple. Define a new file with the same parameters as CCNVZ and write a little program to READ RECORD from CCNVZ and WRITE RECORD to the new file for all existing CCNVZ records. Then erase CCNVZ and rename the new file.

It is possible the "File Transfer And Expand" utility will also work, but I have not tried this.

Sincerely, Mike Eisen, Beacon Systems

P.S. I loved your two act play! How about a sequel?

* * * * * *

Dear Business Executive:

Having learned of your interest in computerizing your business, I am delighted to invite you to attend a major seminar targeted specifically to companies considering, or already engaged in, using small computers for their company accounting.

"Taking the Bull by the Horns: A Hands-On Approach" will be held on Tuesday, October 10 at the Airport Hotel. This program is unique in that it will focus solely on practical approaches to purchasing computers.

Do you know the difference between a "bit" and a "byte"? What is "memory"? "Storage"? "Software"? If you have been agonizing over these complicated computer terms, you will not want to miss this important seminar.

You will be receiving more information next month. In the interim, please contact

me if you have any questions regarding the seminar or would like to register in advance. The registration fee is $1,495.00 which includes a luncheon, breaks and extensive briefing materials.

I look forward to seeing you on October 10.

Sincerely, Chairman of the Board, Computers for Business

"Serving Baltimore since 1988!"

* * * * * *

Attn: Ray or Lynne:

I attempted to provide Concept Omega with the enclosed information but evidently my address for them is an old one. You may be interested in the bug in Payroll I've run into.

Level G Thoroughbred IDOL Payroll

Problem: If an employee draws vacation pay along with his regular weekly salary (with overtime) on the pay period in which his SDI obligation terminates (that is, the gross year-to-date will exceed the established maximum base) more than the maximum deduction will be taken.

Reason: Vacation pay deductions are computed separately to avoid excessive income withholding which would occur if the vacation gross should be added to the other pay categories. Therefore, SDI is lodged against both without determining that their sum would over extract the maximum.

Solution: Since such an occurrence will be seldom and because the overage amount is small, no revision is suggested. If the employee complains, make a petty cash adjustment.

Sincerely, C. L. Meserve, Riverside Marina

* * * * * *

Dear Ray:

That printer gag was great! You MUST send me the documentation on how you did that!

I loved your play about IBM. Please come up with more!

Sincerely, Frank

* * * * * *

Dear Ray:

Contrary to rumors which I understand have been circulating, SCI/Fortune has NOT recently destroyed all of the BAS Accounting software materials. In most cases you can order new BAS packages directly from your representative at SCI/Fortune.

I do not know how these rumors started, but be assured that we plan to continue providing for 32:16 users.

Yours truly, An Insider

* * * * * *

Dear Thoroughbred Business Partner:

Keep an eye on your mailbox as very shortly you will be receiving an announcement of the availability of Thoroughbred's SOLUTION-IV Accounting Inventory Control, Order Processing, and Payroll. The product announcement will provide full details on the availability of these new modules though software is available today on a limited basis.

Do you need an inventory system which is capable of forecasting future usage and recommending both reorder points and order quantities (EOQ) based on prior usage trends using one of the latest available forecasting techniques?

How about tracking outstanding quotes and orders, as well as return material authorizations (RMA's) with the ability to turn an outstanding quote into an order as well as both single step invoicing and two-step order processing?

Would you like a Payroll system that is fully parameter driven with absolutely no "hard coded" tax table calculations?

If so, you may want to look at SOLUTION-IV Accounting, Thoroughbred's full line of 4GL Accounting software written completely in IDOL-IV.

Best regards, Concept Omega Corporation

* * * * * *

Dear Sir or Madam:

I am newly graduated from computer trade school and would like much sales position in your fine organisation. Enclosed please to find resume for my vitals.

Truly, Szbch Czrk

* * * * * *

System Administration: Part 37

No Longer requires a Degree in UNIX Wizardry

This series is a must for all owners and users of the SCI/Fortune computer. Dave is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the SCI/Fortune computer, and is past president of the Houston UNIX User's Group. He contributes independently to **/u/fortune *news***.

Creating Friendly Menus on a 5000 *by Dave Kloes*

In Part 36, we completed our discussion of selection 2 from our "**kmenu**" shell program. In addition, we discussed some backup strategies and talked about the "**tar**" command. Remember that the "**kmenu**" shell script is an example of a user friendly front end program that can be used for SCO Xenix. We have also been presenting some fundamentals of shell programming. Much of the material is also pertinent to the Fortune 32:16. If you are interested in learning more about shell programming, you can order our book "**The New Shell Game**". Refer to our ad in this publication for our phone number and address.

Let's continue our discussion of the "**kmenu**" program we gave you in Part 34:

```
41            3) clear
42               echo "
43    Backup of entire system ...
44
45    Insert tape and depress <RETURN> to continue: \c"
46               read bunk
47               if backup 0ufk /dev/rctmini 39000 /dev/root
48                  then echo "
49    Backup successful."
50               else echo "
51    Backup not successful."
52               fi
53               echo "^GDepress <RETURN> to continue: \c"
54               read bunk;;
```

Selection "3" on our menu is "Weekly (Friday) Backup (full system backup - **dump**)". Remember last time that we suggested using a "**tar**" backup on Monday thru Thursday and a "**dump**" backup on Friday. This selection performs a complete image backup of the "root" partition or filesystem.

```
41          3) clear
```

The "clear" command in line 41 clears the screen.

```
42               echo "
43    Backup of entire system ...
44
45    Insert tape and depress <RETURN> to continue: \c"
```

The "echo" command tells the user what kind of backup is being performed and asks the user to insert a tape.

```
46          read bunk
```

The "read" command pauses for user input (in this case, the <RETURN> key). Whatever they enter is put into the variable "bunk".

```
47               if backup 0ufk /dev/rctmini 39000 /dev/root
48                  then echo "
49    Backup successful."
50               else echo "
51    Backup not successful."
52               fi
```

Lines 47 thru 52 actually do the backup. The "**backup**" command in SCO Xenix actually calls the "**dump**" command. There is one modification to the original script that we recommend. If you are using a full size cartridge tape, SCO Xenix has the "tape" command that can be used to rewind, retension and/or check the status of the tape cartridge. We have found that in some cases the tape may not be rewound. In these cases, your "backup" command would get an error message. If you want to play it safe, modify our original script as follows to add the "**tape rewind**" command:

```
            if echo "Rewinding tape ..."
               tape rewind
               backup 0ufk /dev/rctmini 39000 /dev/root
               then echo "
Backup successful."
            else echo "
Backup not successful."
            fi
```

You might want to modify selection "2" to include the "tape rewind" before doing the "**tar**" backup as well. Here are some tidbits about the "backup" or "**dump**" command:

1. This command does an image backup of a particular partition. If you have multiple partitions, you need to do a "**dump**" of each partition to backup the entire system. Normally, however, we suggest doing a "**dump**" of only the root partition and a "**tar**" of the other partitions. The purpose of the **dump** is to quickly restore the "root" filesystem when your hard disk is replaced or the "root" filesystem is corrupted.

2. One disadvantage of the "**dump**" is that you cannot restore individual directories. Your options are to restore the complete backup or individual files. This is the reason that we suggest the "**tar**" backup of other data partitions. **tar** allows you to restore by directory name.

3. The system should not be in use while the backup is in progress.

```
backup 0ufk /dev/rctmini 39000 /dev/root
       ^
       |
```

4. The "0" in line 47 indicates a full partition backup. If you have a large partition, you might want to do "incremental" backups using 1-9. An incremental backup only backs up files that either been added or changed since the last backup. We prefer to do the full backup. This makes it easier to restore the system when needed.

```
backup 0ufk /dev/rctmini 39000 /dev/root
        ^
        |
```

5. The "u" in line 47 tells "**dump**" to write the date and time of the backup at the beginning of the tape. The following command would be used to display the date/time:

```
dump tf /dev/rctmini
```

Remember that the device name you use with this command or the command on line 47 may be different than "/dev/rctmini". This device name is the default name if your tape unit uses mini size cartridge tapes. The device name "/dev/rct0" is used for full size cartridge tapes for tape units. This is the normal tape unit you will have on your Fortune 5000 system.

```
backup 0ufk /dev/rctmini 39000 /dev/root
         ^       ^
         |-------|
```

6. The "f" on line 47 tell "backup" that the next argument will be the device name for our tape unit - in this case /dev/rct0.

```
backup 0ufk /dev/rctmini 39000 /dev/root
          ^               ^
          |---------------|
```

7. The "k" on line 47 tells backup that we are going to specify the capacity of the tape in kilobytes. We would use "150000" if we had a 150MB tape unit.

```
backup 0ufk /dev/rctmini 39000 /dev/root
                                    ^
                                    |
```

8. The "/dev/root" on line 47 specifies the partition that is to be backed up. This is the default name for the root partition

for SCO Xenix.

9. To look at the files from a "**dump**" tape, use the following command:

```
dump Tf /dev/rctmini
```

If the backup is successful, then the "echo" command on line 48 is executed otherwise we get the error message produced by line 50.

```
53      echo "^GDepress <RETURN> to continue: \c"
 54         read bunk;;
```

Finally, when the backup is complete, the user is asked to depress the <RETURN> key and this selection is complete. This is put here so that the user has the opportunity to see the "successful" or "not successful" message.

```
55      4) clear
56           /etc/shutdown;;
```

Selection 4 (Shutdown system) on our menu is used to shutdown the system. This executes lines 55 and 56 of "**kmenu**". In Part 34, we suggested adding a user called "admin" which is used to access the menu we have been discussing. Whenever you use the "shutdown" command, the program "/etc/shutdown" is executed. Actually, the "shutdown" command is a shell program. If you look at this script, you will see that it checks to see if you are logged in as "root" or "sysadm" before it lets you shutdown the system. This is a problem if we are logged in as "admin". However, since we are now becoming smart system administrators, we realize that we can modify this script to also allow our "admin" user to use the "shutdown" utility. If you want "admin" to be able to shutdown the system, make the following change to the "/etc/shutdown" script:

Now reads:

```
case $1 in
root|sysadm|"") : OK            <----------
```

Change to:

root|sysadm|admin|"") : OK

```
57      5) /usr/bin/kprofile;;
```

Selection 5 (Link New User Into the Menu System) is used after a new user has been added to the system using "mkuser" or "sysadmsh". Once the user has been added, this selection will run the "kprofile" program to copy a default ".profile" file into the user home directory so that "**kmenu**" will be displayed when they login. Here is the "kprofile" script:

Script name: kprofile

```
------------------------------------------------------------
 1    ext="n"
 2    while test "$ext" = "n"
 3    do clear
 4       echo -n "
 5    This program will link a user who has just been
 6    added into the system into the menu system.
 7
 8    Enter name of user: "
 9       read name
10       if grep "$name:" /etc/passwd
11          then cp /usr/bin/kproto /u/$name/.profile
12               echo "
13    User account $name is now ready to be used.
14    Depress <RETURN> to continue: "
15               read bunk
16               ext="y"
17       else echo -n "^G
18    The user name you have entered cannot be found.
19    Be sure you have added the user using 'System Administration'.
20    Depress <RETURN> to continue: "
21            read bunk
22       fi
23    done
------------------------------------------------------------
```

Remember that this is another program that is called from within our "**kmenu**" script. Let's take a look at what it is doing:

```
1   ext="n"
```

Line 1 sets the variable "ext" to the value "n". We are going to use this variable to determine when to exit this program.

A "while" loop continues to execute as long as a certain condition exists. The keywords for the loop are "while", "do", and "done". These can be found on lines 2, 3, and 23.

```
2   while test "$ext" = "n"
```

Basically, this line "tests" to see if the value of "ext" is equal

to "n". Since it is (we set the value in line 1), then the while loop will continue to process until the value of this variable is some value other than "n". The "do" list specifies the actions to be taken. All actions from the "do" in line 3 to the "done" in line 23 are taken.

```
3   do clear
```

The first action is to clear the screen.

```
4       echo -n "
5    This program will link a user who has just been
6    added into the system into the menu system.
7
8    Enter name of user: "
```

Next, the "echo" line reminds the user what the program does and prompts for a valid user login name. Here we enter the name of the user we just added to the system.

```
9   read name
```

The "read" command pauses for the user to enter the name and puts the name that is entered into the variable "name".

```
10      if grep "$name:" /etc/passwd
11         then cat /usr/bin/kproto >> /u/$name/.profile
12              echo "
13   User account $name is now ready to be used.
14   Depress <RETURN> to continue: "
15              read bunk
16              ext="y"
17      else echo -n "^G
18   The user name you have entered cannot be found.
19   Be sure you have added the user using 'System Administration'.
20   Depress <RETURN> to continue: "
21           read bunk
22      fi
```

One use of the "if" statement is to test for whether a command is successful or not. If the "grep" command in line 10 is successful then the actions from line 11 to line 16 are executed. The "grep" command is being used here to see if the name the user entered is present in the /etc/passwd file. If you have entered a valid name, then the test is successful and the lines are executed. If the name is not found in the passwd file (the test is NOT successful) then lines 17 to 21 are executed. Remember that any "if" statement must be ended with an "fi".

The "kproto" file is actually the little front end script that we showed you in Part 34 that is added to a user's .profile file to execute the menu system. Again, here is what it looks like:

File: kproto

```
1  # This is appended to the .profile file
2  trap '/usr/bin/kmenu' 1 2 15
3  /usr/bin/kmenu
4  clear
5  kill -9 0
```

Please refer back to Part 34 for an explanation of the file. In essence, the "cat" command in line 11 appends these lines to the end of the user's .profile file which in turn will execute "**kmenu**" when the user logs in. Note that our system defaults user directories to the "/u" directory. If you system adds user accounts to some other directory, you need to modify this line of the program to reflect the correct directory on your system.

