FEB-80 MAR-81 VOL III NO III

AUSTRALIAN UNIX USERS GROUP NEWSLETTER

CONTENTS

AUSTRALIAN UNIX USERS GROUP NEWSLETTER

Next AUUG Meeting

At the last group meeting, held at the AGSM in March, Queensland or more specifically the two Queenslanders who were stupid enough to show up volunteered (arms twisted behind their backs) to hold the next meeting in sunny Brisbane. Brisbane is rumored to be quite nice in winter.

Well, as luck would have it, Decus Australia will hold their eleventh symposium in Brisbane on 24th-28th August, 1981. Its on at Griffith University and a lot of interest has been expressed in Sydney in holding the UNIX meeting either before or after the Decus meeting to allow attendees to visit the 'impressive exhibition of systems, peripherals and documentation' to be mounted by DEC or even to attend the Decus meeting its self.

Organisation for the Decus meeting looks good and if the Queenslanders are willing we should all get together in Brisbane in August. How about it Ross and Rick?

In this issue

Well you cant say AUUGN is not good value for money. Another reasonable sized issue. A good thing after that last door-stop eh?

I have received a set of US newsletters from Wally Weddle, the US newsletter editor (yes Maude, they do exist!) and look forward to receiving more. The latest CUUGN (now called UNEWS) contains another summary of the SF meeting which is reproduced here. Now you can check what I got wrong last issue.

An article on V6 to V7 driver conversion is worth a read as is 'V7 on the 11/44' The UNIX-C bibliography is a MUST even though the fine print is a strain on the eyes.

Heriot-Watt V7 Strip-down

I have also received a copy of the Heriot-Watt stripped down V7 distribution. The documents accompanying it are reproduced here. This is the only copy you will get out of me, so don't start any fires with this AUUGN. Should you want a copy of the tape, and you are happy to abide by the conditions set out under 'AUUG - Heriot-Watt Software Exchange Agreement', send me a signed copy of the agreement, a good 2400ft magnetic tape, a cheque or money order for ten dollars (thirty dollars Australian if you are an overseas reader) and a copy of your UNIX V7 license. I will return the tape written at 800 BPI (or 1600

BPI if you request it).

Be WARNED!!! Tapes and monies received without the exchange agreement or license copy will be considered as a donation to the AUUGN coffers.

When Rick Stevenson was here for the last meeting, he left a copy of his V7 strip-down (big UNIXed and all that), but Rick said he would rather wait a little longer to put together a real distribution.

Netmail

I have started a new section this issue called Netmail. I receive quite a bit of mail over the net and rather than re-draft it into something prettier I plan to put it in, warts and all.

This issue has some from Antarctica.

Change of AUUGN Editorship

I hereby give notice that my term as editor of this little publication is coming to an end. The last issue I plan to produce is volume three number six, that is in August-September 1981.

The next editor should be chosen at the winter meeting. Nominations may be sent to me for publication prior to the meeting. I have nobody to nominate since all the people I can think of who would be silly enough to take it on could never do as good a job well as me. How is that for pride in your work, yuk yuk yuk.

So thats what kangaroos keep in their purses!

Peter Ivanov
Dept. of Computer Science
Electrical Engineering
PO Box 1
Kensington 2033
AUSTRALIA

(02) 662-3781

Australian UNIX User Group Meeting, March 16 1981

The meeting started with the usual site introductions.

Robert Elz spoke first of his stay at Berkeley but prefaced his comments by saying that any thing he said was all lies. Having disclaimed, he spouted on for a while, and some of the things he said were:

- Judging by the number of Berkeley licenses compared to the number of UNIX 32V licenses it appears that most VAX users run the Berkeley software.
- They are looking at a system that will boot on any 'standard' VAX, that is a self re-configuring system (within limits).
- Developments include performance improvements in the file system and in networking for the ARPA project. An interesting idea is that of 'migratory files', that is the more used files migrate towards the center of the disc.
- Multiple processes should be able to map files into their address spaces.
- Vfork is dead, replaced by a copy-on-write fork. The parent and child share the same address space, the child with read only permission. Pages are only duplicated when the child tries to write them.
- Improved software, eg F77 produces better code, a better i/o library, a new tar etc etc etc

Next was Ian Johnstone. He did not say anything, and even seemed reticent in saying what he didn't. Rumor has it that:

- The Bell internal UNIX 4.0 needs to use kernel overlays, even on a PDP11/70! UNIX on a large IBM machine makes a great personal computer. Response is almost precognitive.
- Ian has been working lately on a hyperchannel link between a bunch of 70s, Vaxes etc. Hyperchannel is nice but a flat out 70 can only manage about 30Kbytes/second. Other methods of machine interconnection are or have been investigated, such as DEC PCLs, back-to-back DMCs, X25 connections etc etc.

Dave Horsfall gave a talk on the SCCS system. His summary is included later.

Ross Gayler from Psychology, University of Queensland, spoke of UNIX in the banana republic. An interesting point from a 'survey' he had conducted lately was that 23% of UNIX sites in Australia were in Computer Science departments, which oddly enough seemed to also possess 52% of the CPU power. Psych have a finger in most aspects of computing, particularly text processing. They have a set of macros for APA style papers. Also an abbreviation expansion program and a referencing system.

Greg Rose gave a very amusing talk about a power station control simulation project. He promised to summarise it, but I have not received the copy at this time.

There was some discussion about network connection via CSIRONET. A letter and demo appears later. A long tape copying and exchange session followed afternoon tea.

SCCS revisited

Dave Horsfall
Computing Services Unit
University of NSW

G'day, ladies and bruces. If you were at the last UUG meeting at AGSM, you will recall that I attempted to present a demonstration on SCCS. However, the gods were not smiling upon me at the time and I lurched from one catastrophe to another. First, an overhead projector could not be found, then there was no power point available, then one was found which was not "active", then I couldn't figure out how to turn the O/P on (it was a bar switch disguised as trim), then there were software problems, then the machine crashed, and finally the meeting ran out of time.

The software differences cropped up when I made a trivial change to ADMIN to preserve the old ownership of a newly copied SCCS file. It would appear that in my ignorance I assumed the standard I/O libraries (we went from the level 6 version to the level 7 version) would be compatible with existing software - the original SCCS programs being compiled with the old library. Silly me. ADMIN got its knickers in a twist when it tried to seek back to the start of the file to rewrite the checksum. There seems to be a problem with seeks on a buffered file which disappeared when I relinked with the old library. So much for standard libraries.

Anyway, all has now been fixed up and I present for your titillation what would have appeared on the big screen. The demonstration was to show a modification to the text editor EM. The editor recognises various files as being illegal e.g. a.out binaries, archives etc. The modification was just to add to the list of magic numbers the code for the new style packed file, viz "017037". It also demonstrates the protection features of SCCS, in that only people in the list of authorised users can make deltas. Now, if you can use your imagination a little and pretend that this is a terminal session, I'll take it from there . . .

The first performance is the protection feature. I log in as someone who has no permission to change the file and I try and change it. An error message results which I interpret with the HELP command, and follow its advice.

```
CSU login: visitor
Wed Mar 18 10:45:37 1981
Uni of NSW CSU PDP-11/40 : UNSWCSU

LIMITS: no disk limit, 6 processes, 1 printer unit

% get -e %s/S/em.c
1.20
ERROR [/srce/usr/source/S/s.em.c]: not authorized to make deltas (col4)
% help col4

col4:
"not authorized to make deltas"
```

```
Your User ID is not on the list of users who are allowed to add deltas to this
file.  You can execute "prt -u file" to see who is allowed.  See your
project administrator to get your login on the list.

% prt -u %s/S/em.c
/srce/usr/source/S/s.em.c:

Users allowed to make deltas --

        nikn
        dave
        greg
        munro
%
```

For my next trick I will show the actual process of getting the file, modifying it and testing it.

```
CSU login: dave
Password:
Wed Mar 18 11:35:06 1981
Uni of NSW CSU PDP-11/40 : UNSWCSU

LIMITS: no disk limit, no process limit, no printer limit

% get -e %s/S/em.c
1.20
2811 lines
% e em.c
54341
/017437/
                017437,                         /* packed */
c
                017437, 017037,                 /* packed */
.
%

checkfile(fstwd)
register fstwd;
{
        static list[]
        {
                01,                             /* dec object */
                0404,                           /* pascal obj */
                0407, 0410, 0411, 0412,         /* objects */
                017437, 017037,                 /* packed */
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                0177545, 0177555,               /* archives */
                0121212,                        /* slup library */
                070707,                         /* cpio library */
                0,                              /* end-of-list */
        };
        register *lp list;
```

```
        while (*lp)
w
54348
!cc %
!cc em.c
!
q
% pack TODO
pack: TODO: 35% Compression
% a.out TODO.z
Illegal file type
0
q
% file TODO.z
TODO.z: packed (new format)
%
```

