

UNIVERSITY OF MINNESOTA COMPUTER CENTER
Deadstart Systems Report

20 April 1982

Vol. 8, No. 6

Send all comments, criticisms and contributions to the editor: T. W. Lanzatella,
University Computer Center, 2520 Broadway Drive, Lauderdale, MN 55113.
The University of Minnesota is an equal opportunity educator and employer.

TABLE OF CONTENTS

NOTICE OF CHANGES TO THE SYSTEM	80
PROPOSED CHANGES TO THE SYSTEM	81
VAX/VMS PORT AUTHORITY - S. E. Collins	81
PWG - A. B. Hastings	81
PACMAN - A. B. Hastings	82
GAME - A. B. Hastings	82
WHEREIS - A. B. Hastings	82
AUTO-DIALER FOR UNIX - B. E. Blasing	83
CU - B. E. Blasing	83
DISCUSSION TOPICS	
NAMES FOR MINNLIB ROUTINES - M. Frisch	84
UNIX TAPE MOUNTING - A. B. Hastings	85
SYSTEM MAINTENANCE	87
LAST WEEKS SYSTEMS GROUP MEETING - T. W. Lanzatella	87
CALLPRG AND LIBRARY TAPE NEWS - M. Riviere	87
CYBER DEADSTART DUMP ANALYSIS - B. E. Blasing	87
VAX/VMS DOWN TIME SUMMARY - M. Riviere	88
VAX/UNIX DOWN TIME SUMMARY - C. Boylan	88

NOTICE OF CHANGES TO THE SYSTEM

Andy Hastings deleted some unneeded code in 1AJ.

Don Mears changed DSP to correct an error in the banner page for Hasp terminals.

Bill Sackett corrected an error in PFDUMP which caused an occasional mode error. Bill also repaired an error in COPYCAT wherein DN and PN processing was confused by family names.

PROPOSED CHANGES TO THE SYSTEM

VAX/VMS Port Authority - by S. E. Collins

There is a shortage of ports on the VMS system. Often, a disproportionate number of ports is tied up by UCC staff, preventing other users from using the system. I propose a program be called at every log in to control port access. This program, PORTQUOTA, will read a quota data file, determine if the user's group has already filled its port quota, and, if so, log the user off with an appropriate message.

The quota data file will assign each user identification code (UIC) to a group, and then limit the number of ports on each controller that group can occupy at any time. A sample portion of a quota data file is shown below. (This is an example only; the final format selected may differ slightly).

Groups					
Graphics		(011,016)	(011,906)	(011,940)	(011,941)
		(011,942)	(011,943)	(011,944)	
System		(00*,***)			
Staff		(011,***)			
Users		(***,***)			
MON	08:00	8	A-----	Staff	
		8	A-----	Graphics	
		4	-BC-----	Staff Graphics	
		32	ABCD-----	Users	
	17:00	6	A-----	Staff	
		6	A-----	Graphics	
		2	-BC-----	Staff Graphics	
		32	ABCD-----	Users	
	24:00	16	ABC-----	Staff Graphics	
		32	ABCD-----	Users	

TUE... etc.

In this example, from 00:00 to 08:00 on Mondays, up to eight staff accounts may log in on controller A; up to eight graphics accounts may log in on controller A; up to 4 staff or graphics accounts may log in on controllers A and B; and up to 32 other users may log in on controllers A, B, C, and D.

//////////

PWG - by Andy Hastings

Each user on the Unix system has an associated group name. These groups are used to determine file access permissions. The initial group name is determined at login from the /etc/passwd file. Certain validated users are allowed to change groups via the newgrp(1) command. Unfortunately, there is no method (apart from creating a file and using ls(1) to see what group it is in) to determine the name of the group to which the user

is currently assigned. Therefore, I would like to implement a pwg(1) command which will print the name of the current group on the standard output. A manual page will, of course, be provided. Hopefully this command may be removed when we install the next release of UNIX, in which a user may be in all validated groups at the same time.