In the next issue, we will continue our discussion of shell scripts and "**kmenu**". In the meantime, we hope you are enjoying the summer weather.❑

UNIX Basics

Some ways to automate system backup

In this article I want to describe a system for backing up our Fortune 5386 computer. As users and administrators of computer systems know, it is absolutely essential that a backup system be used to provide protection for your files.

Perhaps you've never thought about in this manner before, but the files that reside on your hard disk(s) are probably the most non-replaceable part of your computer. Every other piece of hardware and all commercially available software packages can be replaced in a few days. Your files and modifications to software packages if lost, may never be replaceable or are replaceable only after extensive work - all work that you've done before and that's not efficient!

On the 5386 that I use we have many files that are created and worked on over the course of several days to several months. Thus, if we were to lose any of them, it would take an equal amount of time to redo the work. In addition, we do alot of specialized computer programming in both C and using shellscripts. Some of these software programs have been developed over the course of two years - can you imagine the predicament we would be in if we lost any of these programs and couldn't replace them via a backup?

So, in order to ensure that our software investment is maximally protected, we have instituted a fairly comprehensive backup strategy that has proven successful. In addition, one important feature of the strategy I will describe is that it is very easy to use. In fact, other than putting a tape in the computer each morning, the entire process is automatic!

Perhaps you've never thought about in this manner before, but the files that reside on your hard disk(s) are probably the most non-replaceable part of your computer. Every other piece of hardware and all commercially available software packages can be replaced in a few days.

The computer that I use is a Fortune 5386 with 2 70MB hard disks and a 150MB tape unit. The fact that our entire hard disk capacity is less than the capacity on the tape drive allows us to design a comprehensive and easy-to-use backup system. The system includes daily, weekly and monthly backups and each of these backups are "full" backups - that is, every file on each of the two disks are backed up for every backup. In some sense, then, there is actually no difference between daily, weekly and monthly backups except for how we store and save them.

The backups take place automatically (via cron - but more on that later) beginning at 11PM each weekday night (Monday thru Friday). Backups that occur on Monday, Tuesday, Wednesday and Thursday are called daily backups. We keep four tapes for these daily backups labeled with the day of the week (e.g., Monday, Tuesday, etc.) and reuse them each week. Thus, the Monday tape will not be reused for a week.

Please note that there is quite a bit of redundancy built into this system primarily because we are able to do complete backups each day. The redundancy can be understood by imagining that if I lost a file on Wednesday, I would only have to look back on the Tuesday tape to locate it. Theoretically, I should never have to look back more than one day. This raises the potential question of why keep reusing these daily tapes?

There is a two-fold answer to this question. First, my experience has taught me that no single backup is ever fool-proof. Sometimes the system will crash before the backup occurs,

Figure 1 -- nfullback Automatic Backup with cpio

```
 1 :
 2 #
 3 # backup up entire system using cpio
 4 # write to cpioback.log
 5 #
 6
 7 # set the ulimit to a larger size to accomodate really large files
 8 ulimit 500000
 9
10 DOW=`date '+%w'`
11 DOM=`date '+%d'`
12 DOM=`expr $DOM - 1`
13 MON=`date '+%b'`
14 MOY=`date '+%m'`
15 case $MON in
16       Jan|Mar|Jul|Aug|Oct|Dec) ndays=31;;
17       Apr|Jun|Sep|Nov) ndays=30;;
18       Feb) ndays=28;;
19       *) ndays=30;;
20 esac
21
22 case $DOW in
23       1|2|3|4) FN="D${DOW}.Z";;
24       5) # it's friday, figure out if its a weekly or monthly
25                 ndaysleft=`expr $ndays - $DOM`
26                 if [ $ndaysleft -gt 6 ]
27                 then # this is a weekly
28                           WNUM=`expr $DOM BBBBBBBBBc" >> /usr/logs/
cpioback.log
38 find . -depth -print | cpio -ovC51200 > /dev/rmt0 2> /usr/logs/fred
39 echo " Stop: `date`" >> /usr/logs/cpioback.log
40 echo "Tail of Actual Backup" >> /usr/logs/cpioback.log
41 tail -5 /usr/logs/fred >> /usr/logs/cpioback.log
42 echo "Tail of Backup listing" >> /usr/logs/cpioback.log
43 cpio -itv -C51200 < /dev/rmt0 > /usr/logs/table.list 2> /dev/null
44 tail -4 /usr/logs/table.list >> /usr/logs/cpioback.log
45 #
46 # Now copy a compressed version of table.list to /usr/logs/lists
47 #
48 /usr/cmd/compress /usr/logs/table.list
49 /bin/mv /usr/logs/table.list.Z /usr/logs/lists/$FN
```

or the tape becomes or is defective in some manner or even that the backup program could have a bug in it (you see, I modify it from time to time and I sometimes introduce bugs that take me a while to locate). Thus, having the ability to go back more than one day could possibly avert a computer disaster.

Second, the tape media will eventually go bad. Thus, by having four tapes, I prolong the inevitable. In fact, included in my regimen is graduating my daily tapes to weekly tapes and my weekly tapes to monthly tapes so that I always have my freshest tapes as daily backups.

Ok, I've just described the daily backups which occur on Monday thru Thursday. The backup that occurs on Friday is a weekly backup with the exception of the last Friday of the month which becomes a monthly backup. Again, for the weekly backups I keep four different tapes, one for Week 1, one for Week 2 and so on up to Week 4 and these tapes get reused on a monthly basis. On the last Friday of the month the monthly backup occurs. This tape gets put into storage and is never reused.

In the remainder of this article I want to describe a shellscript that takes care of performing these different backups. First I will describe some of its characteristics that I feel are important.

1) It is automatic. Its automatic nature is due to the fact that it is a shellscript which is started by the cron program each night at 11PM. In order to have cron do this for me, I had to add the following line to the appropriate crontab file:

```
0 23 * * 1-5 /usr/cmd/nfullback
```

2) It automatically knows whether to do a daily, weekly or monthly backup. The shellscript logic for this is described below.

3) It keeps a log file of the backup activity. Each night the shellscript adds to this log file some important information including when the backup began, when it finished and a consistency check of the backup. This consistency check is described below.

4) A verbose list of all the files backed up are kept in a special directory and this list is automatically compressed to save space. Keeping lists of this sort aid in locating a file to restore.

5) The backup can handle really big files thru the use of the ulimit command.

6) The backup runs about as fast as possible on the 5386 by utilizing the cpio command.

The Shellscript: nfullback

The actual shellscript that I use is shown in Figure 1. I call the shellscript nfullback and keep it in the /usr/cmd directory. In addition, I have set up two special directories that the shellscript uses. First, /usr/logs is where the backup log is kept (along with other log files that I keep). Second, /usr/logs/lists which is where I keep the listings of the files that were backed up.

Now that we have the appropriate directories in order, we can describe how the script works.

Lines 1 thru 5 are just some comments. Line 7 is also a comment that describes the effect of Line 8. Line 8 is the ulimit command that enables this script to backup really large files. On the 5386, the ulimit is normally set to 4096 which represents the number of blocks that can be written to a file. I added this line to the shellscript when I discovered that a very large file on my system was not being properly backed up. The ulimit of 500000 is obviously overkill, but since it costs me nothing, why not?

Lines 10 thru 14 parse the date command so that I can work with individual units of the output. As a reminder, the output of the date command looks like this:

Mon Aug 6 10:13:18 EDT 1990

However, the date command on the 5386 (and a version so on. This is accomplished by dividing the Day of the Month (DOM) by 7 and then adding 1.

If the number of days left (ndaysleft) is less than or equal to 6, then we conclude that this is a monthly backup and name our backup listing accordingly as specified in Line 32. The monthly backups get named something like 01 for January, 02 for February and so on.

So, up to this point all we have done is figure out what to call the backup listing.

Line 36 changes the directory so that we will start the backup from the root directory thus ensuring that we in fact perform a complete backup of the entire system.

Line 37 writes the beginning date and time to the log file called cpioback.log.

This backup strategy has some obvious strong points including that it is easy to use and is virtually automatic.

Line 38 does the actual backup. I use the find command to send all the files to the cpio command which writes the backup out to the tape device (/dev/rmt0). A listing of the files written is stored in the temporary file called /usr/logs/fred. The actual cpio command is rather simple but please note the sequence "C51200" - this tells the cpio command to buffer its

output using a buffer of 51200 bytes and the effect of this is to significantly speed up the backup. There will be an article that explores the various ways of using the tape backup unit and will show you the fastest possible commands.

Line 39 writes the ending date and time to the log file. Then Lines 40 and 41 write the tail of the listing file to the log file. I do this as the beginning of a consistency check.

Lines 42 to 44 write some more information to the log file and then instruct cpio to read through the backup tape to make a listing of what is on the tape (Line 43). The tail of this listing is also written to the log file. Thus, I can look at the log file and make sure that these two tail listings match exactly. If they don't match, then I know that something was wrong with my backup and I had better do it again.

As a note, the only problems I have had revolve around (1) when I forget to put a tape in the tape drive (happens about once a month) and (2) when there is something wrong with the media. This has not happened recently but several months ago when I was using the tar program instead of cpio and I had a brand new tape, I would sometimes get an error. However, since switching to cpio, new tapes don't seem to be a problem.

Lines 48 and 49 take care compressing the listing file and moving it to the /usr/logs/lists directory.

Final Thoughts

This backup strategy has some obvious strong points including that it is easy to use and is virtually automatic. Its simplicity comes from the fact that the amount of disk space I back up is less than the tape drive capacity. Not every system is configured in this way and it is perhaps just as likely that you have a system where the hard disk capacity is greater than the tape drive capacity. In a situation like this, modifications to the shellscript as presented would be necessary.

Perhaps the most obvious change would be to formulate "incremental" daily and weekly backups. The advantage to this would be that you could still keep the backups on one tape so that the backup could run unattended. However, the drawback to this is that restoring a file (or worse, an entire directory) would be much more complicated and might require restoring from multiple tapes.

Another way to cut down on the amount of information to backup is to not backup the relatively stable portions of the system except on monthly backups. In fact, with a judicious selection of which directories to back up and by sending the backup through compress first, you could even backup a system with 300 MB on one tape. Again, the main point is that if you only need one tape, then your life is greatly simplified because you can schedule the backups to run at some convenient time when no one else is using the system.

In any case, backups are important and nothing can substitute for a regularly scheduled regime of backups. The only sure way of preserving the most important part of your computer - its files - is by backing them up. Don't neglect this important area of system administration.❑

Mark Palmerino

Classified

This Classified section of **/u/fortune** *news* is designed to serve our subscribers and Fortune computer users in general. In order to help everyone use their Fortune hardware and software to its utmost, we are providing a forum for the selling and the buying of used equipment. We will be including this Classified section in each issue of **/u/fortune** *news* as long as there is sufficient interest. So, if you would like to find something for your Fortune or locate a buyer for your equipment, you should not overlook this valuable resource. If you are interested in a listing in this section of **/u/fortune** *news*, please call us at (617) 894-6900.

FOR SALE: 32:16 w/5 Terms

30 MB hard disk, QUME letter quality printer, 12 serial ports, 1 console plus 5 terminals, Fortune:Word, BASIC

No Reasonable offer refused
Call Ted at 203 725-4000
4/90

Fortune 32:16 XP30
1 MB Ram
4 port controller
CRT and keybaord
For:Pro 2.0
Extended Fortune:Word 3.0
Fortune-to-Fortune Copy 1.2
Call **Larry Zoeller** (312) 558-4575 4/90

WANTED -- 32:16 Software for Job Shop Management
of production of single or lot quantities to customer's specifications, estimating based on past history, scheduling of labor and machines, tracking production and promised "Due Dates", material purchasing, accurate job costing and accounting features.
Please call Dave at 419 294-1987 2/90

/u/help!

4GL Comparison
Caution on Interactive Updates -- Console Freeze-up

*Question: I would like to have a copy of the full report of your article in **/u/fortune news** Volume 7, Number 3 by Jeffrey L. Zickler: "Comparing IDOL-IV to other 4 GL's". Would you be so kind as to send me one?*

*Also, I need your help. I've installed **Fortune:Word** on a AT386 and only have a console to use it. There is the file **"mapkey"** that takes care of the various function keys - but if I invoke **"wp2"** with the mapkey file, I have to reset my AT386 because the console doesn't accept any keystokes anymore. So I call **"wp2"** without **"mapkey"**, but as a consequence I can't delete letters in a text and cannot use the **"mode"** key.*

*What do I have to do to be able to run **Fortune:Word** on a console.*

Thank you for your help,
yours sincerely,

Heinrich Boeker, Vienna, Austria

Answer: We don't have a copy of the report by Jeffrey Zickler. However, it can be obtained by contacting Mr. Zickler at:

Advance Ware
1371C Warner Avenue
Tustin, CA 92680
714 259-1761

I believe the charge for it is $5.00.

As to the "mapkey" problem, we put the question to Dave Vantine of Vantine Business Systems. He had experienced the same thing. The source of the trouble is probably an update to the Interactive 2.0.2 operating system, probably either the X6 or X8 update. This update changes the mapping of the international keyboard. The **"mapkey"** program is not compatible with the new keyboard mappings that have been included with the update. Lars Hultin of **SCI/Fortune** reports that they are working on this problem, and we will let you know as soon as they have a new version. They feel that there are still some issues to be worked out with Interactive's new releases, and so they are waiting to resolve those issues. An update should be available shortly. At this time **Fortune:Word** will only work with version 2.0.2 of 386/ix. It will not work with version 2.2.

Unfortunately the only way to fix your current problem is to reinstall the entire operating system. We do not know of a way to deinstall the X6 update.

There have also been some incompatibility problems with the X5 update which helps with setting up in/out ports. Although it does fix several things, it doesn't work with the TERM communications program, which requires ugetty's rather than getty's.

The moral of these stories is that you should check with your software vendors before installing new releases of your operating system. We will try to keep you informed of problems that we're aware of.