And now, to cap it all off, the change will be recorded and the new version "made" and installed. Note that I have to "get" the version again to ensure the internal keywords are replaced. I also show the use of the WHAT utility to demonstrate the identification feature as well. The lines after the DELTA command refer to the new version ID and the number of lines inserted, deleted and left unchanged. Note that a line which is changed is treated as a deletion followed by an insertion.

```
% rm a.out
% delta %s/S/em.c
comments? Recognise the new packed format in the list of\
illegal file types, to wit "017037".
1.21
1 inserted
1 deleted
2810 unchanged
% su
Password:
# cd %s/S
# get em.c
1.21
2811 lines
# make em
>cc -I/usr/include -O -w em.c -s -n -o em
# what em em.c /bin/em
em:
        em.c    1.21
em.c:
        em.c    1.21
/bin/em:
        em.c    1.20
# cp em /bin/em.new
# rm em*
# prt em.c

s.em.c:
```

```
D 1.21  81/03/18 11:52:54       dave    22      21      00001/00001/02810
Recognise the new packed format in the list of
illegal file types, to wit "017037".

D 1.20  80/12/23 13:07:14       dave    21      20      00001/00001/02810
Allow capital Y in response to "Are you sure" prompt.

D 1.19  80/12/01 11:58:02       dave    20      19      00005/00001/02806
Correction to last mod - `.' was not left at last line sometimes.

D 1.18  80/11/28 17:01:43       dave    19      18      00013/00003/02794
>From R. Bullock: When scrolling with %1 or "1, adjust line count
if a line had to be folded so you don't lose the top few lines.

D 1.17  80/11/26 14:16:12       dave    18      17      00006/00001/02791
When verifying spelling, convert to lower case !!!

D 1.16  80/11/18 17:05:35       dave    17      16      00072/00008/02720
1) Acknowledge Richard Bullock's last 'e' change.
2) Add R.B.'s '#' command to enable/disable line numbering.
3) Putchar('\0') will flush out its buffer.

D 1.15  80/10/10 17:15:46       dave    16      15      00010/00005/02718
1) Fix up "e (space) <cr>" - it used to clear the buffer.
2) Allow "e<cr>" to re-edit the file (it may still prompt you).

D 1.14  80/10/08 16:37:42       munro   15      14      00175/00097/02549
1. Allow g command to work on empty file in silent mode.
2. Cause the command "g/pattern/" to give a syntax error.
3. Prevent rubout from aborting reads or writes.
Also tidied up the handling of interupts in general.

D 1.13  80/08/28 12:56:54       dave    14      13      00007/00002/02639
o/string/ where string not found gobbled up next command.

D 1.12  80/08/06 14:34:02       dave    13      12      00001/00000/02640
Clear 'delaywrite' in error - initial read will leave it set if error!

D 1.11  80/08/04 18:10:19       dave    12      11      00012/00011/02628
Tell 'getline' what to use for a buffer !!!
Too many routines assume it uses 'linebuf' ...

D 1.10  80/08/04 13:57:07       dave    11      10      00017/00018/02622
Allow auto-write during open mode by allocating separate buffers.
The use of 'genbuf' & 'linebuf' is a little enthusiastic.

D 1.9   80/06/30 10:41:16       dave    10      9       00001/00000/02639
The variable "argflag" must be cleared in two places!
Case 'r' may call error(), or it may not ...

D 1.8   80/06/24 14:06:02       dave    9       8       00001/00001/02638
Slight blunder - error() calls reset() which means that the
flag 'argflag' did not get cleared - meaning the input buffer did
not get flushed properly leaving the remnants of the command in it!
This gave rise to two "Buffer empty" messages ...

One day I'll clean up these delta's ...
```

```
D 1.7   80/06/18 14:16:59       dave    8       7       00001/00000/02638
Whoops - another patch to "ed file" mod.
This really needs to be cleaned up ...

D 1.6   80/06/18 13:24:04       dave    7       5       00008/00060/02630
1) Lose that silly "AUTOW" conditional compilation.
2) Delete some unused #define statements.
3) Fix error recovery on "ed file" - last mod was wrong.

D 1.5   80/06/17 19:14:36       dave    5       4       00030/00007/02660
Various fixes:
1) Initialise "globp" properly so don't lose next command on error.
2) Be more rigorous in checking files - check for plain type as well.
3) Dispense with "Are you sure" with quit after partial write.
4) Give syntax error on "!<CR>" command - user probably mistyped it.

D 1.4   80/06/11 14:37:21       dave    4       3       00011/00003/02656
Allow the writing of empty files i.e. truncate it.
This is needed for things like Marker files, dump procedures etc
which may encounter empty files and should do the right thing.

D 1.3   80/06/09 12:49:01       dave    3       2       00004/00003/02655
Print "Interrupt" when rubout hit in open mode.

D 1.2   80/05/26 11:24:28       dave    2       1       00002/00001/02656
Change LINES to 20 as per VT05 - they do exist you know!

D 1.1   80/05/01 16:44:18       dave    1       0       02657/00000/00000
#
```

And that completes my demonstration. Unfortunately it loses something in the translation, but at least my faith in electronic aids is now restored. In case you are wondering how I managed to transcribe the terminal output (and input) to this file, let me say only that it was achieved with the "connect" system call, a loop-back plug and a little surreptitious editing. I will be glad to supply the full details, should anybody be interested. Anyway, I am currently writing a tutorial on SCCS to cover all sorts of interesting features not shown here and will be published in a future edition of AUUGN.

EUUG

EUROPEAN UNIX USER GROUP

COMMITTEE
Chairman : Alan Mason, Heriot-Watt University
Editor : Bruce Anderson, University of Essex
Member(s) : Peter Collinson, University of Kent

R.A.Mason
Dept. Computer Engineering
Heriot-Watt University
Mountbatten Building
31-35 Grassmarket
Edinburgh EH1 2HT
(Tel. 031-225-8432 x 155)

Software Exchange

Dear Representative,

Pursuant with our software exchange agreement, I enclose a copy of our latest software release. This tape is a direct copy of that which we send out to our own members, and its format and extraction are described in the enclosed documentation. I also enclose an advert describing the package which you might like to use in your newsletter, with an obvious rider.

If you decide to distribute this package, I must insist that you are consistent with our own practise:

Non Profit Making; A reasonable charge being levied to cover costs (tape, post, packing).

Equal Favour; The package should be distributed in the same form, and for the same fee, to all members regardless of their class of membership.

Non Transference; Recipients should be constrained not to make and/or pass on to other installations copies of the package.

Licence Constraint; Where and when necessary the distribution centre should satisfy itself that the recipient has a valid licence for the package.

I apologise if this seems rather formal and restrictive, but, as most of the user groups are not of 'limited liability' and their office bearers are honorary, then at least the minimum necessary steps should be taken to ensure the integrity and non-culpability of the group. Assuming that you can agree to this outlined procedure, I will continue to redirect any particular software requests from your locale back to your group.

Although we cannot offer support for such packages, we would like to carry on the practice of reporting mods & fixes through the newsletter, and would therefore be grateful if you could filter back to us any that you or your members suggest. It would be of much assistance if, in doing this, you could indicate the source (person and installation) of such modifications.

Yours sincerely,

R.A. Mason

AUUGN - Heriot-Watt Software Exchange Agreement

..... (name or name of institution) agrees not to make and/or pass on to other installations copies of the Heriot-Watt University UNIX V7 Strip-down package.

We further certify that this installation holds a current UNIX V7 license a copy of which is attached.

Signed.....

for.....

date.....

EUUG

EUROPEAN UNIX USER GROUP

R.A.Mason
Dept. Computer Engineering
Heriot-Watt University
Mountbatten Building
31-35 Grassmarket
Edinburgh EH1 2HT
(Tel. 031-225-8432)

Distribution Software

Dear Correspondent,

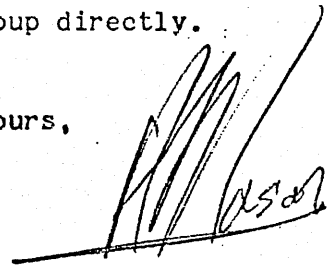
Please find enclosed the software distribution you requested plus documentation describing its format and how it should be extracted. Should you have any difficulty in installing the software please feel free to contact me.

If you should find any bugs/errors in the package or you have any suggested improvements/modifications, then we would greatly appreciate your feedback (preferably by letter!). Any such contact, along with changes we ourselves make would be fully reported in the group newsletter.

The group is continually looking for distributable software which would either form a complete package or, as a single utility, fit into a mixed distribution. A list of the current distribution material is attached.

In conclusion, I must point out that much distribution software is held under licence by its originator and although it may well be freely available, unauthorised copying and circulation of such software would be a licence violation. If you should be approached by another installation requesting software copies, please ask them to contact the group directly.

Yours,



R.A.Mason

EUUG SOFTWARE DISTRIBUTIONS

number	source	system	content
D1	gb.hwat.ee	Unix V7	UNIX V7 - small machine
D2	nl.vrij.inf	Unix V6/V7	PASCAL

EUUG.D1 - UNIX V7 SMALL MACHINE - SETUP

August 1980

Electrical and Electronic Engineering
Heriot-Watt University

This distribution tape is packaged for a DEC PDP-11/23/34/40/60 with RK05, RL01, or RL02 disks and with a TU10 (or equivelant) tape drive. It consists of some preliminary bootstrapping programs followed by a mixture of filesystem images and tape archives; if needed, after the initial construction of the file systems individual files can be extracted (see restor(1), tar(1)).