//////////

PACMAN - by Andy Hastings

We have received through USENET (an informal network of UNIX sites connected via auto-dialers) a new UNIX game named pacman(6). This game is based on the popular Pacman video game. I would like to install this program in /usr/games. To install this program properly, we also need to install a nap() system call to pause for a given number of clock ticks. This system call could also conceivably be used in the termcap library instead of pad characters to reduce system overhead. I will come up with manual pages for nap and pacman.

//////////

GAME - by Andy Hastings

We have also received through USENET a program to control access to games. Game(1) may be used by the super-user to turn off all access to games, to provide unlimited access to games, and to provide guarded access to games. When guard mode is selected, users must use "game gamename arguments" instead of "/usr/games/gamename arguments" to call up a given game. Game will check to see that there are no more than five users on the system before allowing access to a protected game. The list of protected games is stored in /etc/games. Game also provides the capability to run games at a lowered priority (via nice). Additional checks may also be installed in game as needed. Documentation was provided with the program.

//////////

WHEREIS - by Andy Hastings

Whereis(1) is a Berkeley program which reports the locations of the source, the documentation, and the binary for a particular program. It contains a list of directories that it searches to locate the indicated items. Unfortunately, there seems to be a slight oversight: whereis searches /usr/local, where local binaries are stored, but it does not look at /usr/man/man1, where local documentation may be found. I propose to rectify this oversight by including /usr/man/man1 in the list of documentation directories in whereis. No change to whereis documentation is necessary.

//////////

Auto-dialer for UNIX - by B. E. Blasing

This and the ensuing proposal consist of ideas that have already been conceived and implemented. Their presence here is for informational purposes, and also to gain system group input. Any ideas are welcome. Future changes to VAX/UNIX will be preceded by a proposal.

An auto-dialer is a modem that can place an outgoing call on a phone line. There exist already several programs on UNIX that make use of an auto-dialer, the most notable being 'uucp' and 'mail'. uucp allows file transfer and remote command execution between cooperating UNIX systems. Mail is the electronic mail facility which can send mail to remote machines as well as the local machine. To use these programs, we needed an auto-dialer.

TCJ and I constructed an auto-dialer to do pulse dialing using a Penril modem and some spare parts. Special mods to the dz driver were made to accept a phone number and control the dialing from the RS-232 interface. Dialing phone numbers of > 8 digits are allowed only from the super-user or group sys people.

The installation of the auto-dialer and the dz mods are the subject of this proposal.

//////////

CU - by B. E. Blasing

Cu (call up or call UNIX in some circles) is a program which functions the autodialer with a phone number, waits for the called system to answer, and then proceeds to imitate a terminal so one can converse with the remote system. It has facilities to divert output to a UNIX file, send a UNIX file to the remote system, etc. Currently its argument list looks like:

```
cu phonenum (option arguments)
```

I wish to make the phone number also optional, and if left off, cu will not try to function the auto-dialer, but rather just start communicating with the specified communication line. E.g.,

```
cu -s 2400 -l /dev/ttyl5
```

gets you to the Cyber 730 currently (through NAM/CCP alas).

I also wish to change the default speed for the -s parameter to 1200 from 300. Our auto-dialer is almost always set for 1200 baud. I also want to add the speeds 2400, 4800, and 9600 to the -s parameter. The documentation says it stops at 1200 currently (the documentation will be updated).

DISCUSSION TOPICS

Names for MINNLIB Routines - by Mike Frisch

I am planning to revise the way new names are chosen for routines on MINNLIB. The first letter will denote a category, the last letter will denote a variable type (if appropriate) and names will be at most 6 characters to comply with the current FORTRAN standards.

The categories I have defined so far are:

- Approximation
- Differential equations and quadrature
- Eigensystems
- Integer functions
- Matrices
- Plotting (Printer plotting and Plotter plotting)
- Random number generators
- Special functions of math and statistics
- Utility routines
- Vector operations, including sorting and merging
- Zeros of functions

The variable types for the last character (if needed) are:

- Real
- Double precision
- Integer
- Complex
- Text (for type character)

Some examples of old and new names are:

- MXLNEQ becomes MSOLR
- DMXLNEQ becomes MSOLD
- SORT1 becomes VSRT1I for integer or
VSRT1R for real
- SCLPLT becomes PRSCL
- ISERCH becomes ISRCHI

Only names of new routines will follow these rules. Current ones will not change unless the routines are revised to have new parameters that cause us to change names to protect the innocent user. This is happening for a number of MINNLIB routines as a result of the conversion to the Cray.