Hooking up to an AS/400

Question: Have any of your readers had any experience connecting a Fortune 5000 to an AS/400? We have a 5386-33 MHz and are trying to connect up to an AS/400 using 3270 Bisync communications with 32 simultaneous sessions. So far, we haven't had any luck finding the right product(s). If you have any information, please call me at 517 334-6060.

John Goergen
Michigan County Prosecutors Office

Answer: We don't have ideas, but we'll spread the word.

Where do you get the Fortune Laptop /Word Era?

*Question: Just got your May issue. 1) How much is the laptop? 2) From whence it may be ordered? 3) I have a 32:16 with **Fortune:Word** - I take it I would need to buy the **Fortune:Word** software or get **Word Perfect** and use your software to transfer files.*

Jim Turner
Bellevue, WA

Answer: 1) As a matter of long-standing policy, we do not print prices for hardware because different vendors package their services differently. You should check with several dealers for the price and service you require. 2) The Dynabook Laptop is available from any **SCI/Fortune** dealer. If you don't have a relationship with one, you can call any of the advertisers in our newsletter, or call **SCI/Fortune** at the 800 number on page 2 for a dealer in your area. 3) If you'd like to use **Fortune:Word** with the Dynabook, which typically will use the DOS operating system, you can purchase **Word Era** from Wang Laboratories (or they will recommend a dealer). They can be contacted at 415 594-9981. You'll need 3 megabytes of disk space to install the program. Dave Vantine reports that he has clients who use Word Era on their DOS machines and **Fortune:Word** on their Fortune's, and transfer files back and forth with no problem.

If you prefer, you could use our program to convert your **Fortune:Word** files to **Word Perfect**.❑

Time to Upgrade?

Some thoughts on your options

(In last year's June edition of **/u/fortune news**, Dave Kloes discussed many of the issues involved in trying to decide whether to upgrade your computer system. In the past year, there have been many new developments in hardware, and even in the past month or so **SCI/Fortune** has introduced several new computers. This article further explores upgrade options. In future issues, we will be printing the viewpoints of various dealers and users. If you would like to tell us about any upgrades that you have made, and what has worked well, or what has been a disappointment, please call or write to us at the address on the back cover. Our phone number is 617 894-6900.)

Like many of our readers, we still use our trusty 32:16 to process most of the information that floats around The Cambridge Consortium. We have our accounting system on it, as well as our subscriber database. Of course it's grown some since we originally bought it back in 1983 with the help of some bank loans and even a few personal credit cards. It's now an SX70T with a couple of extra hard disks, a NEC laser printer for all of our correspondence, a dot-matrix printer for printing labels, another for printing invoices, and an older QUME letter-quality just for the heck of it. We also have three terminals on it, most of them running at least 4-5 open windows with **Fortune:Windows**. At times things can get a little slow.

So the question is, is it time to upgrade, and if it is, where do we go from here? We're going to try to explore that question here, and hopefully some of our thoughts will be useful to you.

The primary reasons most users will upgrade are system age, lack of desired performance, and compatibility with other software or hardware. As you weigh these decisions, always choose your software first and make sure that it will run on the hardware that you are buying. Let's look at some of these possibilites.

Everything is fine -- but is my Fortune getting old?

Let's say that you only have a couple of people working on your Fortune, mostly doing **Fortune:Word**, maybe some accounting, and occasional **Multiplan** work. The computer works as fast as you can think and/or type, and you're perfectly satisfied with everything. That situation describes quite a few Fortune owners.

It is true that your computer is aging, and the probability of failures does increase with age. Because Fortune 32:16/SX and Formula computers are proprietary, their parts are more expensive to replace than more standardized (read PC compatible) systems. The makes repairs and/or maintenance contracts more costly. However, there is also a market for used 32:16/SX computers, and it may be cost effective to purchase another system as a spare or for parts.

Most parts are interchangeable between machines, with the most notable exception being the security PAL which is necessary to run protected software such as **Fortune:Word**. If it is necessary to change the motherboards between two systems, it will also be necessary to move your old PAL onto the new motherboard.

Buying a used computer may also help you increase the total memory in your computer. Generally one megabyte is preferable for most work, and two megabytes are even better. Used memory boards are quite affordable, although some of the older 32:16's will only accommodate one megabyte of RAM. If you're buying a used computer, try to get one with a Western Electric power supply (a hollow round dot is on the switch) and a WD-06 hard disk controller (which has two small "daughter boards", separate printed circuit cards attached to the main board. The hard disk controller is just to the right of the floppy disk and runs from front to back in a 32:16/SX.) These are the current revision levels for these two parts, and are highly recommended for the larger disk drives.

The hard disks in all Fortune's are completely industry standard ST506 MFM drives. (However, some are

The primary reasons most users will upgrade are system age, lack of desired performance, and compatibility with other software or hardware. As you weigh these decisions, always choose your software first and make sure that it will run on the hardware that you are buying.

specially modifed at **SCI/Fortune**.) It's best to get disks of the makes and models directly supported by Fortune, but as long as you keep to those guidelines, upgrading or replacing hard disks is feasible.

We suspect that with a few spare parts, it won't be too difficult to keep your 32:16 going for several more years. However, you may still want to upgrade to reap benefits of improved speed and greater software flexibility.

My system is steadfast, but too slow for my needs.

There are several options to improve speed -- in fact there are so many that this is where many users have difficulty making decisions. Within the Fortune product line, there are several options:

- upgrade current system to SX or Formula, add more memory/disk
- 286/386SX/386 microprocessor, running DOS, single user only
- 386 microprocessor, running UNIX
- 386 microprocessor, running a network, such as Novell
- 88000 microprocessor, running UNIX

Each of these has its pros and cons.

Upgrade current system

As mentioned above, adding used memory or a new hard disk should be relatively cost effective. A 70-megabyte disk that cost $2,000 a couple of years ago should be well under $1,000 now. If you want to do the work yourself, it could be less than $600. With the standard hard disk controller, the 32:16/SX computers are limited to three 70-megabyte drives. Larger drives will not work.

If you find that your computer frequently pauses for a moment before you can get its attention, it is likely that it is "swapping", which means that portions of the electronic memory (RAM) have been copied over to the hard disk. Before it can do your work, it has to transfer that information back off of the hard disk into its working memory. This often takes a few seconds because hard disks are mechanical and much slower than electronic RAM chips. Increasing the amount of memory in you computer will help prevent swapping by providing more RAM workspace. The **vmstat** command indicates how much RAM is free and how much is swapped, and is a good indicator of a swapping problem.

Ultimately, you may be able to improve your performance by 25-75%. However, in our situation with three terminals all running **Fortune:Windows** (which is a great program), the system is often a little pokey.

The upgrade to a Formula 4000 or 8000 is really more of a change than an upgrade, because very few of the parts in the 32:16 line will work in a Formula. There is a substantial difference in performance though. Either of the Formula models will support many more users than a 32:16/SX.

Switch to a 286/386 computer running DOS

If you only need one person to have access to your information, this is a viable solution. For most functions, you will find that a single user 286 machine will run much faster than your multiuser Fortune. Even if it doesn't run faster, it seems much faster because the video display is written to directly, rather than a character at a time as is done on a terminal. Also, since the machine is only running a single process, it has much less to think about and thus can be faster.

You can also get a very nice color display that will be more fun to look at that the green or amber terminal you are used to. It's not clear that color leads to greater productivity, although many new programs use the many color options to differentiate between different parts of the screen. This is particularly helpful in database and spreadsheet programs.

In addition, you will have a large selection of software to choose from. It will be necessary to purchase new copies of all of your software, because none of what you have on the Fortune will run on a DOS machine, and **SCI/Fortune** doesn't sell any DOS software. If you would like to continue using a word processor like **Fortune:Word**, I believe it is available as **Word Era** from WANG, who recently bought the software from Tigera Corp (which got it from Fortune). Our products can help you convert **Fortune:Word** documents from your Fortune to a DOS machine. **Multiplan** is available under DOS, and actually is much more powerful than the Fortune version which is fairly old at this point. Microsoft's Windows spreadsheet, **Excel** is also available. Either of these products will directly import SYLK files that you create with the **Multiplan** program on your Fortune. If you want to go to Lotus, you'll have to import to **Multiplan** or **Excel** first, because **123** can't read a SYLK file directly.

Industry conventional wisdom would actually recommend getting at least the 386SX microprocessor in order to run many of the new programs that are due out. The 386SX manages memory in much more efficient ways, and is much more flexible than the older 286 processor. **SCI/Fortune** just released one of the fastest 386SX machines available -- running at 20 MHz. (see sidebar on microprocessors and speed). Programs like **Microsoft Windows** and desktop publishing programs such as **Pagemaker** or **Ventura Publisher** will run faster on the 386 machines.

If you are really a power user, running intensive graphics programs or number crunchers, you may want to go with a true 386 machine. These will provide somewhat better performance than the 386SX versions.

386 microprocessor running UNIX

This was the first new hardware option that **SCI/Fortune** introduced with its **Fortune 5000** line. Since that original 16 MHz offering, they have added 20, 25, and 33 MHz options. The 386 chip runs UNIX quite well. For 1-10 users, you will see a very significant performance difference vs. a 32:16 or SX. The comparison to a Formula 4000 or 8000 is more subtle, with each system having its own strengths.

The Fortune 5000 that we have here is

a 16 MHz running **Interactive 386/ix**. The reason that we don't use it for most of our business operation is that we haven't made the commitment to upgrade all of our software. The **Progress** database that we run would have to be converted, and we can't afford that at the moment. **Fortune:Word** is available directly from **SCI/Fortune** for the very reasonable price of $90 as an upgrade to your existing **Fortune:Word**. Some users are also using the UNIX version of **Word Perfect** which is has wider use among DOS users. We still feel that **Fortune:Word** is one of the easiest word processors to use, especially for untrained staff, and that is a considerable benefit.

The performance of the 5386 at 16 MHz is impressive. We have hooked up 6 terminals, all running a Progress database, and had performance that was at least twice as fast as a 32:16 SX computer. **SCI/Fortune's** current low end 386 now runs at 20 MHz, and they go all the way up to the 33 MHz cache model, which must be quite fast.

In addition to the greater speed of the 5386 line, you get the benefit of a much greater selection of hardware and software. Disks of 300 - 700 megabytes are quite common these days. Tape backup systems hold from 150-1000 megabytes and are cost effective so that you can relieve your guilt about never backing up. There are several UNIX based spreadsheet programs, although we have yet to find one that is a clear winner. **SCI/Fortune** has found that the **Tactition** program runs well, but we have found some glitches with it still.

Database programs run very well on the 386 machine running UNIX. Since memory is affordable these days, it is reasonable to load up the machine with 8 or more megabytes of memory. With a fast I/O card for the terminals, it should be possible to run somewhere around 20 users with the 33 MHz machines. With lighter use, it is probably feasible to run even more users.

Running DOS under VP/ix

As we have reported in the past, it is possible to run most DOS software as a process under UNIX using the VP/ix program. Almost any program will run from the system console, which is usually a graphics terminal of some sort. Other DOS programs will run from the other terminals, but typically only character-based software will work. Because of the keyboard differences between most DOS programs and terminals (there isn't any ALT key on most terminals), it is often a hassle to run DOS programs on terminals. Two products that make it easier to do this are the **UnTerminal** which makes it possible to add up to 7 screens and keyboards, and **Sun River** boards, which user fiber-optic cables to attach graphics workstations to a 386 computer.

However, if you are primarily running DOS applications, we wouldn't recommend running them all on separate terminals under UNIX. There is considerable overhead with VP/ix, and it will have a significant impact on overall system performance. In addition, although DOS performance is acceptable, it is not equal to a stand-alone 286 running at 12 MHz. We consider VP/ix to be a major convenience, but not a substitute for a DOS machine.

386 running Novell Network

There is an age old (at least microcomputer age) debate between the respective strengths and weaknesses of networks vs. multiuser systems. If more than one person uses your Fortune, you already know the virtues of having the same data available to several people. We have many clients who have between 2 and 20 DOS based computers which are completely separated from each other. When they want to share information, they use the "sneaker-net" method, which means they walk over with a floppy disk. We wish there was an easy method to hook them all up, but there isn't a single solution we can recommend for everyone.

The main difference between a network and a multiuser system is that in a network, each user has their own CPU (microprocessor) which does most of the actual computing for them -- the network is used only to move information between the different computers.

A Microprocessor Rundown

What is all this talk about 286, 386, 68000, etc. These numbers all refer to different microprocessors that are the heart and brains of any computer system. The microprocessor performs all of the calculations that are necessary to run a computer. Each one is designed by a team of people, and has its own "machine" language that is used to give it instructions. Perhaps the most helpful way to explain the differences between the various chips is to give a short history.

Not too long ago, most microcomputers ran on a Z80 microprocessor. This chip was greatly limited in its ability to use memory and move data in and out. In the early eighties, several new CPU's were developed. One was an 8088 processor developed by Intel corporation. This was the chip that was used to create the first IBM PC computers. At the same time Motorola created the 68000 chip. This is the chip that is used in the Fortune 32:16. It is also the chip that was chosen for Apple's Macintosh computer. Most people feel that the 8088 chip was chosen by IBM because it was inexpensive. Those same people feel that it is limited, and that the 68000 chip is a much more powerful chip. As one quick example, the 68000 chip allowed a multiprocessing operating system like UNIX to work even back in 1983.

Since that time, both Intel and Motorola have developed newer and more powerful versions of their chips. Intel has created the 80286, 80386SX, 80386 and 80486 chips. The 80286 was the original chip in the IBM AT computer. It would run programs about 6 times faster than the original PC. The newer 80386 may run 20 to 30 times faster, and the 80486 faster still. The newer chips can handle more information more quickly, and often also integrate functions of several other chips into their works. Motorola has countered Intel's development with the 68020, 68030, and 68040.