If you are set up to do it, it might be a good idea immediately to make a copy of the tape to guard against disaster. The tape is 9-track 800 BPI and contains some 512-byte records followed by many 10240-byte records. There are interspersed tapemarks.

The system as distributed contains binary images of the system and all the user level programs, along with source and manual sections for them-about 2100 files altogether. The binary images, along with other things needed to flesh out the file system enough so UNIX will run, are to be put on one file system called the `root file system'. The file system size required is 4000 blocks for an RK05 system, 9000 blocks for an RL01 system, and 18000 blocks for an RL02. (These sizes are smaller than the maximum available on the disk to allow some space for swapping. A non 'swap' disk would have a filesystem of 4872, 10240 and 20480 blocks respectively.) The remainder of the tape has all of the source and documentation.

This distribution is merely a repackaging of the WECo one, it doesn't contain any of the numerous alterations to the system and utilities which have been made, other than those necessary to make it fit on a small PDP-11 better and to accomodate the new tty handler. The only extras which are included are the `em' editor, which is a superset of the original 'ed', `cptree' (useful until you get the hang of the new stuff), `poke6', which allows you to investigate a V6 filesystem while running V7 (it's nothing fancy), and a simple disk copy program 'vcopy' which may be run standalone if required.

This guide is obviously a variation of **Setting Up Unix - Seventh Edition** by Charles Haley and Dennis Ritchie, which should also be read and understood fully before attempting anything, as common material is not repeated here.

1. Making a Disk From Tape

Perform the following bootstrap procedure to obtain a disk with a root file system on it.

1. Mount the magtape on drive 0 at load point.

2. Mount a formatted disk pack on drive 0.

3. Key in and execute at 100000

```
012700
172526
010040
012740
060003
000777
```

The tape should move and the CPU loop. (The TU10 code is not the DEC bulk ROM for tape; it reads block 0, not block 1.)

4. Halt and restart the CPU at 0.

5. The console should type

```
Boot
:
```

Copy the magtape to disk by the following procedure. The machine's printouts are shown in italics or are underlined, explanatory comments are within ( ). Terminate each line you type by carriage return or line-feed. The name `tm' is used for the TU10. There are two classes of disks: `rk' is used for the RK05, and `rl' is used for the RL01 (the shorthand r[kl] will be used to mean whichever is appropriate for you).

If you should make a mistake while typing, the character '#' erases the last character typed up to the beginning of the line, and the character '@' erases the entire line typed. Alternatively to match the new teletype handler, delete (rubout) may be used for single character deletions and control U (CTRLU) for complete lines. Since these in fact echo as '#' and '@', a retype line facility (CTRLR) has also been inserted to remove any doubts about what you have typed. Some consoles cannot print lower case letters, adjust the instructions accordingly.

```
(bring in the program mkfs)
:tm(0,3)
file system size: 4000  (9000 for RL01,18000 for RL02)
file system: r[kl](0,0)
isize = XX
m/n = XX
(after a while)
exit called
Boot
:
```

This step makes an empty file system.

6. The next thing to do is to restore the data onto the new empty file system. To do this you respond to the ':' printed in the last step with

```
(bring in the program restor)
:tm(0,4)
tape? tm(0,5)
disk? r[kl](0,0)
Last chance before scribbling on disk. (you type return)
(the tape moves, perhaps 5-10 minutes pass)
end of tape
Boot
:
```

You now have a UNIX root file system.

2. Booting UNIX

You probably have the bootstrap running, left over from the last step above; if not, repeat the boot process (step 3) again. Then use one of the following:

```
:rk(0,0)rkunix
:rl(0,0)rl1unix
:rl(0,0)rl2unix
```

The machine should type the following:

```
mem = xxx
login:
```

The mem message gives the memory available to user programs in bytes.

UNIX is now running, and the `UNIX Programmer's manual' applies; references below of the form X(Y) mean the subsection named X in section Y of the manual. The system is now running single-user and since their are few user names installed, you will have to login as the super-user The user name of the super-user is `root', and initially he has no password. You are strongly advised to rectify this (passwd (1)) before opening the system to mortals. The same goes for the 'bin' user name, which also gives considerable power!

At this time the system assumes that you are on a DECWRITER I and sets modes (upper case and CR/NL delays) appropriately. If this is not the case then it will eventually have to be changed in getty (1), for the moment you may use stty (1) to temporarily adjust it.

To simplify your life later, copy the appropriate version of the system as specified above plain 'unix', keeping a copy of the distributed binary to boot if something goes wrong. For example, use cp (1) as follows if you have an RK05:

```
cp rkunix unix
```

In the future, when you reboot, you can type just

```
rk(0,0)unix
```

to the `:' prompt. The 'current' system should always be known as '/unix' since certain utility programs (e.g. ps (1)) expect this and reference that file.

You now need to check the special file entries in the dev directory. These specify what sort of disk you are running on, what sort of tape drive you have, and where the file systems are. The file `r[kl]0' refers to the root file system; `swap' to the swap-space file system; `r[kl]1' to the user file system. The devices `rr[kl]0' and `rr[kl]1' are the `raw' versions of the disks. Also, `mt0' is tape drive 0; `rmt0' is the raw tape, on which large records can be read and written; `nrmt0' is raw tape with the quirk that it does not rewind on close, which is a subterfuge that permits multifile tapes to be handled. The file `swap' should be linked (ln(1)) to the appropriate root disk, rk0 or rl0:

```
ln r[kl]0 swap
```

The next thing to do is to extract the rest of the data from the tape. How this is done depends on whether you have RK05 or RL01 disks - you will need 6 RK05 packs, 3 RL01 packs or 2 RL02 packs, including the root disk already used - and how many drives you have. The contents of the tape follows, along with examples of how to extract it on to a variety of (small) disks. The examples assume only 2 drives are available, you may be able to speed up this process if you have more. Again, explanatory comments are shown in round brackets, so don't type these. For clarity, the output generated by these commands has been ommitted, when an error occurs you'll know!! The multiplicity of 'do-nothing' dd's are required because tar (1) knows the exact size of the current file and never actually reads the end-of file, so you have got to do it for it.

```
File 1:
    mtboot      -  magtape bootstrap (2 copies)
    boot        - The standalone bootstrap
File 2:
    cat     - A file to console copy program
File 3:
    contents    - This list
File 4:
    mkfs        - standalone make file system
File 5:
    restor      - standalone filesystem restore
File 6:
    /       - dump to get started with
File 7:
    /usr        - tar(1) format
File 8:
    src/cmd/[a-m]*  - tar(1) format
File 9:
    src/cmd/[n-z]*  - tar(1) format
    src/[d-z]*  - tar(1) format
File 10:
    man/docs    - tar(1) format
File 11:
    man/man[0-8]    - tar(1) format
```

RK05 system

```
(load a new pack in drive 1 for /usr)
# dd if=/dev/nrmt0 of=/dev/null files=6(skip records already processed)
# /etc/mkfs /dev/rrk1 4872 3 24
# /etc/mount /dev/rk1 /mnt
# cd /mnt
# tar xbf 20 /dev/nrmt0
# cd /
# /etc/umount /dev/rk1

(load a new pack in drive 1 for first half of /usr/src)
# dd if=/dev/nrmt0 of=/dev/null files=1(skip over tape mark)
# /etc/mkfs /dev/rrk1 4872 3 24
# /etc/mount /dev/rk1 /mnt
# cd /mnt
# tar xbf 20 /dev/nrmt0
# cd /
# /etc/umount /dev/rk1

(load a new pack in drive 1 for remainder of /usr/src)
# dd if=/dev/nrmt0 of=/dev/null files=1(skip over tape mark)
# /etc/mkfs /dev/rrk1 4872 3 24
# /etc/mount /dev/rk1 /mnt
# cd /mnt
# tar xbf 20 /dev/nrmt0
# cd /
# /etc/umount /dev/rk1

(load a new pack in drive 1 for first half of /usr/man)
# dd if=/dev/nrmt0 of=/dev/null files=1(skip over tape mark)
# /etc/mkfs /dev/rrk1 4872 3 24
# /etc/mount /dev/rk1 /mnt
# cd /mnt
# tar xbf 20 /dev/nrmt0
# cd /
# /etc/umount /dev/rk1

(load a new pack in drive 1 for remainder of /usr/man)
# dd if=/dev/nrmt0 of=/dev/null files=1(skip over tape mark)
# /etc/mkfs /dev/rrk1 4872 3 24
# /etc/mount /dev/rk1 /mnt
# cd /mnt
# tar xbf 20 /dev/nrmt0
# cd /
# /etc/umount /dev/rk1

# dd if=/dev/rmt0 of=/dev/null files=1(this will rewind the tape)
```

RL01 system

(/usr goes on rl0)

```
# dd if=/dev/nrmt0 of=/dev/null files=6(skip records already processed)
# cd /usr
# tar xbf 20 /dev/nrmt0
# cd /
```