The background for this naming scheme is that the best known commercial library, IMSL, follows a similar one. I have chosen different first letters than IMSL because our categories are somewhat different than theirs. They have only a couple of routines with different variable types so they haven't chosen a specific position for variable types as I have had to do. We have a fair number of routines that apply to different variable types and will have even more in the future.

A second part of the background is that in the past we have chosen names referring to the method used rather than to what the routine does. For example, RK solves differential equations by the Runge-Kutta method. Only users who are quite familiar with numerical methods would know what RK means.

The last part of the background is that in converting to Cray, a number of nonstandard FORTRAN features we have used in the past needed to be revised in order for the routines to work on the Cray. While the length of names of routines hasn't been a problem, the trend is to make our codes as standard as possible to ease the transfer to future machines.

A negative aspect to this plan is that names of routines will probably be less mnemonic. SORT1 is probably easier to remember or recognize than VSRT1R.

If you have any comments about these ideas, I am interested in them.

//////////

UNIX TAPE MOUNTING - by Andy Hastings

The current implementation of UNIX does not provide any method for queueing and processing magnetic tape mounting requests. Several users have requested the capability of using tapes on the UNIX system, and thus it is appropriate that we consider implementing some method for handling tapes. Here are a few suggestions that Chris Boylan and I have discussed.

- 1) Discourage usage of tapes on the UNIX system. Do not provide any system software to control access to the tape drive. Users wishing to use tapes must call up the operator at Lauderdale to request that a tape be mounted, and again to request that a tape be dismounted. Advantages: No software development necessary. Disadvantages: Considerable inconvenience for user and operator. No accounting for usage of tape drive. No protection for user's tape (anyone may read and write the tape drive at any time).
- 2) Require that user "operator" always be logged in to the UNIX system. Either move the VAX console so that it is near the Cyber consoles, or use the DECwriter next to the MERITSS console as a "remote console" for monitoring the UNIX system. At this point, there are several options.
 - a) Do not provide any system software to control access to the tape drive. Users wishing to mount tapes must use "write operator" to send an inter-terminal message to the operator requesting that a tape be mounted, and again to have the tape dismounted. Advantages: No software development necessary. More convenient for users than using the telephone. Disadvantages: Still no accounting for tape drive usage. No protection for user's tape.

- b) Write a systems program which will run a user's shell script (procedure file). Before running the script, the program will request that the operator mount the tape. Whenever the shell script terminates (normally or abnormally), the program will request that the operator dismount the tape. The program will run with special privileges so that it may switch ownership of the tape drive to the user running the shell script. It will also record tape drive usage in a log file somewhere. Advantages: User's tape is protected. Tape drive usage is accounted for. Tape drive is not tied up for long periods of time (unless the shell script contains some sort of loop). Disadvantages: Extremely inconvenient for users. Batch access to tapes is contradictory to UNIX time-sharing (i.e., interactive) philosophy.
- c) Write a systems program which requests that the operator mount a tape (in a manner similar to proposal 2b). The program will run with special privileges so that it may switch ownership of the tape drive to the user requesting the tape. After the tape has been mounted, the user may access the tape interactively; the tape-mounting program will sleep in the background, waking up periodically to see if any other user is requesting the tape drive. When the user is done with the tape, he or she enters "dismount" or kills off the sleeping tape-mounting program. The tape-mounting program will then notify the operator that the tape is to be dismounted, and terminates after releasing the tape drive. (The program will also perform these operations if it is killed by the user's logging off.) Tape drive usage will be accounted in the same manner as in proposal 2b. Note that this proposal implies a queue to keep track of pending tape requests. To discourage long-term usage of a tape drive, the tape-mounting program will issue warning messages after 15 minutes of use or when another user requests the tape drive. These messages may become more frequent if the user continues to monopolize the tape drive, and if another user is waiting, the tape will be automatically dismounted after 30 minutes. Advantages: User's tape is protected. Tape drive usage is accounted for. Convenient to use. Queueing mechanism to prevent frustration. Fits in well with UNIX philosophy. Disadvantages: Non-trivial software development. Possibly annoying messages when tape drive is monopolized.