Rundown , cont'd

SCI/Fortune Formula 4000/8000 use the 68020 processor. Unlike Intel's implementation, the program code for the Motorola chips is slightly different, making new software necessary for the Formula computers.

In the Intel/DOS world, an attempt has been made to maintain downward compatibility so that programs that run on the 80386 will also run on an 8088. Recently this has changed because newer programs are often too demanding for the lowly 8088. Improved memory management may even require an 80386. The 80386SX is a compromise chip that offers complete compatibility with the full 80386, but with a reduced capacity for processing information. It has become very popular because it is much less expensive than a full 80386.

SCI/Fortune has chosen not to develop further products with the Motorola 68XXX line of microprocessors -- instead they are now shipping the 5880 RISC computer which is based on the much more powerful Motorola 88000 chip. This processor uses a Reduced Instruction Set which allows it to operate more efficiently than standard CPU's. This is a new technique that was invented a couple of years ago. IBM originally marketed a RISC computer, and has now introduced the 6000 line, which is based on their own RISC chip. Likewise, DEC, Sun and several other vendors make RISC based machines.

Clock speeds make a difference too

In addition to the differences in raw computing power of various microprocessors, each CPU can typically be run at different speeds. The faster the "clock" rate, the faster the computer runs. The original IBM PC ran at 4.77 MHz. The 32:16 runs at 6 MHz. AT computers with the 80286 processor started at 6 MHz, and are now available up to 20 MHz. The Fortune SX machine was faster than the 32:16 because it had a 12 MHz clock. (Don't confuse the SX in Fortune's machine with the 80386SX -- they are not related.)

The newer 80386 machines run up to 33 MHz, and the RISC machines from **SCI/Fortune** run at 16, 20 or 25 MHz. In general, these numbers should only be used for comparisons within a certain microprocessor group. In addition, the same chip running twice as fast will not necessarily make your computer twice as fast for all applications. Other factors such as hard disk speed, memory speed, Input/Output, etc. all have an impact.

Caching is the final advantage

As processors became faster, the ability to access memory rapidly became a bottleneck. Rather than install very fast and very expensive RAM chips into a computer, many vendors found that it was more economical to create a small area of memory that was much faster than the main memory. When the computer needed information, it could be put into this holding tank, while the information that was just kind of hanging around could be left in the much larger tank of main RAM, which is slower and less expensive. The cache lets the computer move information in and out even more quickly. However, if the information it needs is not stored in the small cache space, it has to go out and get more information from the main RAM memory. Thus the larger the cache, the greater the likelihood that it will contain whatever the computer wants next. **SCI/Fortune** has cache memory available on their 5386/33.

In a multiuser computer, there is usually one main microprocessor, and the terminals don't do much of anything besides making requests to the CPU, and displaying the results.

The general consensus is that networks make more sense for offices that are doing very demanding work on each workstation, and multiuser systems are better for less CPU intensive work and sharing databases. For instance, a desktop publishing service bureau would be much better served by several 386 computers running Pagemaker on their local computer, and just using a network to share common files like clip art, and sharing a printer. Another office that processes insurance claims or order entry, and is doing a lot of keyboarding into a common database or accounting system is much better served by a multiuser computer. Unfortunately, most choices are not so clear cut.

One factor is how much it will cost to make the switch. If you already have an office full of PC's (IBM compatible computers), then the cost of adding a network may be primarily adding a file server machine, the network cards, and a the software and installation cost. If you already have Fortune terminals for everyone in the office, you would need to purchase PC's for everyone to install a network, and so a multiuser solution might make more sense.

Another issue is system administration. If you already know UNIX, then you have an advantage over many other computer users. For:Pro is a version of UNIX, although Fortune did a very good job in insulating end users from the UNIX language itself by designing the Global Menu system. Although System V UNIX has its differences from For:Pro, if you're even familiar with the ls, cp and mv commands, you'll have a good start.

Novell networks require different expertise for setting up new users, creating directory trees, login profiles, etc. The hardware part of a network is also more complex, because data is typically sent in packets rather than as a simple ASCII stream. Network hardware options include Ethernet, Arcnet, Token-ring, etc., and each has its own set of issues.

Multiuser systems are much easier to wire and maintain.

Finally, there is the issue of your vendor. If your vendor is more familiar with network operation, they will probably steer you towards networks as the best solution. Similarly, if they know multiuser systems better, that is probably what they will recommend. If you are unsure about a decision, get quotes from several people, and references from installed sites that are similar to yours. Be sure to check and see what the projected cost per user will be for expansion, and how many users can theoretically work on the system. As is always the case, make sure the software you want to use will run well on the hardware.

Creating a network of UNIX computers

One option that some people have pursued that hasn't been mentioned here is using UNIX to network together several 386 computers, each with a number of terminals attached. This can preserve the use of an operating system that you are familiar with, while spreading the computing load out some. Ethernet can be used to share file systems, and is offered by several different vendors.

88000 RISC processor running UNIX

RISC processors are among the fastest processors made today. The Motorola 88000 CPU is an extremely fast RISC microprocessor, and is the heart of the new 5880 computer offered by **SCI/Fortune**. The new IBM 6000 computer is also based on a RISC CPU, as are Sun Sparcstations, the new Dec's etc..

The technical details of how the 5880 uses the RISC processor are beyond the scope of this article, but in brief, the 5880 uses the 88000 for all of the main computing, while a 80386 or 80286 is used for terminal Input and Output (I/O). The memory for the 88000 is located on the same CPU board as the 88000, and is very fast.

This architecture creates a machine which runs at 19/23 VAX MIPS (16/20 MHz), and can comfortably support 50+ users in most situations. (In comparison, an IBM POWERstation 320 runs at 27.5 VAX MIPS, a Sun SPARCstation1 runs 12.6 VAX MIPS, and an Everex Step486is (with an 80486 CPU) runs at 10.2 VAX MIPS) The 5880 runs System V.3 UNIX, which is the same version as Interactive's 386/UNIX and SCO's UNIX/XENIX. However, the 5880 needs software to be ported specifically to the 88000 chip. There are several companies that have already ported their software, including **SCI/Fortune** with **Fortune:Word** and **Fortune:Windows**. Any Binary Compliant software should run on **SCI/Fortune's** 88000 machine. (**Fortune:Word** and **Fortune:Windows** are not certified to run on other hardware).

There is an organization (88open Consortium, Ltd.) that has been established by the industry to help companies port their software to the 88000 chip, and so there should be a even more software available for it in the near future. It will run DOS applications, but only on the system console, and it should be noted that DOS use at the console can affect performance system-wide.

If you have a need for more than 20 people using your computer at once, it would be wise to consider the 5880. It provides a lot of performance for the price.

In Conclusion

The right decision for each reader obviously depends on the usual sale criteria -- benefits and costs. Upgrading will probably result in lower maintenance costs, but will require expenditures for the upgrade. Most routes will provide much better performance, even if the processor is a 386 running at 16 MHz.

At the core of any hardware decision should be a software choice. What software do you want to run, and will it run on the hardware you're looking at? For instance, although it is possible to run Business Basic on many of the new systems, you may find that your Fortune terminals may not be functional for Business Basic. A particular database program you depend on may not be available for all potential hardware platforms. Or in certain cases, the price of the software may vary with the power of the hardware. For instance, a database program may cost $3,000 on the Fortune 32:16, $3,500 on a 386 based computer, and $8,000 on the RISC based 5880. The theory is that you are supporting many more users with a much more powerful machine.

Once you have chosen your software, you need to determine your budget. Prices for all computers have been coming down in the past year, primarily because of increased integration (a computer that may have required 60 chips 4 years ago may now only need 4) and falling memory prices. Because of the low cost of memory, we encourage most users to get as much as they can reasonably afford. More memory allows users to run programs like **Fortune:Windows** with greater ease.

If you decide to continue to have multiuser capabilities, you will be paying $1,000-$2,000 more for a multiuser operating system like SCO UNIX/XENIX or Interactive's 386/ix. You'll probably also pay more for any software that you use on the system. However, you'll be gaining the ability to share information between workers, which we think is indispensable.

Finally, see if you can get a reference from someone who is in a similar situation as yours. We have spoken with many people who are thrilled with their new 386 based systems.

Good luck.❑

Josh Lobel

Fortune 5000 Product Roadmap

	Dynabook	5286/10	5286/16	5386SX/16	5386SX/20	5386/20	5386/25	5386/25 Cache	5386/33 Cache	5880 RISC
Processor/Speed	80286/16 MHz	80286/10 MHz	80286/16 MHz	80386SX/16 MHz	80386SX/20 MHz	80386/20 MHz	80386/25 MHz	80386/25 MHz	80386/33 MHz	88100/16, 20 MHz
Bus Slots	1 104-pin PC/AT	8-ISA	6-ISA, 2-XT	8-ISA	8-ISA	7-ISA	7-ISA	8-ISA	8-ISA	7-ISA
Serial/Parallel	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	26/1
Implementation	VLSI	Zymos	Zymos	VLSI	VLSI	C&T	C&T	C&T	C&T	C&T
On-Board Controllers	Floppy & Hard Disk	Dual Floppy[3]	Dual Floppy[3] IDE Hard Disk	Dual Floppy[3] IDE Hard Disk	Dual Floppy[3] IDE Hard Disk	Dual Floppy[3]	Dual Floppy[3]	Dual Floppy[3]	Dual Floppy[3]	Dual Floppy[3,5]
Floppy Drive	3.5"	5.25" or 3.5"[1]	5.25" or 3.5"[1]	5.25" or 3.5"[1]	5.25" or 3.5"[1]	5.25" or 3.5"[1]	5.25" or 3.5"[1]	5.25" or 3.5"[1]	5.25" or 3.5"[1]	5.25"[4]
Hard Drive	40 MB	See Below[1]	See Below[1]	See Below[1]	See Below[1]	See Below[1]	See Below[1]	See Below[1]	See Below[1]	ESDI Only[1]
Memory: **System Capacity** **System Board -** **32-bit Board -** **Cache Size -**	 4 MB 1 MB[4] to 4 MB – 	 16 MB 1 MB[4] – –	 16 MB 1 MB[4] – –	 16 MB 1 MB to 4 MB – –	 16 MB 1 MB to 4 MB – –	 16 MB – 2 MB to 16 MB –	 16 MB – 2 MB to 16 MB –	 16 MB 1 MB to 8 MB – 32 KB Cache	 16 MB 1 MB to 8 MB – 32 KB Cache	 20 MB 8 MB to 20 MB – 32 KB Cache
Coprocessor[1]	80C287-12[1]	80287-10[1]	80287-16[1]	80387SX-16[1]	80387SX-20[1]	80387-20[1]	80387-25[1]	80387-25[1]	80387-33[1]	Integrated
Video Adapters	VGA[4]	EGA[4] VGA 640x480[2] SVGA 800x600[2]	Mono/CGA/Herc EGA VGA 640x480[2] SVGA 800x600[2]	Mono/CGA/Herc EGA VGA 640x480[2] SVGA 800x600[2]	Mono/CGA/Herc EGA VGA 640x480[2] SVGA 800x600[2]	Mono/CGA/Herc EGA VGA 640x480[2] SVGA 800x600[2]	Mono/CGA/Herc EGA VGA 640x480[2] SVGA 800x600[2]	Mono/CGA/Herc EGA VGA 640x480[2] SVGA 800x600[2]	Mono/CGA/Herc EGA VGA 640x480[2] SVGA 800x600[2]	Mono/CGA/Herc[4]
Video Monitors[1]	VGA LCD[4]	Mono White, SVGA	Mono White, SVGA	Mono White, SVGA	Mono White, SVGA	Mono White, SVGA	Mono White, SVGA	Mono White, SVGA	Mono White, SVGA	Mono White
Tape Drive	–	QIC 150[1]	QIC 150[1]	QIC 150[1]	QIC 150[1]	QIC 150[1]	QIC 150[1]	QIC 150[1]	QIC 150[1]	QIC 150[4]
Operating Systems Supported	MS-DOS[4] SCO XENIX	MS-DOS, SCO XENIX	MS-DOS, SCO XENIX	MS-DOS, MS-OS/2, ISC UNIX, SCO XENIX, SCO UNIX	MS-DOS, MS-OS/2, ISC UNIX, SCO XENIX, SCO UNIX	MS-DOS, MS-OS/2, ISC UNIX, SCO XENIX, SCO UNIX	MS-DOS, MS-OS/2, ISC UNIX, SCO XENIX, SCO UNIX	MS-DOS, MS-OS/2, ISC UNIX, SCO XENIX, SCO UNIX	MS-DOS, MS-OS/2, ISC UNIX, SCO XENIX, SCO UNIX	UNIX System V.3.2[4] (88open Compliant)
Cabinet Design	Portable	3-Bay Desktop	3-Bay Desktop	3-Bay Desktop	3-Bay Desktop	5-Bay Tower	5-Bay Tower	5-Bay Tower	5-Bay Tower	5-Bay Tower

[1]Optional, [2]16-bit, [3]Dual floppy controller supports 5.25" and/or 3.5" drives, [4]Included, [5]Tape & ESDI controller included.

Hard Disk Drives, Unformatted Capacity:

767 MB ESDI, F/H	53 MB ST-506, H/H
382 MB ESDI, F/H	43 MB IDE, H/H
182 MB ESDI, H/H	24 MB ST-506, H/H
100 MB ST-506, H/H	

SCI/Fortune, 2000 Ringwood Avenue, San Jose, CA 95131
800-443-7072

SCI FORTUNE TECHNOLOGY

/u/fortune news
225 Crescent Street
Waltham, MA 02154

If you're reading a xeroxed copy of **/u/fortune *news***, don't. Copying this publication is prohibited. We depend on subscription revenue to continue publishing **/u/fortune *news***. Please play fair. There aren't many of us out there and we have to stick together.