(load a new pack in drive 1 for /usr/src)

```
# dd if=/dev/nrmt0 of=/dev/null files=1(skip over tape mark)
# /etc/mkfs /dev/rrl1 10240 8 40
# /etc/mount /dev/rl1 /usr/src
# cd /usr/src
# tar xbf 20 /dev/nrmt0
# dd if=/dev/nrmt0 of=/dev/null files=1(skip over tape mark)
# tar xbf 20 /dev/nrmt0
# cd /
# /etc/umount /dev/rl1
```

(load a new pack in drive 1 for /usr/man)

```
# dd if=/dev/nrmt0 of=/dev/null files=1(skip over tape mark)
# /etc/mkfs /dev/rrl1 10240 8 40
# /etc/mount /dev/rl1 /usr/man
# cd /usr/man
# tar xbf 20 /dev/nrmt0
# dd if=/dev/nrmt0 of=/dev/null files=1(skip over tape mark)
# tar xbf 20 /dev/nrmt0
# cd /
# /etc/umount /dev/rl1
```

```
# dd if=/dev/rmt0 of=/dev/null files=1(this will rewind the tape)
```

RL02 system

```
(/usr goes on rl0)
# dd if=/dev/nrmt0 of=/dev/null files=6(skip records already processed)
# cd /usr
# tar xbf 20 /dev/nrmt0
# cd /

(load a new pack in drive 1 for /usr/src)
# dd if=/dev/nrmt0 of=/dev/null files=1(skip over tape mark)
# /etc/mkfs /dev/rrl1 20480 8 40
# /etc/mount /dev/rl1 /usr/src
# cd /usr/src
# tar xbf 20 /dev/nrmt0
# dd if=/dev/nrmt0 of=/dev/null files=1 (skip over tape mark)
# tar xbf 20 /dev/nrmt0
# cd /

(/usr/man can go on rl0)
# dd if=/dev/nrmt0 of=/dev/null files=1 (skip over tape mark)
# cd /usr/man
# tar xbf 20 /dev/nrmt0
# dd if=/dev/nrmt0 of=/dev/null files=1 (skip over tape mark)
# tar xbf 20 /dev/nrmt0

# dd if=/dev/rmt0 of=/dev/null files=1(this will rewind the tape)
```

The operations that follow use files in subdirectories of '/usr'. On an RL system these will be on the system drive (rl0). For an RK system they must be brought on-line:

```
/etc/mount /dev/rl1 /usr
```

Before anything further is done the bootstrap block on the disk (block 0) should be filled in. This is done using the commands:

```
cd /usr/mdec
make r[kl]uboot
dd if=r[kl]uboot of=/dev/r[kl]0 count=1
```

Now the DEC disk bootstraps are usable. See Boot Procedures(8) for further information.

Before UNIX is turned up completely, a few configuration dependent exercises must be performed. By this point, it would be wise to have read all of the manuals (especially `Regenerating System Software').

3. Reconfiguration

The UNIX system running is configured to run on a PDP-11 without separate I/D space and with the given disk tape combination, a console, and no other device. This is certainly not the correct configuration. You will have to correct the configuration table (/usr/sys/conf/r[kl]conf) to reflect the true state of your machine.

It is wise at this point to know how to recompile the system. Print (cat(1)) the files /usr/sys/conf/READ_ME, /usr/sys/conf/m.h and /usr/sys/conf/makefile. The READ_ME file and the m.h file contain reconfiguration information to enable the system to run on the small PDP-11's; you may have to edit m.h for your particular machine - follow the instructions therein and in the READ_ME file. The makefile is input to the program `make(1)' which if invoked with `make all' will recompile all of the system source and install it in the correct libraries.

The program mkconf(1) prepares files that describe a given configuration (See mkconf(1)). In the /usr/sys/conf directory, the files rkconf, rl1conf, rl2conf were input to mkconf to produce the versions of the system that reside in the root i.e. rkunix, rl1unix, rl2unix. Pick the appropriate one, and edit it to add lines describing your own configuration. (Remember the console typewriter is automatically included; don't count it in the kl specification.) Then run mkconf; it will generate the files l.s (trap vectors) and c.c (configuration table). Take a careful look at l.s to make sure that all the devices that you have are assembled in the correct interrupt vectors. If your configuration is non-standard, you will have to modify l.s to fit your configuration.

There are certain magic numbers and configuration parameters imbedded in various device drivers that you may want to change. The device addresses of each device are defined in each driver. In case you have any non-standard device addresses, just change the address and recompile. (The device drivers are in the directory /usr/sys/dev.)

Similarly, the quantity of each device type is held in the driver and should be checked.

dc.c - The DC11 driver is set to run 4 lines.
dh.c - The DH11 driver is set to handle 1 DH11 with 16 lines.
dn.c - The DN11 driver will handle 4 DN's.
du.c - The DU11 driver can only handle a single DU.
This cannot be easily changed.
kl.c - The KL/DL driver is set up to run a single DL11a-A/B/C (the console), and no DL11-E's.
NKL11 reflects the number of DL11-A/B/C's.
NDL11 reflects the number of DL11-E's.
So far as the driver is concerned, the difference between the devices is their address.
dz.c - The DZ11 driver is set up for one 8-line device.

The block device drivers (rf.c, rk.c, rl.c, rp.c, tm.c, tc.c, hp.c, ht.c) are set up to run a reasonable number of units and should not need to be changed. The big disk drivers (rp.c, hp.c) have partition tables in them which you may want to experiment with.

There is also an optimised RK05 driver (rk.boston), and a System Industries/CDC 9762 SMD driver (si.c - only one drive).

After all the corrections have been made, use `make(1)' to recompile the system (or recompile individually if you wish: use the makefile as a guide). If you compiled individually, say `make unix' in the directory /usr/sys/conf. The final object file (unix) should be moved to the root, and then booted to try it out. It is best to name it /nunix so as not to destroy the working system until you're sure it does work. See Boot Procedures(8) for a discussion of booting. Note: before taking the system down, always (!!) perform a sync(1) to force delayed output to the disk.

4. Floating Point

UNIX only supports (and really expects to have) the FP11-B/C floating point unit. For machines without this hardware, there is a user subroutine available that will catch illegal instruction traps and interpret floating point operations. (See fptrap(3).) To install this subroutine in the library, change to /usr/src/libfpsim and execute the shell files

```
compall
mklib
```

The system as delivered has this code included in any command which needs it, and the operating system adapts automatically to the presence or absence of the FP11, unless you are using the `m34.c' system.

To compile floating point programs, when you have no floating point hardware (or firmware) use the `-f' flag to cc(1). This flag ensures that the floating point interpreter is loaded with the program and that the floating point version of `cc' is used.

5. Disk Layout

If there are to be more file systems mounted than just the root and /usr, use mkfs(1) to create any new file system and put their mounting in the file /etc/rc (see init(8) and mount(1)). (You might look at /etc/rc anyway to see what has been provided for you.)

There are two considerations in deciding how to adjust the arrangement of things on your disks: the most important is making sure there is adequate space for what is required; secondarily, throughput should be maximized. Swap space is a critical parameter. The system as distributed has 872 (rkunix), 1240 (rl1unix) or 2480 (rl2unix) blocks for swap space. This should be large enough so running out of swap space never occurs on the RL's, but the RK05 might run into trouble. You may want to change these if local wisdom indicates otherwise.

Many common system programs (C, the editor, the assembler etc.) create intermediate files in the /tmp directory, so the file system where this is stored also should be made large enough to accommodate most high-water marks. If you leave the root file system as distributed (except as discussed above) there should be no problem on RL's, again, things will be tight for RK05 systems. All the programs that create files in /tmp take care to delete them, but most are not immune to events like being hung up upon, and can leave dregs. The directory should be examined every so often and the old files deleted (e.g. at boot time in /etc/rc).

6. Odds and Ends

Appearing (in alphabetical order):

Hugh Conner
Alan Mason
Jim McKie
Zdravko Podolski
Colin Prosser
Dave Rosenthal

EUUG.D1 - UNIX V7 SMALL MACHINE - NOTES

September 1980

Computer Engineering
Heriot-Watt University

1) It has been pointed out that the ROM bootstrap on PDP-11/23's, whilst loading to location 0 does not execute from there! If this causes problems, simply HALT and start from 0. You may actually get something of the tape then!

2) Once the tape has been dumped to your chosen disks, the next step is to make a system which truly matches your configuration. This requires you to edit the configuration file (/usr/sys/conf/r[kl]conf) to reflect your available devices. Before you do this ensure that the date (/etc/dateset) is set correctly or 'make' will get confused. You should also link the file (conf) to whichever configuration file you would normally use. Note that although the system will generally treat RL01/RL02 disks identically (i.e. as /dev/rl), a different configuration file is required (rl1conf or rl2conf) as the system drive will have swap space in a different place.

3) If you decide to make an XBUF system i.e. a system with the buffers ported out from the kernel, then for safety recompile and reinstall everything. Turning on XBUF in '/usr/sys/h/param.h' may effect code in files you might not otherwise compile.