In all cases, it will be necessary for the user requesting the tape to specify the UCC storage VSN, the tape density (if not 1600), the ring status (in or out), and, if the write ring is in, the paper label ID.

SYSTEM MAINTENANCE: People and Procedures

Last Week's Systems Group Meeting - by T. W. Lanzatella

Andy Hastings' proposal to change the UNIX error processor, ERROR(1), to process nonstandard Pascal errors in the same way it processes other errors was approved (see DSR 8, 5 p 68).

Regarding Susan Steffen's article describing various aspects of language processor support, Larry Liddiard recommended that the portion describing which languages we offer and their support level be excerpted and published in the UCC Newsletter.

//////////

Callprg and Library Tape News - by M. Riviere

On April 27th, A. Arndt will modify the Callprg index of the Cybers 74/172 to update the entries for the future versions of COBOL, COBOL5, FORM, FILE and SORTMERG. All the new products are from the 552 PSR level.

Also on April 27th, S. Steffen will change the entries for the future version of BASIC on the Callprg index of the three Cybers. The new future BASIC is the 535 PSR level of the compiler. In addition, Susan is also adding an entry for future GPSS and for SIMII5 (SIMSCRIPT version 2.5). These two additions are only for the Cybers 74/172. The new version of SIMSCRIPT is available for academic use only and is therefore restricted to be accessed by selected user's accounts.

The next set of Library Tape and Callprg changes will take place on May 18th. Requests for that date should be submitted before May 6th by noon.

On April 27th, R. Fletcher will replace REDACT on the Library Tape of the three Cybers. Rod fixed a bug that was introduced with the last modification and made REDACT miss some entries when writeups and permanent file names were not identical.

//////////

Cyber Deadstart Dump Analysis 4/5 - 4/18 - by B. E. Blasing

Cyber 730

No unscheduled deadstarts (again) for the period. Last crash was 3/18.

Cyber 74

Sat 4/10 - 15:52

DD2022

Mode 1 in CPUMTR. analysis showed several exchange packages had been written two words off from where they should have been.

No resolution.

Sat 4/10 - 16:00 No dump!
Mode exit again in CPUMTR. For unknown reasons, no dump was taken.

Wed 4/14 - 03:47 DD2024
All jobs began aborting. A memory core stack broke and was replaced.

Cyber 172 - MERITSS

Tue 4/6 - 21:47 DD14
Mode 1 in CPUMTR looks like bit 12 in register A0 got picked in a SB4 instruction. CE's couldn't find anything. The problem is very intermittent.

Thur 4/8 - 09:44 DD21
Mode 40 in CPU1 while running the idle program.

//////////

VAX/VMS Down Time Summary From 3/31/82 to 4/15/82 - by M. Riviere

Friday 4/2 8:00 - 8:22

System came up late due to delay on system pack backup.

Monday 4/5 10:34 - 10:36

Fatal bug check. System auto rebooted.

Friday 4/9 8:00 - 9:15

Delay on reformatting and reloading the USERA pack. The reloading job started at 6:50 and was aborted to bring up the production system.

//////////

VAX/UNIX Down Time Summary - by C. Boylan

Tuesday, 4/6 16:00 - 16:03

System was rebooted to unhang DH. This problem looked to be the same as with the VMS DH but it now appears that the DH was merely broken. No dump.

Wednesday, 4/7 16:00 - 16:03

Same as the previous day.

Thursday, 4/8 08:05 - 08:40

CMI came in late to work on DH.

Saturday, 4/10 08:15 - 09:20, 15:30 - 15:45

DH stopped working and hung the UBA. Rebooting fix things.

Monday, 4/12 15:50 - 16:23

DH hung UBA again. CMI was here and took the system to check out problem.