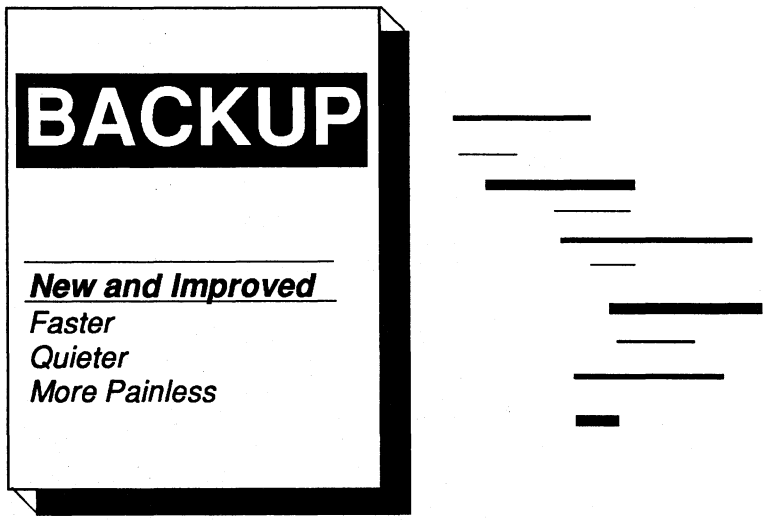
/u/fortune* *news*

The Newsletter for Users of SCI/Fortune Computers

July 1990 / Volume 7 Number 6

Backing up on a 5000 can be much quicker

Turn to page 10 to find out how you may be able to cut backup time by 80%!



Upgrade to Fortune's new superfast 5880 computer and you'll get Fortune:Word and Fortune:Windows *FREE!!!!*

For the next sixty days you won't have to give up the programs you know and love when you move up to the lightning fast Fortune 5880 based on the 88000 RISC processor. And you'll love them even more when you see them run on this TURBOCHARGED machine.

Call for a demonstration TODAY!

Offer expires December 31, 1990

A full range of business and professional systems is available from SCI/Fortune. Combining expert manufacturing with practical third-party products, we supply resellers with the solutions you need today. Call your local SCI/Fortune dealer for assistance in identifying the right solution for you.

For information about our products or a dealer who can help you, please call (800) 443-7072.

SCI FORTUNE
TECHNOLOGY

/u/fortune *news*

Editors
Josh Lobel
Mark Palmerino

Associate Editor
Benzion Chanowitz

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, Inc. are affiliated with SCI/Fortune Systems.

Subscriptions cost $60 for one year within the USA. For foreign subscriptions, add $36 each year for added postage and handling costs.

Back issues are offered at:

Volume 1, Numbers 1 - 8	$2/each - $14/set
Volume 2, Numbers 1 - 7	$3/each - $18/set
Volume 3, Numbers 1 - 12	$4/each - $44/set
Volume 4, Numbers 1 - 12	$4/each - $44/set
Volume 5, Numbers 1 - 12	$5/each - $55/set
Volume 6, Numbers 1 - 12	$5/each- $55/set

For Back Issues delivered to a foreign address, please add $15 to the order total for added postage and handling. Software diskettes cost $15 each, quantity discounts available. 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 the contents of **/u/fortune *news***.

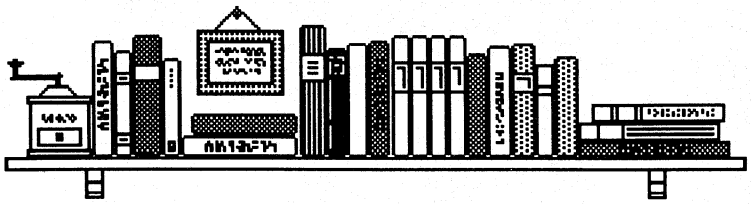
For more information, call or write:

The Cambridge Consortium, Inc.
225 Crescent Street
Waltham, MA 02154
(617) 894-6900

The text for this newsletter was prepared using an SCI/Fortune 5000 for both text input and typesetting. The typesetting was accomplished by using the Aldus Pagemaker program under VP/ix, which is a DOS emulation.

CONTENTS

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.

September 25, 1990

This month we complete our trilogy on "How I Spent My Summer Vacation". We *promise* to get serious again starting next month.

In the fast paced world of computer technology it seems that what was gospel last week is obsolete today. We find ourselves in a constant decision-making mode, feverishly trying to keep up with the so called "state-of-the-art". To do this, we try to make lists of reasons why we should or should not follow certain paths. This summer we had occasion to review several of these lists, and with apologies to "Late Night With David Letterman", we have compiled a few "Top Ten Lists".

THE TOP TEN REASONS WHY YOU SHOULD KEEP YOUR FORTUNE 32:16.

10. You finally got the modem to work with it.
9. You still owe more money on it than it's worth.
8. You have over 1500 documents stored in Fortune:Word.
7. Your programmer knows For:Pro, but not Unix or Xenix.
6. It took seven years to learn where to kick it, why change now?
5. You've given it a name and consider it a family member.
4. It makes a nice boat anchor.
3. Some parts of it still work.
2. You can play "pacman" on it.
1. It counts to nine when you start it up.

THE TOP TEN REASONS TO BECOME AN IBM BUSINESS PARTNER.

10. They will protect your verticals, unless they like someone else better.
9. Their television commercials feature cast members from the old "M.A.S.H." television series.
8. The AIX boxes are "like nothing else on the market".
7. They send you lots and lots of mailers.
5. You can also sell the PS-2's.
6. They are MUCH bigger than SCI/Fortune.
4. Their downtown offices are security protected.
3. They assemble the machines for you.
2. You get to talk to their salespeople.
1. Sooner or later blue will come back into style.

THE TOP TEN REASONS TO SUBSCRIBE TO /u/fortune news.

10. Just to find out who will get married next.
9. It looks impressive lying on your reception room table.
8. It's easier to read than most technical publications.
7. They use honest to goodness photographs as well as neat cartoons.
6. You can get back issues.
5. You can learn where to buy a used 32:16.
4. The color matches your 32:16.
3. They feature two-color mailing labels.
2. Dave Kloes is cuter in person than the wizard pictured in his column.
1. They always show you how many issues you have left on your subscription (including this one).

TOP TEN EXCUSES MOST FREQUENTLY USED BY PROGRAMMERS

10. I forgot to save the final version.
9. It's not finished because I had a day off last week.

8. I'm accustomed to writing new code, not fooling around with maintenance work.
7. You should have told me that before I did all of this work.
6. It can't be done within the limits of the design.
5. I thought you wanted it that way.
4. I'll discuss it with you AFTER my coffee break.
3. I can't think when I have to wear a necktie.
2. I didn't write this junk! I'm just trying to make it work!
1. You're not a programmer, you just don't understand.

TOP TEN EXCUSES MOST FREQUENTLY USED BY SALESPEOPLE

10. Someone else didn't let me handle my own prospects.
9. I couldn't get by to see him because I had to make cold calls that day.
8. I feel that I am over-supervised.
7. They never called me back.
6. It takes TIME to get this vertical going.
5. I have to spend too much time generating my own leads.
4. I'd do better if I wasn't restricted to this dead-end territory.
3. I expect three to close next week.
2. So how would you have handled it?
1. I'd be more motivated if I had a better commission.

TOP TEN REASONS THE SHIPMENT OF YOUR ORDER WAS DELAYED.

10. One of our suppliers is going through a labor dispute.
9. It's in the mail.
8. I didn't know you needed it right away.
7. Give me a break! You just ordered it last month!
6. The U.P.S. truck was involved in a horrible, nine-car collision on Interstate 95. Everything it was carrying was totally destroyed, except your package. As soon as we can get another truck out there we'll get your order to you.
5. We have a new policy. We don't accept orders under $250.00.
4. This is a big company. These things take time.
3. We have plenty of tower models, but desktops are backed up at least six weeks.
2. There is a $100.00 rush fee, otherwise you have to wait like everyone else does.
1. Your purchase order is still on the credit manager's desk.

TOP TEN MOST AGGRAVATING CALLS FOR ASSISTANCE.

10. I'm going to take this *&#!@# computer and flush it down the john!
9. How do I turn this thing on?
8. About this bill you just sent me . . .
7. Do you service home computers?
6. It seems I've accidently erased eight years of sales history.
5. It has been working perfectly for five years, and I didn't do anything different or unusual, and all of a sudden, out of a clear blue sky, for no apparent reason . . .
4. I would have called you earlier, but I didn't want to look stupid.
3. What do you mean, "good morning"! What's "good" about it?
2. What does it mean when the disk partition cannot be found?
1. Do you speak COBOL?

System Administration: Part 38

No Longer requires a Degree in UNIX Wizardry

This series is a must for all owners and users of the SCI/Fortune computer. Dave is President of UNI-KOMP which is located in Houston, Texas. He provides UNIX seminars, software for the SCI/Fortune computer, and is past president of the Houston UNIX User's Group. He contributes independently to **/u/fortune *news***.

Creating Friendly Menus on a 5000 *by Dave Kloes*

For the last few issues, we have been discussing an example of a front end menu shell script that you can use with your system. This has also provided us the opportunity to talk about shell programming commands and concepts in general. In the last issue we worked our way through lines 41-57 of the "kmenu" shell program that was initially presented in Part 34. We also discussed the "backup" and "dump" commands and presented another shell script called "kprofile" that can be used to link a user into the menu system. Let's continue our discussion of "kmenu" by looking at lines 58 and 59:

```
58        6) clear
59          /usr/bin/kcallout;;
```

These two lines are executed if the user selects option 6 from "kmenu" (Callout on the modem). Line 58 clears the screen and line 59 executes another shell program we have written. Actually, we have two versions of "kcallout". The one we will present here is the "simple" version that executes the "cu" command. The other is much more comprehensive and can be used to automatically set a modem port for dialin or dialout at a specified baud rate. In the short version, here is the "cu" command line:

```
Syntax:  cu -s <baud rate> -l <device name> dir

Program:  kcallout

cu -s 2400 -l /dev/tty1a dir
```

The command "cu" allows you to dialout from your modem and log in to another system - it is interactive. Here are the basic steps required to setup a dialout modem:

1. Select the port you want to connect your modem to. Set the modem switches (or send the appropriate commands) for the state you want - dialin or dialout. Connect your cable from the desired port to the modem. We highly recommend a "Smartcable" to eliminate having to make your own cable.

2. Edit the /usr/lib/uucp/Devices (/usr/lib/uucp/L-devices on older versions of Xenix/Unix) to include the modem port:

```
Examples:  Direct tty2A - 2400 direct
           ACU tty2A - 2400 dialHA24
```

3. Be sure all users have read and write permissions for the device being used:

```
chmod 666 /dev/tty2A
```

4. Insure the modem port is **DISABLED** for calling out and **ENABLED** for calling in.

```
disable tty2A or
enable tty2A
```

Also note that device names tty1A and tty1a are two different names for the Com 1 port and tty2A and tty2a are two different names for the Com 2 port. If you are using the modem for dialin on either Com 1 or Com 2, do not enable both names. For example, if tty1a is enabled, tty1A should not be enabled.

There are plenty of traps here, but assuming your cabling and modem switches are correct, you should be able to use "cu".

```
60              7) /usr/bin/kgames;;
```

For those of you that are interested in the SCO Xenix Games, this selection on line 60 will run another shell program called "kgames". Even if you don't use it, it provides an opportunity to look at another shell program.

```
Shell Program:  kgames

    1   ext=n
    2   chg="n"
    3   while test "$ext" = "n"
    4   do clear
    5   echo -n "
    6              Games Menu
    7
    8     1.  backgammon      13. ocelot
```

```
    2.  chase          14. ogre
    3.  fish           15. pig
    4.  fortune        16. reversi
    5.  greed          17. rogue
    6.  hack           18. stars
    7.  hangman        19. trek
    8.  jewell         20. wimp
    9.  life           21. worms
   10.  maze           22. wump
   11.  mind           23. chess
   12.  number         24. Exit Menu

ENTER YOUR SELECTION: __^H^H"
read sel
clear
if test -d /usr/games
   then cd /usr/games
else clear
     echo -n "^G
Games directory /usr/games not found.
Exiting.
Depress <RETURN> to continue: "
     read bunk
     exit
fi
prog=""
case $sel in
     1|23|4|5|6|9|13|14|17|20) instr="yes";;
                            *) instr="no";;
esac
clear
case $sel in
          1) prog=bgm;;
          2) prog=chase;;
          3) prog=fish;;
          4) prog=fortune;;
          5) prog=greed;;
          6) prog=hack;;
          7) prog="hangman -";;
          8) prog=jewell;;
          9) prog=life;;
      10) prog=maze;;
      11) prog=mind;;
      12) prog=number;;
      13) prog=ocelot;;
      14) prog=ogre;;
      15) prog=pig;;
      16) prog=reversi;;
      17) prog=rogue;;
      18) prog=stars;;
      19) prog=trek;;
      20) prog=wimp;;
      21) prog=worms;;
      22) prog=wump;;
      23) prog=chess;;
      24) ext=y
          exit;;
      ^D) ext=y
          exit;;
       *) echo -n "^GNot a valid response! "
esac
if test "$prog" != "" -a "$instr" = "yes"
  then echo -n "
Instructions are available.

BE SURE TO NOTE HOW TO EXIT THE GAME!

Do you want to print instructions? (y/n): "
       read ans
       case $ans in
            y|Y|yes|YES) clear
                         if lp doc/$prog
                           then echo -n "
Instructions sent to printer que.
Depress <RETURN> to continue: "
                                read bunk
                         else echo -n "^G
No instruction file in /usr/games/doc directory.
Depress <RETURN to continue: "
                                read bunk
                         fi;;
                      *) ;;
       esac
fi
if test -f $prog
   then $prog
        echo -n "
Depress <RETURN> to continue "
read bunk
else echo -n "
Could not find game command '$prog' in /usr/games.
Depress <RETURN> to continue: "
     read bunk
fi
done
```

This shell program presents a menu to the user of the available games. The games that are available differ some-

what from Xenix version to Xenix version so we assume you will have to customize this program to suit your needs. As usual, we will go through this program line by line.