4) The tty's (/etc/ttys) file will have to be edited to reflect your terminals and their types, as well as those types documented in 'getty' (1) a number of additions have been made:

b	bantam	(9600 baud)
d	dacoll	(4800 baud)
t	tektronix 401?	(9600 baud)
s	satellite computer	(9600 baud)

Further you may have to adjust the type of your console as a Decwriter I is assumed.

5) The tty devices (/dev/tty?) are not made as the system is distributed. Change in to /dev and adjust the makefile according to the devices you have and their device numbers. Device numbers (major) are to be found in the conf directory (/usr/sys/conf/c.c)

6) The device driver supplied for si.c has built in to it a number of possible logical device configurations, which are

selectable by setting two 'defines'.

Firstly it may be used with a 'flipped' or 'unflipped' system. The flipped system, actually treats certain logical disks as if their last cylinder were first and vice-versa. This means that the 'first' cylinders of two adjacent logical disks are back-to-back and is more efficient in terms of head travel. There are problems, however, in that such areas of the disk may only be accessed through a UNIX, and not for example through the boot software, or standalone utilities etc. Only use this feature if you know what you are doing (it is not on by default!).

Secondly the tail end of the disk is setup to hold a number of images of the same size as standard DEC small disks, RK05 (default), RL01 or RL02. These are there to facilitate copying and/or development. Select that which best suits your needs.

7) Versions of UNIX on the distribution tape have been made using a simple machine support (m40.s) which uses minimal facilities of the machine (does not expect I/D space). So that these may be used, to get started on large (11/44,45,70) machines the boot software (in '/usr/src/cmd/stand') has had to be modified so as to not setup the feature.

To explain, the boot run time start off files (M.s, srt0.s) attempt to be intelligent and test to see if they are being used on a separate I/D machine. If this is the case the facility will be setup and thus used. This is alright if the software is expecting to be used in that environment, but as has been explained, this is not the case with the distribution UNIX's. The lines dealing with this testing have had to be commented out, so that the boot program always thinks it is on a small machine:

M.s (lines 28-30)

```
/   tst *$KDSA6      / Test for separate inst & data
/   mov $KDSA6,_ka6  / Point dummy appropriately
/   inc _sep         / Set the global flag
```

srt0.s (lines 44-45)

```
/   tst *$KDSA6      / Test for separate inst & data
/   mov $KDSA6,_ka6  / point dummy appropriately
```

In consequence, if you are using a large machine, then the first time you make a system which truly reflects that machine you will have to de-comment these lines and remake the boot software:

```
make xcp
```

Note, that from this point you will not be able to boot the original distribution system using your new 'boot' program. You will, however be able to boot it directly if you are stuck by simply typing its name in place of typing 'boot':

Instead of (for example):

```
@boot
BOOT
:rk(0,0)rkunix
```

simply type:

```
@rkunix
```

This shortcut can only be safely used for the distribution versions of UNIX because they use no special features, are built using m40.s and are particularly small. In general always use 'boot'!

EUUG.D1 - UNIX V7 SMALL MACHINE - FIXES

September 1980

Computer Engineering
Heriot-Watt University

The fixes noted here are continually being collected and the distribution updated accordingly. Release marks (down side of page) show the various points at which new tapes have been made. All fixes up to that point have already been made and thus will be on the released tape. Fixes are made and recorded in the order given here, and thus will assume that all other (pertinent) previous fixes have been made. This is particularly important when line numbers are referred to!! Fixes are noted with reference to the last release level and line numbers etc. may not correspond between releases.

RELEASE 1 ->

/etc/rc

gb.edin.mi — Alen Shapiro

The references to '/dev/tty' in line 14 should be changed to '/dev/console'. This is required as, at boot time there is no controlling teletype.

/usr/include/sys/param.h

gb.hwat.ee — Alan Mason

The reference to '/usr/sys/conf/local.h' on line 145 should be changed to '/usr/sys/conf/m.h'. It should be noted that the files in '/usr/include/sys' and '/usr/sys/h' are not direct copies of each other (nor indeed are they links, which would be more desirable!) and that as you make changes in one set you may have to adjust the other set appropriately.

PDP-11/60

gb.edin.arc — John Hannah

Attempts to remake a system on the 11/60 will fail since the backup installed is that of an 11/40. Only solution is to go to convenient non 11/60 site with your boot disk and make a system with 11/60 backup.

/usr/sys/conf/m34.c

gb.edin.arc — Dave Rosenthal

It transpires that Edinburgh's 'backup' is inadequate for 11/34's with FP11A floating point options. The test programs distributed were unable to detect this. The remedy is to change your backup code near the label 'fp60' to read:

```
fp54:  / stcfi
fp70:  / ldcif
       stfps       r0          / if long integer mode
       bit         $100,r0
       bne         1f          / its 4 bytes
#ifdef BUP_M2D
       mov         r1,r0       / on a /34
       mov         $setreg,pc  / its different
#else
       mov         $u5,pc      / else its really 2 bytes
#endif

fp60:  / stcfd
fp74:  / ldcfd
       incb        bflg        / assume 4 bytes
       stfps       r0
       tstb        r0          / if floating double
       bmi         0f          / its really is
       br          1f          / else its 8 bytes
```

Some people have been confused by the output during these tests. Unless it specificially announces that a test has failed, by for example:

i2.s: fails

then the test has succeeded.

/usr/src/cmd/ps.c

gb.hwat.ee Alan Mason

As distributed the version of 'ps' to be found in '/bin' is from the original Bell 'ps.c' which can be found in '/usr/src/cmd/original/ps.c' and does not match the system. Note that ps uses system header files and may need to be recompiled when you make changes and remake the system. A newer and more informative 'ps' is available in '/usr/src/cmd/ps.c'. It is suggested that you 'mv /bin/ps /bin/ops' and compile and install the new one as 'ps'. The new ps contains a list of useful name list entries (nlist) which may need to be tuned to your system e.g. the addition of:

rrlbuf
rrpbuf
rrmbuf
etc.

The new ps also uses 3 files in '/tmp' which it creates the first time it is called, making succesive calls faster.

/usr/sys/dev/kl.c

gb.hwat.ee Hugh Conner

Stopping and starting of output will not work on a KL11/DL11 as a test is missing in the driver. The fix is in the routine 'klstart' line 140 and this line should be changed for

```
if(((addr->tcsr&DONE) == 0)
|| (tp->t_state&TTSTOP)
||(tp->t_xstate&XPAGE1))
```

/usr/sys/dev/tty.c

gb.hwat.ee Jim McKie

Certain terminals (e.g. some Decwriters) insist upon putting MARK parity on input. Parity however is only stripped after certain testing is done in the current driver and thus '^q' and '^s' will not work from these devices. The required changes are in the routine ttyinput, firstly delete line 432 which reads:

```
c &= 0177;
```

and then append the following code after line 406:

```
if((tp->t_flags&RAW) == 0)
    c &= 0177;
```

/usr/sys/conf/makefile

gb.edin.arc Dave Rosenthal

The supplied makefile is deficient in that it does not have the dependancy of 'c.c' upon certain of the header files. This dependancy is easiest added to give the line:

```
c.o :   m.h  ../h/*.h
```

/usr/src/cmd/as

gb.edin.arc Dave Rosenthal

Another deficient makefile, this time for the assembler! Certain references are incorrectly made to the current assembler 'as' when making a new one. The makefile should be changed so as to refer to 'as1':

```
all:    as1 as2

cmp:    as1 as2
    cmp as1 /bin/as
    cmp as2 /lib/as2
    rm as1 as2 a.out

cp: as1 as2
    cp as1 /bin/as
    cp as2 /lib/as2
    rm as1 as2 a.out

as1:
    as /usr/include/sys.s as1?.s
    ld -n -s a.out -o as1

as2:
    as /usr/include/sys.s as2?.s
    ld -n -s a.out -o as2
```

/usr/src/cmd/as

gb.edin.arc Dave Rosenthal

The instruction 'stst' has been omitted from the assembler and should be inserted. This must be done in two places, firstly in 'as19.s' by inserting after line 213:

```
<stst\0\0\0\0>; 15;170300
```

and in 'as29.s' by inserting after line 180:

```
15;170300   /stst
```

/usr/sys/dev/tty.c

gb.hwat.ee — Jim McKie

The lack of unsigned chars in the distributed PDP-11 C compiler can lead to certain funnies in the teletype driver. Firstly add the definition:

```
#define ubyte(c) ((c)&0377)
```

after line 75 along with the other macro definitions. Secondly references (or assignments) using character variables as counters should be protected. In particular the following lines (some of which are just tidying up!) should be changed to:

```
495: if(ubyte(tp->t_col) > 0)
575: width = tp->t_width ? ubyte(tp->t_width): 0377;
579: if(ubyte(*cp++) == 0377)
686: n = ubyte(tp->t_col) - max(n, ubyte(tp->t_htdly));
687: else if((n = ubyte(tp->t_col) - ubyte(tp->t_htdly)) == 0)
689: if(ubyte(tp->t_htdly) >= (ubyte(tp->t_col) - n))
727: if(((cp = q->c_cl) == NULL) || ((c = ubyte(*--cp)) == 0377)){
845: if((c == '\n' && (ubyte(++(tp->t_lnum)) >= ubyte(tp->t_length))) ||...
893: if(tp->t_width && (ubyte(*colp) >= ubyte(tp->t_width)))
997: if((c != 0377) && ubyte(*(tp->t_rawq.c_cf)) == 0377)
```

/usr/sys/dev/rl.c

gb.edin.ee — Alan Mason

The rl driver supplied will only handle RL01 disk drives. The upgraded/reworked driver will test to see what type of rl's (RL01,RL02) are on and handle them appropriately. It should be possible, though this has never been tested, to mix and match RL01/2 drives on the same controller. Changes are however too extensive to document here and separate contact should be made for a copy of the new driver.

/usr/sys/dev/tty.c

gb.hwat.ee — Jim Mckie

Software tab expansion can in some cases lose count of the screen column number. This occurs because of a short-cut taken to speed up the output of the spaces. To correct the situation lines 887-889 should be changed to read:

```
do
    ttyoutput(' ', tp);
while((*colp)&07);
```

This means that the routine 'ttyoutput' calls itself recursively, but at least its right!

/usr/sys/dev/tty.c

gb.hwat.ee — Jim McKie

Tab deletions when a line is longer than a screen width (i.e. display has wrapped around) can cause the system to loop, depending on your processor, for a number of minutes (1-3) while a negative number is decremented back to zero, all the while transmitting to the terminal. The fix is to replace line 700:

```
while(n--) {
```

with the code:

```
if((n < 0) || (n > ubyte(tp->t_col))) {
    /* if(tp->t_col) */
        ttyoutput('\n',tp);
    ttyretype(tp,1);
}
else
while(n--) {
```

The line shown as being commented out may by choice be commented in to give a generally better presentation. If present this line will ensure that a newline is not taken if the cursor is at position zero (i.e. at the beginning of the line already). This will fail under two conditions, firstly if the terminal is not set up to wrap text around or secondly if people are typing backspaces in their input stream.

/usr/src/cmd/init.c (/etc/init)

gb.hwat.ee Alan Mason

The 'shell' executed in 'init' which allows single user work, also leaves a great breach in system security. The simples fix is to add after line 18:

```
char login[] = "/bin/login";
```

and to precede line 89 which executes a shell with:

```
printf("single user ");
execl(login,login,(char *)0);
```

This ensures at least that single users can do no more than their permissions allow. The printf is used to emphasise (distinguish) this 'login' from other multi user logins. The shell call is left in place lest due to file system corruption or, for other reasons, login may not be executed. N.B. In this case it becomes even more important that ordinary users may not fiddle the permissions on login to make it inaccessable. An end-of-file (CTRL Z) typed in response to this request will cause the system to come up multi-user.

A better solution is to rework this area so as to offer either single or multi user as an option, either in init itself or in the start-up procedure ('/etc/rc'). This has been done in a number of forms, but it is not yet clear which is best!

/usr/src/cmd/init.c (/etc/init)

gb.hwat.ee Alan Mason

The facility to bring/take terminals on/off line at runtime is not present in the supplied 'init'. This may be done by using the software termination signal (SIGTERM) and using it to activate the 'merge' subroutine. This is done by ammending the declaration on line 41 to read:

```
int reset(),merge();
```

and inserting after lines 44 and then 209 the statement:

```
signal(SIGTERM,merge);
```

Line 208 which reads:

```
close(create(utmp,0644));
```

should then be removed from the 'merge' routine and moved to after line 49 so that it is called only once. Finally the 'return' statement in line 124 should be changed to a 'contin-

ue'. Note that the version 6 mechanism for (de)activating terminals ('kill -1 1') now re-starts (not re-boots) the system and that the new mechanism is simply 'kill 1'.

/usr/sys/sys/sig.c

gb.edin.arc Dave Rosenthal

The routine grow in this file has a slight correction. Line 246 should read:

```
si = btoc(-sp) - u.u_size + SINCR;
```

This is unlikely to have caused you problems!

/stand

gb.hwat.ee Alan Mason

Compiled versions of the standalone utilities have not been provided. The makefile for these resides in '/usr/src/cmd/stand' and they should be compiled and installed in the directory '/stand' lest you suddenly require them. A standalone utility 'xxx' then be accessed at boot time in place of a unix by typing:

```
r[kl](0,0)stand/xxx
```

The standalone rl driver, as noted with the normal system one above has been reworked to handle RL02's. At the same time a standalone volume copy ('vcopy') program has now been written.

/usr/sys/sys/sys3.c

gb.hwat.ee Hugh Conner

The mount system call ('smount') will overwrite any valid errors generated by the open call to a device driver, substituting the general error 'EBUSY'. To correct this line 183 should be changed to read:

```
goto out1;
```

This will only have the desired effect (meaninful error messages from mount) if the drivers in use actually set valid errors.

/usr/sys/dev

gb.hwat.ee Alan Mason

As noted in the previous fix (to smount system call) device drivers do not as yet return many meaningful errors. The drivers rk.c, rl.c and tm.c have been corrected to at least give errors if the device is not on line or if an open for writing is attempted on a read-only filesystem. Previously errors would only occur when you actually accessed the device, thus defeating anyone who tried to write reasonably intelligent programs. This involves creating open & close routines (instead of using the default 'nulldev') for these devices and carrying out the appropriate tests. The names of these routines must then be inserted in the appropriate places in the file '/usr/sys/conf/mkconf.c'.

/usr/sys/dev/kl.c

gb.newc.cl Pete Lee

The kl/dl driver incorporates a fix for a hardware bug in early KL/DL11's. In effect if it recieves a null it will re-

transmit it. This is alright as long as you are not wanting to use the line for binary data from another machine. The worst situation is if two kl's are used back to back, leading to both devices transmitting nulls to each other ad infinitum. If you are sure of your device, and you need the facility simply comment out line 120 so that lines 120 and 121 read:

```
        /* if((c&0177) == 0)
            addr->tbuf = c; /* hardware botch (er....?) */
```

This code will not be changed in the distribution in case it gives you problems with the console device.

RELEASE 2 ->

January 1981

UNIX ™ and C Bibliography

This bibliography contains 181 references to published documents on the UNIX operating system and the C programming language. It consists of two input sources. References dated 1978 and later were obtained from a computerized search of the INSPEC and NTIS databases and from printed indexes on computing. Earlier references were selected from *BTL Software: The Published Record*, issued by the Libraries & Information Systems Center in June, 1978.

The citations are arranged by author within the general class, UNIX or C Language. Permuted title and author-title indexes follow the bibliographic listing.

The assistance of B.A. Stevens in the preparation of this bibliography is greatly appreciated. Any comments on this document may be directed to Martha Broad, MH x5674.

0001 PROCEEDINGS OF THE DIGITAL EQUIPMENT USERS SOCIETY.
PROC DIG EQUIP USERS SOC 2(4): (1977)

0002 NETWORKING AND THE PROCESS STRUCTURE OF UNIX: A CASE STUDY.
BALOCCA R
P306-11 OF PROC OF COMPCON FALL '78 COMPUTER COMMUNICATIONS NETWORKS, WASHINGTON, DC, USA, 5-8 SEPT 1978, IEEE NEW YORK, USA, 1978
DEPT OF COMPUTER SCI, UNIV OF ILLINOIS, URBANA-CHAMPAIGN, IL, USA
SAC 1979: 001747

0003 UNIX WITH SATELLITE PROCESSORS.
BARAK AB + SHAPIR A
SOFTWARE-PRACT EXPER 10(5): 383-92 (MAY 1980)
DEPT OF COMPUTER SCI, HEBREW UNIV, JERUSALEM, ISRAEL
SAC 1980: 024451

0004 UNIX TIME-SHARING SYSTEM: UNIX PROGRAMMER'S MANUAL.
BELL LABORATORIES, 7TH EDITION, 1979.
CONSISTS OF THREE VOLUMES: VOL. 1 CONTAINS DESCRIPTIONS OF PUBLICLY-AVAILABLE FEATURES OF UNIX; VOL. 2A & 2B CONTAIN DOCUMENTS FOR USE WITH THE SYSTEM.