Line 1 is a comment line. Line 2 sets the variable "ext" equal to the value "n". This variable will be used to determine when to exit the program. The "while" test in line 3 says to "do" all of the things from line 4 until the "done" in line 103. Line 4 clears the screen.

```
 5   echo -n "
 6              Games Menu
 7
 8    1.  backgammon      13. ocelot
 9    2.  chase           14. ogre
10    3.  fish            15. pig
11    4.  fortune         16. reversi
12    5.  greed           17. rogue
13    6.  hack            18. stars
14    7.  hangman         19. trek
15    8.  jewell          20. wimp
16    9.  life            21. worms
17   10.  maze            22. wump
18   11.  mind            23. chess
19   12.  number          24. Exit Menu
20
21   ENTER YOUR SELECTION: __^H^H"
22   read sel
```

The program uses the "echo" command to paint the menu screen. Note that the starting quote is on line 5 and the ending quote is on line 21. The ^H^H in line 21 does two backspaces so that the cursor is positioned over the first underscore where the user makes their selection. The "read" statement in line 22 pauses for the user to enter a selection and puts whatever they enter into the variable "sel".

```
23   clear
24   if test -d /usr/games
25      then cd /usr/games
26   else clear
27        echo -n "^G
28   Games directory /usr/games not found.
29   Exiting.
30   Depress <RETURN> to continue: "
31        read bunk
32        exit
33   fi
```

The "if" conditional statement starts on line 24 and ends with the "fi" on line 33. The test on line 24 tests to see if the directory "/usr/games" exists. If it does (the test is successful) then the program changes directory to /usr/games. If the directory does not exist (the test is not successful) then the screen clears and the error message on line 28 is displayed. The program waits for the user to depress the return key and exits.

```
34   prog=""
35   case $sel in
36        1|23|4|5|6|9|13|14|17|20) instr="yes";;
37                                  *) instr="no";;
38   esac
```

Line 34 initializes the variable "prog". The "case" statement begins with line 34 and ends with the "esac" on line 38. If the value of the variable "sel" is equal to one of the values listed in line 36, then the variable "instr" is set to "yes", otherwise it is set to "no". The "instr" variable is used here to indicate whether their are instructions for a particular game in the directory /usr/games/doc.

The "case" statement which begins on line 40 and ends on line 69 sets the "prog" variable equal to the name of the game that is to be executed depending on what the user selected. Remember that this value is in the variable "sel". If the user enters something other than the numbers 1 thru 24 or Ctrl-d, then the error message on line 68 is displayed.

```
70   if test "$prog" != "" -a "$instr" = "yes"
71     then echo -n "
72   Instructions are available.
73
74   BE SURE TO NOTE HOW TO EXIT THE GAME!
75
76   Do you want to print instructions? (y/n): "
77          read ans
78          case $ans in
79               y|Y|yes|YES) clear
80                            if lp doc/$prog
81                              then echo -n "
82   Instructions sent to printer que.
83   Depress <RETURN> to continue: "
84                                    read bunk
85                            else echo -n "^G
86   No instruction file in /usr/games/doc directory.
```

```
87    Depress <RETURN to continue: "
88                                    read bunk
89                               fi;;
90                            *) ;;
91          esac
92    fi
```

The "if" conditional statement on line 70 checks for two conditions. In order for the test to be successful, the value of the variable "prog" must not be equal to nothing and the value of the variable "instr" must be equal to "yes". In other words, the user did not just depress the return key and there are instructions for the particular game the user selected. If there are instructions, the user is asked whether they want to print them.

If the user responds with any of the four responses in line 79, then the instructions are printed to the printer using the "lp" command in line 80. If the print command is successful, the user is so notified, otherwise they are give the error message in line 86. For any other answer to the print question, no action is taken as indicated on line 90.

```
93    if test -f $prog
94       then $prog
95            echo -n "
96    Depress <RETURN> to continue "
97    read bunk
98    else echo -n "
99    Could not find game command '$prog' in /usr/games.
100   Depress <RETURN> to continue: "
101        read bunk
102   fi
```

The "if" conditional in line 93 checks to see if the program file exists. If it does, the the program is executed. When the user exits the game, they are asked to depress the return key. If the program name cannot be found, the message in line 99 is displayed.

Based on the logic of the program, there are only two ways for the user to exit the program. First of all, remember that the program exits if there is no /usr/games directory (line 32). Obviously, if there are no games then there is no sense in continuing with our program. They will also exit if they select option 24 from our menu (line 65).

Again, even if you do not have the games on your system, we hope the "kgames" shell script will enhance or reinforce your abilities to write shell programs. For those that are interested in our shell programming or Xenix/Unix seminars, give us a call at (713) 895-9900.

Next time we will once again continue our discussion of "kmenu". We will also discuss the "cu" command in more detail.❑

Classified

This Classified section of **/u/fortune** *news* is designed to serve our subscribers and Fortune computer users in general. In order to help everyone use their Fortune hardware and software to its utmost, we are providing a forum for the selling and the buying of used equipment. We will be including this Classified section in each issue of **/u/fortune** *news* as long as there is sufficient interest. So, if you would like to find something for your Fortune or locate a buyer for your equipment, you should not overlook this valuable resource. If you are interested in a listing in this section of **/u/fortune** *news*, please call us at (617) 894-6900.

FOR SALE: 32:16 w/5 Terms

30 MB hard disk, QUME letter quality printer, 12 serial ports, 1 console plus 5 terminals, Fortune:Word, BASIC

No Reasonable offer refused
Call Ted at 203 725-4000 4/90

Fortune 32:16 XP30
1 MB Ram
4 port controller
CRT and keybaord
For:Pro 2.0
Extended Fortune:Word 3.0
Fortune-to-Fortune Copy 1.2
Call **Larry Zoeller** (312) 558-4575 4/90

WANTED -- 32:16 Software for Job Shop Management
of production of single or lot quantities to customer's specifications, estimating based on past history, scheduling of labor and machines, tracking production and promised "Due Dates", material purchasing, accurate job costing and accounting features.
Please call Dave at 419 294-1987 2/90

UNIX Basics

Cutting backup time by 80%!!!

(David Vantine is President of Vantine Business Systems, Inc. in Waltham, MA. He has been working with Fortune computers since the very beginning. He is contributing to /u/ fortune news independentlly.)

I have been asked on several occasions by Josh to write this article on my test results of backups on the Fortune 5000 running the Interactive 386/ix operating system. It seemed appropriate to write this as a follow up to Dave Kloes' excellent article on backups printed in the May 1990 **/u/ fortune news**.

Those of us who have been supporting the Fortune user community over the years have continuously requested our customers to be sure they have sufficient backups should some major system problem develop, especially a hard disk failure. Without a good backup ones stands to loose all the information that has been stored on the hard disk. We have been lucky that over the past seven years that only one of our customers lost everything and we were unable to recover any of their files. The results needless to say were catastrophic and cost them a severe financial loss. So why is it that many of us, including yours truly, neglect what should be a daily, or at least weekly, routine?

I discovered the tape to stream at a very rapid rate. The time to backup decreased to 3 minutes 27 seconds, or 1.9 megabytes/ minute (five times as fast as the original tar backup).

I think the major issue is that we don't consider the consequences, especially when day to day operations are running smoothly. It is only through a bad experience that we tend to become religious about backup. A second major issue is that backups on the Fortune 32:16 have taken a long time and impact the daily operation of the business. The original 32:16 only had a floppy drive to do backup. This was a very time consuming process to backup 30 megs of data. When the tape cabinet was introduced it helped alleviate the inconvenience of someone constantly having to insert floppy disk after floppy. Although the tape unit was significantly faster than floppy disk backups, it could still take 30-60 minutes to backup a large disk drive.

As the size of PC hard disks increased in size, many manufactures realized the need to incorporate tape backup into their systems. I was astonished at the speed that these units were able to backup. Many could backup 25 megabytes/minute and faster. Why then could our Fortune 5386 only backup at about 1-1.5 megabytes/minute?

Several years ago I began selling a physician's practice management system that ran under Xenix. I was amazed that tape backups using tar were considerably faster than Fortune's, roughly 2.5 megabytes/minute.

I had also sold other hardware that ran System V Unix, and tape backup was equally fast.

When **SCI/Fortune** introduced the 5000 I had to upgrade many of our customers who had stretched the limits of their 32:16's. The first installation I did was for a law office. After the initial conversion problems and new procedures were put into place, things were running fairly smoothly. About nine months after the system was installed, they called one day to say that they had a problem with their Progress database, and asked me to fix it. When I asked when they had last done a backup I was told that it had been several months ago! I was surprised to say the least. However, my first order of business was to correct the situation. Fortunately, dumping and reloading their database solved the problem. I then asked why they were not backing up daily. I was told the backup took too long.

At first I was puzzled because I had expected the backups on the 5000 to be as fast as other System V machines. I decided to run some benchmarks on my 5000.

In order to run my tests, I used the /etc directory as the source files. Before backing up I ran a **du** (disk usage) on /etc and found that the directory contained 13418 blocks (512K blocks), which is roughly 6.7 megabytes. The first command I used to backup was the tar command. The exact command was:

```
tar cvf /dev/tape etc
```

The system took 19 minutes to complete the backup, or .34 megabytes/minute. I then understood why they were neglecting backups of their system with 60 megabytes of data!

I discussed this with Mark Palmerino of The Cambridge Consortium and he confirmed tar's slowness. He had since modified his backups on the 5000 and had created a shell

script called pdtar. His script used tar, but piped the output through to dd. His script syntax was as follows:

```
tar cvf - etc | dd of=/dev/tape bs=5120
```

Basically this is taking the output from the tar command and sending it to dd which converts the files and copies them to the tape device. The bs flag specifies that the input and output should be blocked to 5120 bytes. The time to do the same backup using his script was 11 minutes, 29 seconds, or roughly .58 megabytes/minute. Still not a real speed demon.

I next tried using **cpio** which stands for copy in copy out. The command syntax was:

```
find etc -depth -print | cpio -oBv > /dev/tape
```

What this command is doing is using the find command to print out all of the file names in the etc directory and the "depth" flag tells find to also search any subdirectories under /etc. The output (file list) is passed to cpio, which then outputs the files to the tape device. The o flag specifies cpio to output, the B flag specifies a default block size of 5120, and the v flag is verbose (the file names will be listed on the screen as they are copied). This time the backup was 9 minutes, 29 seconds, about .71 megabytes/minute.

In SCI's Product Release Notes for Interactive 2.02 they stated that "Tape operations that use the cpio command will perform much faster if the cpio output is pied through the dd utility". Using the following command:

```
find etc -depth -print | cpio -ovc | dd of=/dev/tape bs=1024K
```

the time required to back up was 12 minutes 59 seconds, or .5 megabytes/minute.

I then increase the block size to 5120K and found the time to be 12:49, essentially no different. Increasing the block size to 10240K caused the system to run out of memory.

Reading the documentation for cpio, I discovered a flag that I was not familiar with from some other System V operating systems. This was the C flag. As I began experimenting with it I found that the C flag was similar to the B flag, however one could specify the block sizes, rather than using B's default of 5120. Incidentally, without either the B or C flag, cpio's default buffer size is 512 bytes.

Running further tests, I found the optimal size was 51200 with the C option. Using the following command:

```
find etc -depth -print | cpio -ovC51200 > /dev/tape
```

I discovered the tape to stream at a very rapid rate. The time to backup decreased to 3 minutes 27 seconds, or 1.9 megabytes/minute (five times as fast as the original tar backup). Thinking that more ought to be better, I increased the buffer size to 102400. I was surprised to find that the backup only decreased 6 seconds. It was interesting that improvement in speed was very similar to the improvement when I had increased the block size when piping through dd.

Deciding to find out the details of this C option, I called Interactive's technical support. I learned that when using this option there is an additional 0 added to the number that is actually entered.

To summarize, when using cpio without the B or C flags, output is blocked to 512 bytes. The B option blocks to 5120 bytes and C51200 actually blocks to 512000 bytes.

One futher point is that you should notice when I run my backups I don't specify full path names (/etc). I chose to use the relative path name (etc). Either one will work--the difference is that if a full path name is used then *all the files are put back into the /etc directory when they're restored, no matter what the current working directory is.* The user will not have any choice. My preference is to use relative paths so that I have the option to restore a tape to disk without overwriting what is already on the hard disk.

Restoring tapes with relative path names restores the contents *into your current working directory.* Therefore I can make a temporary directory anywhere in the file system and restore the whole tape under that temporary directory. Should I wish to restore it to the /etc directory, I would simply have to be sure my current working directory is the / directory. My reason is simply that I feel this method gives me more options for restoring.

A Final Word of Caution

If you are planning to restore vital files from a tape, it is always prudent to read through the tape to get a listing of what's on it (and hopefully thereby varifying that it's a good tape) before performing the restore. The one client of ours who did lose data did so by restoring a bad tape over their existing files -- wiping out the files on the hard disk, and replacing them with incomplete ones.

I hope this information is helpful, and I would suggest that you experiment without yourself. (A good way to start would be by doing a backup today!!)❑

/u/help!

Dear Mark,

Isn't one of the simultaneously graces and frustrations of Unix that someone is always waiting in the wings to tell you that you could have done something differently. Different strokes! I refer to your article in the May 1990 issue on a shell script handling file names.

The "for ..." has an unusual property in that unless you give it something specific, it assumes that it is supposed to gather a file list from the current directory, thus:

*for file in *.z*

would have the same final results as

for file in `ls *.z`

with the exception that the former would operate faster inasmuch as the global expansion process on the command line would pass the list as arguments to the "for" in lieu of passing them to the "ls" which would in turn pass to the "for." Thus a process is bypassed, and time is saved.