0005 UNIX/32V TIME-SHARING SYSTEM: UNIX PROGRAMMER'S MANUAL.
BELL LABORATORIES, VERSION 1.0, 1979.

0006 A USER'S VIEWPOINT ON THE PROGRAMMER'S WORKBENCH.
BIANCHI MH + WOOD JL
P193-199 OF PROC IEEE NATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 2ND, 1976, SAN FRANCISCO
BELL LABORATORIES

0007 A SHAPE ORIENTED SYSTEM FOR AUTOMATED HOLTER ECG ANALYSIS.
BIRMAN KP + ROLNITZKY LM + BIGGER JT
P217-20 OF COMPUTERS IN CARDIOLOGY 1978, STANFORD, CA, USA 12-14 SEPT 1978, IEEE, NEW YORK, USA, 1978
COLL OF PHYSICIANS AND SURGEONS, COLUMBIA UNIV, NEW YORK, NY, USA
SAA 1979: 070662

0008 SOFTWARE DEVELOPMENT FOR TASK-ORIENTED MULTIPROCESSOR ARCHITECTURES.
BISIANI R + MAUERSBERG H
P20-7 OF COMPCON 79 PROC USING MICROPROCESSORS, EXTENDING OUR REACH WASHINGTON, DC, USA, 4-7 SEPT 1979, IEEE, NEW YORK, USA, 1979
COMPUTER SCI DEPT, CARNEGIE-MELLON UNIV, PITTSBURGH, PA, USA
SAC 1980: 005372

0009 AN INTERACTIVE STATISTICAL PROCESSOR FOR THE UNIX TIME-SHARING SYSTEM.
BLOOMFIELD P
P2-8 OF COMPUTER SCIENCE AND STATISTICS TENTH ANNUAL SYMP ON THE INTERFACE, GAITHERSBURG, MD, USA, 14-15 APRIL 1977, NAT BUR STANDARDS, WASHINGTON, DC, USA, 1978
DEPT OF STATISTICS, PRINCETON UNIV, PRINCETON, NJ, USA
SAC 1978: 025687

0010 THE INSTALLATION OF ALICE ON THE PDP 11/45 UNDER UNIX.
BOEHM APW
N80-25033/5, JAN 1978, 70P.
STICHTING MATHEMATISCH CENTRUM, AMSTERDAM, NETHERLANDS

0011 AN INTRODUCTORY COURSE IN THE APPLICATIONS OF COMPUTER TECHNOLOGY IN THE HEALTH SCIENCES.
BORDAGE G + LAKE RB
SIG CSE 8(3): 86-90 (SEP 1976)
CASE WESTERN RESERVE UNIV, SCHOOL OF MEDICINE, CLEVELAND, OH, USA

0012 THE MH MESSAGE HANDLING SYSTEM: USER'S MANUAL.
BORDEN BS + GAINES RS + SHAPIRO NZ
AD-A081 992/0, NOV 1979, 48P.
RAND CORP, SANTA MONICA, CA

0013 UNIX TIME-SHARING SYSTEM: THE UNIX SHELL.
BOURNE SR
BELL SYST TECH J 57(6): PT2 1971-90 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 031195

0014 COMBINED QUARTERLY TECHNICAL REPORT NO. 10: PACKET BROADCAST BY SATELLITE, PLURIBUS SATELLITE IMP DEVELOPMENT, UNIX SYSTEM DEVELOPMENT.
BRESSLET RD
AD-A059 297/2ST, AUG 1978, 53P.
BOLT, BERANEK & NEWMAN, CAMBRIDGE, MA

0015 SCHEDULING TECHNIQUES FOR OPERATING SYSTEMS.
BUNT RB
COMPUTER 9(10): 10-18 (OCT 1976)
SASKATCHEWAN UNIV, SASKATOON, SASKATCHEWAN, CANADA

0016 AN INTEGRATED APPROACH TO MICROCOMPUTER SUPPORT TOOLS.
CERMAK IA
P16/3/1-3 OF 1977 ELECTRO CONF RECORD, NEW YORK, USA 19-20 APRIL 1977, ELECTRO, EL SEGUNDO, CALIF, USA, 1977
BELL LABS, HOLMDEL, NJ, USA
SAC 1978: 009323

0017 DISTRIBUTED MEDICAL DATA-BASE: NETWORK SOFTWARE DESIGN.
CHANG E
P166-182 OF CANADIAN COMPUTER CONFERENCE, MONTREAL, MAY 17-19 1976
WATERLOO UNIV, ONTARIO, CANADA

0018 NETWORK UNIX SYSTEM.
CHESSON GL
OPER SYST REV 9(5): 60-6 (1975) (SPECIAL ISSUE)
ILLINOIS UNIV, URBANA, ILL
SAC 1976: 13988

0019 UNIX TIME-SHARING SYSTEM: THE NETWORK OPERATIONS CENTER SYSTEM.
COHEN H + KAUFELD JC
BELL SYST TECH J (USA) 57(6): PT2 2289-304 JULY-AUG 1978
SAC 1978: 031260

0020 MULTILEVEL SECURITY FOR INTELLIGENCE DATA PROCESSING SYSTEMS.
COTRELL J + SHU C + SHORT G
AD-A070 141/7ST, APR 1979, 227P.
TRW, REDONDO BEACH, CA

0021 PLOT: A UNIX PROGRAM FOR INCLUDING GRAPHICS IN DOCUMENTS.
CURTIS P
LBL-10690, APR 1980, 60P.
CALIFORNIA UNIV, BERKELEY, CA

0022 NUCLEAR PHYSICS DATA ACQUISITION WITH THE UNIX TIME-SHARING SYSTEM.
CUSTEAD LR + MCALPINE JL
IEEE TRANS NUCL SCI (USA) 26(1): 1949-51 FEB 1979
ACCELERATOR LAB, UNIV OF SASKATCHEWAN, SASKATOON, SASKATCHEWAN, CANADA
SAA 1979: 041085

0023 UNIX TIME-SHARING SYSTEM: THE PROGRAMMER'S WORKBENCH.
DOLOTTA TA + HAIGHT RC + MASHEY JR
BELL SYST TECH J 57(6): PT2 2177-200 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 029156

0024 LEAP LOAD AND TEST DRIVER.
DOLOTTA TA + LICWINKO JS + MENNINGER RE + ROOME WD
P182-186 OF PROC IEEE NATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 2ND, 1976, SAN FRANCISCO
BELL LABORATORIES

0025 INTRODUCTION TO THE PROGRAMMER'S WORKBENCH.
DOLOTTA TA + MASHEY JR
P164-168 OF PROC IEEE NATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 2ND, 1976, SAN FRANCISCO
BELL LABORATORIES

0026 USING A COMMAND LANGUAGE AS THE PRIMARY PROGRAMMING TOOL.
DOLOTTA TA + MASHEY JR
P35-55 OF COMMAND LANGUAGE DIRECTIONS: PROC OF IFIP TC 2.7 WORKING CONF ON COMMAND LANGUAGES, 10-14 SEPT 1979, BERCHTESGADEN, GERMANY, NORTH-HOLLAND PUB, AMSTERDAM, NETHERLANDS
BELL LABS, MURRAY HILL, NJ, USA

0027 AN ENVIRONMENT FOR PRODUCING WELL-ENGINEERED MICROCOMPUTER SOFTWARE.
EANES RS + HITCHON CK + THALL RM + BRACKETT JW
P386-98 OF PROC OF THE 4TH INT CONF ON SOFTWARE ENGINEERING, MUNICH, GERMANY, 17-19 SEPT 1979, IEEE, NEW YORK, USA, 1979
SOFTECH INC, WALTHAM, MA, USA
SAC 1980: 005606

0028 A LISP SHELL.
ELLIS JR
SIGPLAN NOT 15(5): 24-34 (MAY 1980)
COMPUTER SCI DEPT, YALE UNIV, NEW HAVEN, CT
SAC 1980: 024341

0029 THE DEVELOPMENT OF A PARTITIONED SEGMENTED MEMORY MANAGER FOR THE UNIX OPERATING SYSTEM.
EMERY RW
MASTER'S THESIS, JUN 1976, 91P, AD-A027 251/8ST.
NAVAL POSTGRADUATE SCHOOL, MONTEREY, CA

0030 MAKE-A PROGRAM FOR MAINTAINING COMPUTER PROGRAMS.
FELDMAN SI
SOFTWARE-PRACT EXPER 9(4): 255-65 (APRIL 1979)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1979: 016008

0031 COMPUTER SCIENCE AND TECHNOLOGY: COMMON COMMAND LANGUAGE FOR FILE MANIPULATION AND NETWORK JOB EXECUTION: AN EXAMPLE.
FITZGERALD, ML
PB-284 459/5ST, AUG 1978, 37P.
NATL BUR STANDARDS, WASH DC,
COMPUTER SYSTEMS ENGINEERING DIV

0032 MESS-A MACROLANGUAGE FOR STRUCTURED SYSTEMS PROGRAMMING.
FORGACS T + VAN DEN BOS J
ANGEW INF 20(1): 25-32 (JAN 1978)
INFORMATICA FACULTY OF SCI, UNIV OF NIJMEGEN, NIJMEGEN,
NETHERLANDS
SAC 1978: 009293

0033 OPERATING SYSTEMS IN SHARED TIME-THE UNIX PHENOMENON.
FOURTANIER J-L
AUTOM AND INF IND (FRANCE) (88): 37-41 (JUNE-JULY 1980) (IN FRENCH)

0034 UNIX TIME-SHARING SYSTEM: CIRCUIT DESIGN AIDS.
FRASER AG
BELL SYST TECH J 57(6): PT2 2233-49 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAB 1978: 049062

0035 WORD PROCESSING WITH UNIX.
GILLOGLY JJ
P22/2/1-3 OF 1978 MIDCON TECHNICAL PAPERS, DALLAS, TX, USA, 12-14 DEC 1978 WESTERN PERIODICALS CO, NORTH HOLLYWOOD, CA, USA, 1978
INTERACTIVE SYSTEMS CORP, SANTA MONICA, CA, USA
SAC 1979: 033969