Also, if it is a one-time process, it is simple to do this as a "here document" where you type it in and execute it immediately. Of course, that has some drawbacks if you make a mistake. However, if you happen to be using the Korn shell the "fc" permits easy modification with an editor such as vi with immediate re-execution.

As a last picky point, it is a matter of convention with me: In script construction, I never use a variable name that may conflict with a known Unix name. Accordingly, the "file" in the script could be confused with the "file(1)" command.

Best to you,

Mike Smith, Atlanta, GA

Mike,

Thanks for your comments. It's nice to know that my work is read and scrutinized. I like the idea of saving a process with the use of "for file in *.z". I didn't know the for command supported that. I take your point about the word file, although I like my variables to have descriptive names -- I guess I should be more creative.

Question: I have a custom cable which was provided by a Fortune dealer for the HP LaserJet. What is the correct pin-out for the LaserJet cable on the Fortune 32:16? On the Fortune 5000?

While we are on the topic, what is the correct pin-out for a modem cable on the Fortune 32:16? Is pin 25 really used on the computer side to receive the DSR (Data Set Ready) signal? This seems very non-standard, but the use of DCE on the Fortune seems to result in a non-standard connection for DCE to DCE.

Very truly yours,

Rob Heinrich, Chicago, IL

Answer: This is our suggested cable configuration for the HP LaserJet:

32:16	LaserJet
2-----	2
3-----	3
7-----	7

In our experience, only pins 2,3 and 7 need be connected. Handshaking on the HP LaserJet should be set to ROBUST Xon/Xoff. On the 5000, the connections depend on which port you are connecting the LaserJet to. If you use an adapter on the port to get to a 25-pin connector, the connections from the 25-pin connector are as follows:

5000	LaserJet
2-----	3
3-----	2
7-----	7

If you are connecting directly to the 9-pin connector, you would use the following connections:

5000	Laserjet
2-----	3
3-----	5
7-----	9

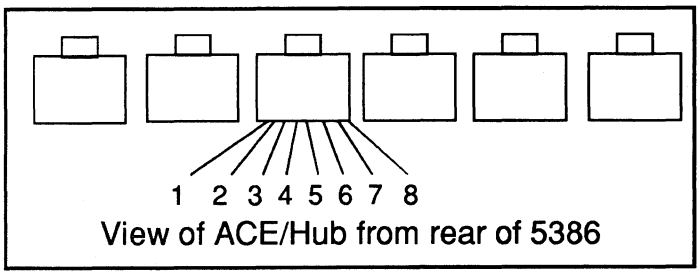
Perhaps it would be helpful to list out all of the signals carried by different connectors/machines:

Function	32:16	5000/9-pin	Hub-Ace	Maxspeed
Transmit Data	2	3	3	2
Receive Data	3	2	5	5
Request to Send	4	7	8	
Data Set Ready	6	6		
Data Term Ready	20	4	4	6
Data Car Detect	8	1	6	1
Ground	7	5	7	3

For serial communications, the transmit data line on one side must connect to the receive data on the other side. The same goes for receive data on the first going to transmit data on the other. Finally, ground must be connected together on both sides.

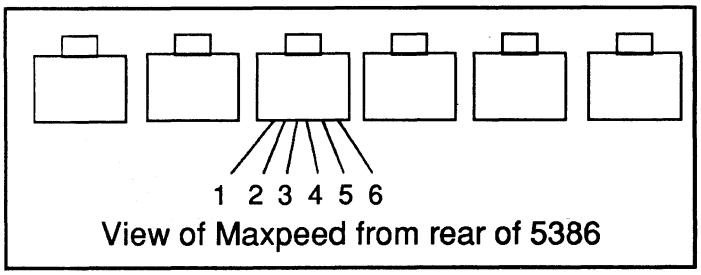
Fortune 32:16's/4000/8000 and modems all are DCE devices. Terminals, printers, and serial ports on the 5000 computer all are DTE devices. Straight-through cables can be used when going DTE-DCE, and null-modem, or wires with several crosses are used for DTE-DTE, or DCE-DCE connections.

To determine the wire positions on the ACE/HUB board modular connectors, the following chart should help:



View of ACE/Hub from rear of 5386

The connections for the Maxpeed Card are as follows:



View of Maxpeed from rear of 5386

To attach a modem to a Fortune 32:16/SX/4000/8000, the wiring is as follows:

Fortune	Modem
1-------	1
2-------	3
3-------	2
4-------	5
6-------	4
7-------	7
8-------	20
20-----	8

We are not aware of any particular use of pin 25 on the 32:16.

The connections to connect a modem to port com1 or com2 on a Fortune 5000 are using either a 9 or 25 pin connector:

5000/9	5000/25	Modem	
3	2-->---	2	Transmit Data
2	3--<---	3	Receive Data
6	6--<---	6	Data Set Ready
5	7-------	7	Signal Ground
1	8--<---	8	Carrier Detect
4	20-->--	20	Data Terminal Ready

Generally on a 5000, we connect the modem up to either the com1 or com2 ports, rather than ports on any serial board.

Question: I have purchased a "LAN" version of an uninterruptable power supply (UPS), which provides a wire to connect to the computer to warn of power failures. I am adapting a program from a UNIX bulletin board to automate the shutdown procedure, based on a signal through a serial (modem-control) port. When I have successfully completed this task, I will submit a full write-up for ***/u/fortune news****.*

Essentially, the program tries to open the serial port, but blocks in the open routine until the data carrier detect (DCD) signal is available. On DTE serial ports, this signal is detected on pin 8, with pins 6 (DSR) soldered to pine 8 (DCD). The other wire from the UPS is then connected to pin 20 (DTR).

On the Fortune 32:16 (a DCE device), does this mean that pin 20 (DCD) and pin 25 (which appears to be the DCE equivalent of DSR on the 32:16) should be soldered? Your thoughts would be appreciated.

Very truly yours,

Rob Heinrich, Chicago, IL

Answer: Thanks for all of your letters. It's nice to see people experimenting with new things for their computers. What Mr. Heinrich is doing is to create a program which will automatically shutdown his computer if the power fails for more than 5-10 minutes, after which time the battery in his UPS would expire. This is important if you leave your system on all of the time and you don't want to risk an extended power outage while you are out of the office.

I think the only line you'll need to monitor is DTR, which should be on pin 20 on the Fortune. I'm not quite sure what you mean by "the other wire" from the UPS. If you must attach another pin, you might try pin 4, Request to Send. As you point out, on page 2-11 of the For:Pro manual it indicates that DSR might go into pin 25 on the Fortune, although I've never heard of anyone using pin 25.

Good luck with the project. I'm sure many readers will be interested in your results.❑

Dumping BASIC Data to ASCII Files

We recently had a customer who needed to transfer the data out of BUSINESS BASIC into an ASCII file. There are many reasons one might want to do this: to dump a customer list into a database or Records Processing, to transfer information to another type of computer, etc. There are several ways to approach this problem, so we'll give you tips on how it can be solved.

The easiest way by far is to purchase the ASCII-LINK program from Unikomp in Texas (713 895-9900). You will find their advertisement elsewhere in this issue. The ASCII-LINK program offers several useful features including the ability to only output selected records, selected fields within each record, and the advanced capability to conditionally select records--for instance if the credit balance is greater than $1,000. Since we were just doing one file, we decided to be adventurous and try to do it ourselves.

It should be noted at the outset that these techniques will be useful to people who are familiar with BASIC and who like to experiment. Those of you who are less patient may be best served with ASCII-LINK.

Another preface is that this article was written by a real novice in BASIC. I am much more comfortable coding in C, so I opted for a rather inelegant BASIC approach. The assumption was that once I got the data out of BASIC, I could massage it. Users more experienced with BASIC may prefer to do more of the work within BASIC.

The first trick in moving from BASIC to ASCII is to create a file to print to, rather than a printer. In theory, this can be done by creating an empty file in the /dev directory, (e.g. the command > /dev/myspool will create an empty file called myspool), and then adding an entry for it into the ipl files in the /b/ipls directory. I've had some mixed success with this technique, so now I take a different approach. Suppose I know that printer 1 is attached to port /dev/lp. I temporarily rename (move) that device to a new name, /dev/lp.real. Then I create my empty file called /dev/lp as described above. This avoids changing any ipl files. Now when I tell BASIC to print to printer 1, the output will all go into the file called lp. When you're done, you copy lp to another file, perhaps /tmp/basdump, and move /dev/lp.real back to /dev/lp (mv /dev/lp.real /dev/lp). This may be all you need to do if you have a report which prints all of the desired data. However, in our case, we had to create a program to dump the data.

The next step after taking care of your printer output, is to determine the name of the file that your data is in. The file is probably in /b/BDATA somewhere. If there are any programs that already access the data you need, you may be able to check for the file name there.

If the data in each record of the file is less than 132 characters, then you may have it easy. The following program will print out all of your data with 1 record per line:

```
00010 BEGIN
00020 OPEN (1) "MEMBR"  (that's our data file)
00025 OPEN (6) "LP"
00030 READ RECORD (1,END=01000) A$
00040 PRINT (6) A$
00050 GOTO 00030
01000 END
```

Unfortunately, many files have record lengths greater than 132 characters. We were stymied about how to print all of the fields without creating a complicated I/O list. A timely phone call to Dave Kloes of Unikomp provided the technique that would be required. The trick is to pass the fields to the Unix echo command, and have that output to a file. With this program, it is not necessary to fool BASIC with the printer file, because we can specify our own destination file.

The BASIC program now looks like this:

```
00010 BEGIN
00020 OPEN (1) "MEMBR"
00030 READ RECORD (1,END=01000) A$
00040 SYSTEM "echo '"+A$+"' >> /tmp/memdump"
00050 GOTO 00030
01000 END
```

Line 40 calls the Unix echo command with the arguments of A$ (which contains the data). Please note the double quotation and single quotation marks and enter them exactly as shown. The single quotation marks are all the normal apostrophe. The output of the echo command is all diverted into the file /tmp/memdump. The reason for the double greater than signs is that these tell Unix to Append or add to the output file, rather than creating it fresh each time. If a single > was used, the output file would just contain the very last record in the file. You can change the name of the output file to whatever you'd like.

It should be noted that there are some pitfalls with this technique. For instance, if you have any special characters such as quotation marks, apostrophes, ampersands, etc. they may confuse the echo statement and the Unix shell.

One difference between the two methods of dumping (<132 or > 132) is that the first method outputs each record all on

one line, while the second puts each field on its own line. This latter approach may give you more flexibility if you need to reformat the data for input into another program. One way to reformat the data is with a "C" program. That's what we used because it does the formatting very quickly. However, if you don't know C, you could also accomplish the mission with a shell script. For a reference, we checked Carlton Haywood's script to reformat rolodex data that appeared in vol 6.4 of **/u/fortune news**.

In order to write the script, you need to know how many fields (lines) make up each record. Let's say for this example it's 9. A shell script to put out each field separated by tabs would be as follows:

```
while read l1
do        read l2
          read l3
          read l4
          read l5
          read l6
          read l7
          read l8
          read l9
          echo $l1  $l2       $l3       $l4       $l5       $l6
$l7       $l8       $l9 > /tmp/memformat
done < /tmp/memdump
```

In this example, we've put the actual tab character between each of the fields we're echoing. It would be as easy to surround each field with quotes, etc. If you want to put in quotes in the echo statement, precede them with a \ so that they will actually print out, e.g. echo \"$l1\" \"$l2\"...

The way the program works that for as long as the read of l1 works, it will continue on and read the rest of the data, which should take it to the end of the file. Note that the output of the formatting will go into the file specified in the echo statement, and that the input is specified in the done statement. If you want to make this more generic, you can cat the input in and redirect the output as in

```
cat /tmp/memdump | reformat > /tmp/memformat
```

If you do this, remove the > /tmp/memformat and < /tmp/memdump from the shell script.

One byproduct of this reformat is that the shell will trim off blank spaces at the end of each field. This confused me when I had miscounted the number of lines for each record. Each record had several blank lines, and although I was off in the total number of fields per record, it appeared that everything was fine for the first few records. In fact, it wasn't, but since $l1 was blank (it was actually the eighth line), it didn't affect the output until several records were read.

Have fun.❑

Josh Lobel

News from SCI/FORTUNE

Reduced Pricing:

SCI/Fortune has announced new reduced prices on several products in their line, including their newly released laptop computer, as well as memory on several other computers. These reductions reflect changes in the industry, and keeps **SCI/Fortune** competitive in the marketplace. Please contact your Fortune dealer for new price information.

Increased Speed:

(reprinted from **SCI/Fortune** press release)

SCI/Fortune is pleased to announce an even faster, more powerful version of the Fortune 5880 RISC System! We now offer 25 MHz processor speed on the Motorola 88000 chip set. Combined with RAM in Processor/Memory Packages, our high speed processor is rated at 32.0 VAX MIPS. This is significantly faster than many systems offered by our competitors at much higher prices.

The 25 MHz Processor/Memory Packages for the Fortune 5880 RISC System are available with 8 MB, 12 MB, 16 MB and 20 MB of RAM. As with the 16 and 20 MHz versions of the Fortune 5880, the Base System and a Hard Disk Option must be specified along with the selected Processor/Memory Package in order to configure a complete system.

Pricing and part numbers for the new 25 MHz versions of the Fortune 5880 are shown in the new September 1990 edition of the Fortune 5000 Price List. We think you'll agree that these new RISC systems offer price/performance that's hard to beat!

Informix Available for SCO/XENIX

SCI/Fortune is pleased to announce the availability of Informix software applications, orderable directly from **SCI/Fortune**, for Fortune 5000 series systems running Santa Cruz Operation's XENIX 2.3.1 or 2.3.2 operating systems.

Product Descriptions

The Informix product family includes several products:

Informix-SQL: Informix-SQL is a relational database management system that provides the tools required for building and maintaining a database, as well as tools for building sophisticated applications. Based on the Structured Query Language, SQL, developed by IBM, Informix-SQL offers users a powerful and flexible data management environment.

Major features of Informix-SQL include:

- RDSQL database language, an extension of SQL
- Screen and menu building utilities
- Relational report writer facility
- System menus, English prompts, and Help feature
- Transaction log, protecting the integrity of the database

Informix-4GL: Informix-4GL is a fourth-generation language designed specifically for Informix-SQL database applications. Using a fourth-generation language, a programmer can design and debug an application much more rapidly than in standard programming languages, such as COBOL or C.

Informix-4GL offers the developer many features:

- Self-contained programming environment
- RDSQL database language, an extension of SQL
- Screen and menu building utilities
- Report generator
- Customized on-line help
- Transaction log to protect integrity of database

Informix-ESQL/C: Informix-ESQL/C offers two sets of complementary tools for highly specialized applications. First, SQL statements can be embedded in C language applications. The programmer can prepare dynamic queries and manipulate SQL databases without ever leaving the C program. Second, this package includes programming tools that allow the user to call C functions and special C library routines while working in Informix-SQL.

The programming capabilities of Informix-ESQL/C include:

- Embedding SQL statements in C programs and routines
- Calling C functions and special C library routines from INFORMIX-SQL
- Performing DECIMAL and DATE type conversions and manipulations
- Using C functions in ACE reports (the report writer)
- Using C functions in PERFORM forms (the screen builder)

C/ISAM: C/ISAM is the industry standard "Indexed Sequential Access Method" for the UNIX operating system and the internal access and retrieval methods for Informix-SQL products. It is a library of C language functions for

creating and manipulating indexed files. C-ISAM also offers protection against data loss by providing audit trails, concurrency control, and a variety of locking capabilities.

Informix-4GL Rapid Development System: Informix-4GL Rapid Development System (RDS) is an interpreted fourth-generation programming language that dramatically reduces application development time by reducing compilation time. Informix-4GL source code is compile into p-code (pseudo-code), read into memory, and executed by a p-code runner (interpreter). Interpreted code has the advantage of being easily portable from one platform to another, greatly reducing development investment. Users can move applications to new environments without recompiling or rewriting any code, and without purchasing a second copy of Informix-4GL RDS.

Informix-4GL Interactive Debugger: Informix-4GL Interactive Debugger allows the developer to interactively debug applications compiled by the Informix-4GL RDS.

Compatibility with Other Versions of Informix

This series of Informix products is compatible with SCO XENIX operating systems 2.3.1 and newer.

The original Informix product, standard Informix, which is still offered on **SCI/Fortune's** 32:16 series, is not among the Informix products being offered here under SCO XENIX 2.3.1/2.3.2. Programs and databases built from the standard Informix product, based on a proprietary query language, are incompatible with the Informix-SQL based versions, and cannot be "migrated" from 32:16 systems to **SCI/Fortune** 5386 systems. Each Informix product (4GL, SQL, ESQL/C, C/ISAM, and 4GL RDS) is a standalone product, except that the 4GL Interactive Debugger can only be used in conjunction with the 4GL RDS product. The user need only buy the single item desired. All are built around C/ISAM, and each development version includes its own runtime. A users who wishes only to run third party applications (i.e.: the user has no development requirements) should purchase just the runtime Informix product.

RAM Expansion Board for Fortune 5386/33

SCI/Fortune is pleased to announce the availability of a Memory Expansion Board for the Fortune 5386 33 MHz system. This 32-bit board was designed by SCI specifically for SCI cache system boards. The memory board can hold eight SIMM memory modules, and can be loaded with either eight 80ns 256x9 SIMMs (providing an additional 2 MB of memory) or eight 80ns 1x9 SIMMs (providing an additional 8 MB of memory). All eight sockets must be loaded.

With this board, a user who currently has a Fortune 5386 33 MHz system fully loaded with 8 MB of memory can now double the memory to 16 MB.

Open Problem Report

FORTUNE:WORD 3.2.1

Spell Checker/Records Processing

Running Spelling Checker and then Records Processing (without editing in between) produces two copies of every record -- the second one not reflecting changes made during spell-checking.

Workaround: Edit and save the document after having run Spelling Checker and before running Records Processing.

Document Assembly

Running Document Assembly (or Records Processing without a List Document) produces an empty Output Document.

Workaround: Use Records Processing and specify and List Document (for instance an empty document).

Printing /Document Summary

Attempting to print a document that has a Document Summary with an empty Document title field and a Comments field with more than 36 characters causes **Fortune:Word** to abort.

Workaround: Use less than 37 characters in the Comments field or have something in the Document title field. Recommendation: Enter something in the Document title field of your Prototype(s).

Terminal lockup

Note: If **Fortune:Word** aborts it may seem that the terminal is locked -- to get back to a normal mode type (even though nothing may display while you're typing) stty sane and press the LF/GL key. If you're using a keyboard without a Line Feed key, you should instead press ^J (that is, hold down the <CTRL> key and hit the letter "j").

Solving Communications Problems with TERM

Easy of use, Powerful Scripting, Flexibility Make this Package a Winner

Some time ago, one of our clients came to us with an interesting problem. They had access to a computerized database of sales information which they were accessing with a small teletype terminal. The data would print out on rolls of paper, and they would reenter it into their Progress database on a Fortune 32:16 SX computer. They wanted to move a little further into the computer age by having their Fortune automatically download the information and import it into their database. Of course I said it would be "no problem".

As is the case with any new task, the first step is to break it down into its step-by-step components. This analysis produced the following steps:

1) Enter part numbers into database
2) Have Fortune call the remote database and log in
3) After successful login, send part numbers to remote system
4) Capture sales information into a file on the 32:16
5) Incorporate data into local database

Exporting the part numbers out of their database and importing the sales data are pretty straightforward with Progress. It seemed that the tricky part would be the automatic calling and downloading of sales information. In the past, I had used communications packages like **ckermit** or **Handshake** to automate the login process, but there didn't seem to be an easy way to capture the data that was coming back. Then I discovered the **TERM** communications program which has all of these capabilities built in.

Term is a general purpose communications package. In addition to providing terminal emulation for any Fortune, it also includes many bells and whistles. In the area of emulation, it let's your Fortune act like a vt52/vt100, tv926, tv912, wyse50, adm1, ib3101b, and ib3101l. On the 5000, SCOansi is also available. File transfer protocols include its own proprietary error-checking schemes, as well as Kermit, unformatted, and line-by-line. On the 5000, Xmodem and Ymodem are available as well.

The features which really enabled Term to be able to meet my client's needs were all included in the very rich script language which is a part of Term. It's quite possible to create a script which will automatically call the desired number, wait for a login message, login, send data up, start capturing data back based on a certain response, close the capture when certain text is received (or after a timeout), and logoff the computer. If there are problems in the communication, Term will automatically try to correct the trouble, e.g. dialing the number again if the line is busy.

Key features for me were the ability to easily send the part numbers up, and to download the response back into specified capture files. In essence, with the help of Term, most of the requirements for the job were met by just creating the appropriate Term scripts.

One of the nice things about this package is that it runs on many different hardware/software platforms, including all of Fortune's line. Scripts that are created on one machine should run on other machines with minor modifications.

Another valuable feature is that Term has been designed with ease of use in mind. It is operated primarily from a strip menu (similar to Multiplan/Lotus) which is used to access most features. The configuration can be changed through the menu, and specific settings for specific sites can also be stored. If you want to capture into files on the fly, it's easy to turn the capture on and off. When (and if) you get stuck, the F2 key will provide online help.

Although complicated scripting will take some experimenting, Term gives you the benefit of several sample scripts which use a great variety of its functions. You can even create a bulletin board just by using a Term script!

Term was installed at our client's site over two years ago, and it has been working fairly reliably ever since. We have recommended it to several other users on both 32:16's and 5000's, and they have been pleased.

Term is a product of Century Software in Salt Lake City, Utah. Their number is 801 268-3088. If you are interested in more information, contact your dealer, /u/fortune news, or Century Software.❑

Josh Lobel

Goings on in Boston

Using DOS files under Unix

It's been kind of a busy month in Boston. I guess the early cool weather scared everyone into thinking summer was over, and the time had come to get back to work. Fortunately in the last few days the temperatures have gone back up to the high 70's and the minds are beginning to melt once again.

Working with DOS under UNIX

Today I was working with a client who was using AIX on an IBM model 70. AIX is IBM's version of UNIX. They asked me if it was possible to copy files between DOS and UNIX partitions on a hard disk. I answered that I was sure it was, although I couldn't confirm exactly how AIX would handle that. I came back to my office and started playing with some of the options under Interactive 386/ix to familiarize myself with how this can work. (I still don't have the answer on AIX yet.)

The easiest method for letting DOS communicate with Unix is with VP/ix. This is a program that lets DOS function under Unix. We've talked about it before in these pages. With VP/ix, you can run Lotus, Pagemaker, Word Perfect, etc. from within the Unix operating system. This works best on the system console, but can work with some character based programs on other terminals. We've come to the conclusion that this program is extremely useful as a convenience for running the occasional DOS program. We probably wouldn't suggest it for heavy-duty DOS use because of the impact it has on the other users, and because it isn't quite as fast as a stand alone DOS computer.

VP/ix provides an easy way to move files between DOS and Unix partitions. When you enter VP/ix, you'll probably get a C> prompt, which is the familiar DOS prompt. It would even appear as though you are working on the C drive. In fact, you are working on a facsimile of a disk drive -- it's a pseudo drive. Typically the C drive is a file called C: in the /usr/vpix/defaults directory. This file has a peculiar property which you should be aware of -- it will grow as you copy files to it, but it will never shrink when files are removed. Users should be cautioned not to copy files to the C drive, or it will become very unwieldy. (We'll talk about ways to shrink it next time.) We find it preferable to work with the Z drive. The Z drive is actually the Unix file system. You'll probably find familiar directories on the Z drive, like all of your usr directories. Under VP/ix, like DOS, you should reverse the directory slashes, e.g. /usr/josh is \usr\josh. You may find some interesting file or directory names when you're in VP/ix, like CONVAB. These occur when Unix file names exceed the DOS file name restrictions (8 chars with 3 char extension) or when file names contain capital letters.

You can use the DOS copy command to copy files onto the Z drive. If you want to convert a DOS text file to a Unix text file, you can add the /u flag to the copy command (copy /u dosfile unixfile). Going the other way requires a /d flag. If you want to move around in the Unix file system, you can go manipulate directories or files that have had their names altered -- you'll just need to address them by their VP/ix names. For instance, to copy the C:\AUTOEXEC.BAT file to Z:\usr\conversion, you might end up with a command like this:

```
COPY C:\AUTOEXEC.BAT Z:\USR\CONV~AB
```

Using Dossette

Even if you don't have VP/ix, you can still read and write from DOS formatted floppies. The dossette command includes a small subset of DOS commands, such as dir, copy, erase, type, etc. If you simply type dossette, you'll be greeted with an A> prompt. If you insert a DOS diskette and type DIR, you'll get a directory of the DOS drive. You can copy DOS files with the DOS command. If you created a DOS partition on your hard disk and formatted it for DOS, when you type C: you'll be moved over onto the C drive which is the actual C drive (unlike the one under VP/ix). You can copy files between the floppy and the C drive as you normally would under DOS. If you want to interact with the Unix file system, you'll need to use the get and put commands. These work from the perspective of the Unix file system. For instance, if you wanted to copy C:\AUTOEXEC.BAT to the /tmp directory under Unix, you'd type:

get C:\AUTOEXEC.BAT /tmp

Even though your prompt may still say A> or C>, the get command is working from the Unix file system. The put command will copy from the Unix side over to DOS:

```
put \usr\josh\filename C:/DATA
```

copies from Unix to DOS. The put and get commands

Goings on in Boston, Using DOS with Unix, cont'd

will attempt to determine whether the files are ASCII or binary. If they're ASCII, new-lines will be changed to carriage-return/linefeed. This is similar to the /u and /d flags under VP/ix. If you choose, you can force the conversion by including a -a with the put and get commands. A -b will insure that no conversion is performed. You can read about these commands under dossette in section 1 of the User's/Administrator's Reference Manual from Interactive. You leave dossette by hitting a CTRL-d.

You can activate dossette commands without actually entering dossette, by preceding the commands with dos, e.g. doscopy, dosdir, dosget, dosput. These commands all default to the floppy disk, so if you want to use the DOS partition (C drive), you'll need to explicitly say C:. For instance:

dosdir gives directory of floppy (A:)
dosdir c: gives directory of hard disk DOS partition (C:)

Mounting the DOS partition under Unix

This provides a rather elegant solution to those users who prefer Unix commands. It is possible to actually mount the DOS partition as a file system under Unix. With interactive, you may want to first create a directory name to mount the partition to, e.g. /dos. Simply use the mkdir command to create it:

```
mkdir /dos
chmod 666 dos
```

The chmod command makes the /dos directory accessible to everyone. To mount the first DOS partition as /dos, type:

```
mount -f DOS /dev/dsk/0p1 /dos
```

You'll get a message warning you that /dev/dsk/0p1 [DOS] has been mounted as /dos. This is normal. If you type

```
ll /dos
```

you'll see a Unix style directory of your DOS partition. You can use Unix file manipulation command like cp, ll, cat, grep, etc. to work with the DOS files.

It's also possible to have the DOS partition automatically mounted every time the system is booted by adding the line

```
/dev/dsk/0p1 /dos DOS
```

to the end of the /etc/fstab file.

You'll probably have to experiment some to do exactly what you want, but this should give you an idea of the possibilities. We'll be writing more about this in the future.❑

Josh Lobel

/u/fortune news
225 Crescent Street
Waltham, MA 02154