0036 IMPLEMENTATION AND PERFORMANCE OF A UNIX LINK.
GREEN SL + ALEXANDER ST
P197-8 OF 1979 INT MICRO AND MINI COMPUTER CONF, HOUSTON TX, USA, 14-16 NOV 1979, IEEE, NEW YORK, USA, 1979
LOS ALAMOS SCI LAB,
LOS ALAMOS, NM, USA
SAC 1980: 012397

0037 DYNAMIC MICROPROGRAMMING IN A TIME SHARING ENVIRONMENT.
GUHA RK
P55-60 OF MICRO 10 PROC, NIAGARA FALLS, NY, USA, 5-7 OCT 1977 IEEE, NEW YORK, USA, 1977
COMPUTER SCI DEPT, SOUTHERN ILLINOIS UNIV, CARBONDALE, IL, USA
SAC 1978: 001489

0038 DESIGN OF A USER MICROPROGRAMMING SUPPORT SYSTEM.
GUHA RK + EBELING C
P446-50 OF 15TH IEEE COMPUTER SOCIETY INT CONF, WASHINGTON DC, USA, 6-9 SEPT 1977, IEEE, NEW YORK, USA, 1977
DEPT OF COMPUTER SCI, SOUTHERN ILLINOIS UNIV, CARBONDALE, IL, USA
SAC 1978: 001481

0039 DESIGN DESCRIPTION OF THE NOVA 3 CARTRIDGE DISK EMULATOR ON THE STANFORD EMMY SYSTEM.
HAFEMAN DR
SU-326-P.39-29, JUN 1978, 18P.
STANFORD UNIV, DIGITAL SYSTEMS LAB, STANFORD, CA

0040 USE OF SCIENTIFIC DATA WITH THE MASTER CONTROL AND INGRES DATA MANAGEMENT SYSTEMS.
HAMPEL VE + MCGROGAN K + SWANSON JE
UCRL-81160, MAY 1978, 41P. CONFERENCE ON ENGINEERING AND SCIENTIFIC DATA MANAGEMENT, HAMPTON, VA, USA, 18 MAY 1978
CALIFORNIA UNIV., LAWRENCE LIVERMORE LAB, LIVERMORE, CA

0041 A PORTABLE FILE DIRECTORY SYSTEM.
HANSON DR
SOFTWARE-PRACT AND EXPER 10(8): 623-34 (AUG 1980)
DEPT OF COMPUTER SCI, UNIV OF ARIZONA, TUCSON, AZ, USA

0042 HIGH SPEED DATA ACQUISITION: RUNNING A REALTIME PROCESS AND A TIME-SHARED SYSTEM (UNIX) CONCURRENTLY.
HARLAND DM
SOFTWARE-PRACT EXPER 10(4): 273-81 (APRIL 1980)
DEPT OF COMPUTER SCI, UNIV OF ST ANDREWS, ST ANDREWS, SCOTLAND
SAC 1980: 021723

0043 INTER-PROCESS COMMUNICATIONS FOR A SERVER IN UNIX.
HAVERTY JF + RETTBERG RD
P312-15 OF PROC OF COMPCON FALL '78 COMPUTER COMMUNICATIONS NETWORKS, WASHINGTON, DC, USA, 5-8 SEPT 1978, IEEE NEW YORK, USA, 1978
BOLT BERANEK AND NEWMAN, INC, CAMBRIDGE, MA, USA
SAC 1979: 001748

0044 MUNIX, A MULTIPROCESSING VERSION OF UNIX.
HAWLEY JA + MEYER WDB
MASTER'S THESIS, JUN 1975, 58P.
NAVAL POSTGRADUATE SCHOOL, MONTEREY, CA

0045 PDP 11 IMAGE PROCESSING SOFTWARE.
HAYES KC + HERMAN M + SMITH R
AD-A049 586/1ST, DEC 1977, 60P.
MARYLAND UNIV, COMPUTER SCI CENTER, COLLEGE PARK, MD

0046 STORAGE STRUCTURES AND ACCESS METHODS IN THE RELATIONAL DATA-BASE MANAGEMENT SYSTEM, INGRES.
HELD G + STONEBRAKER M
PROC ACM PACIFIC CONF, SAN FRANCISCO, APR 17-18, 1975
CALIFORNIA UNIV, ELECTRONICS RES LAB, BERKELEY, CA

0047 INGRES: A RELATIONAL DATA-BASE SYSTEM.
HELD G + STONEBRAKER M + WONG E
PROC 1975 NAT COMPUT CONF, ANAHEIM, CALIF, MAY 1975
CALIFORNIA UNIV, BERKELEY, CA

0048 GIML REFERENCE MANUAL.
HENNEGAN NM
UIUCDCS-R-77-857, UILU-ENG-77-1709. AVAIL FROM ERDA, POB 62, OAK RIDGE TENN, 37830. ATTN: TIC
ILLINOIS UNIV, URBANA, ILL

0049 RESOURCE SHARING UNIX.
HOLMGREN SF
P302-5 OF PROC OF COMPCON FALL '78, COMPUTER COMMUNICATIONS NETWORKS WASHINGTON, DC, USA, 5-8 SEPT 1978, IEEE, NEW YORK, USA, 1978
DIGITAL TECHNOL INC, CHAMPAIGN, IL, USA
SAC 1979: 001746

0050 USING PERSONAL COMPUTERS AS TERMINALS IN COMPUTER NETWORKS.
HORTON RE
P80-3 OF PROC OF 18TH AEDS ANNUAL CONV, 13-16 APRIL 1980, ST LOUIS, MO, AEDS, WASHINGTON, DC, USA
COMPUTER ENGNG AND COMPUTATION CENTER, IOWA UNIV, AMES, IA, USA

0051 AN IMPLEMENTATION OF A CODASYL BASED DATA-BASE MANAGEMENT SYSTEM UNDER THE UNIX OPERATING SYSTEM.
HOWARD JE
MASTER'S THESIS, JUN 1976, 167P.
NAVAL POSTGRADUATE SCHOOL, MONTEREY, CA

0052 I/O DEVICE EMULATION IN THE STANFORD EMULATION LABORATORY.
HUCK J + NEUHAUSER C
SIGMICRO NEWSL 10(4): 101-8 (DEC 1979)
COMPUTER SYSTEMS LAB, STANFORD UNIV, STANFORD, CA, USA
SAC 1980: 021390

0053 SOFTWARE SYSTEMS RESEARCH: REPORT FROM NOVEMBER 16, 1977, TO NOVEMBER 15, 1978.
DEPT OF COMP SCI, ILLINOIS UNIV, URBANA, ILL
COO-2383, 1978, 23P.

0054 UNIX-AN EASY-TO-USE OPERATING SYSTEM DEVELOPED BY BELL TELEPHONE LABORATORIES.
ISHIDA H
INF PROCESS SOC JPN (JOHO 18(9): 942-9 (1977)
(IN JAPANESE)
COMPUTER CENTRE, UNIV OF TOKYO, TOKYO, JAPAN
SAC 1978: 012216

0055 PROGRAMMER'S WORKBENCH: A MACHINE FOR SOFTWARE DEVELOPMENT.
IVIE EL
COMMUN ACM 20(10): 746-753 (1977)
BELL LABORATORIES

0056 THE LINE DRAWING EDITOR, AN EXPERIMENT IN COMPUTER VISION.
JARVIS JF
COMP GRAPHICS IMAGE PROCESS 6(5): 133-39 (OCT 1977)
BELL LABORATORIES

0057 LANGUAGE DEVELOPMENT TOOLS ON THE UNIX SYSTEM.
JOHNSON SC
COMPUTER 13: 16-21 (AUG 1980)
BELL LABORATORIES

0058 UNIX TIME-SHARING SYSTEM: LANGUAGE DEVELOPMENT TOOLS.
JOHNSON SC + LESK ME
BELL SYST TECH J 57(6): PT2 2155-75 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 031191

0059 UNIX TIME-SHARING SYSTEM: PORTABILITY OF C PROGRAMS AND THE UNIX SYSTEM.
JOHNSON SC + RITCHIE DM
BELL SYST TECH J 57(6): PT2 2021-48 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 031197

0060 IN-HOUSE SOFTWARE DEVELOPMENT IN THE AGSM (AUSTRALIAN GRADUATE SCHOOL OF MANAGEMENT).
JOHNSTONE IL + TAYLOR P
P59-68 OF PROC OF THE NATIONAL CONF ON LIBRARY AND BIBLIOGRAPHIC APPLICATIONS OF MINICOMPUTERS, SYDNEY, AUSTRALIA 22-24 AUG 1979, MIDDLETON, MR(ED), KENSINGTON, NSW, AUSTRALIA UNISEARCH LTD, 1979
SAC 1980: 024707

0061 IMPLEMENTATION OF AN ADAPTIVE SCHEDULING ALGORITHM FOR THE MUNIX OPERATING SYSTEM.
JOY RE
MASTER'S THESIS, 1975
NAVAL POSTGRADUATE SCHOOL, MONTEREY, CA

0062 ADDITION OF DATA PAGING TO THE UNIX OPERATING SYSTEM.
JUNG P HONG
P199-200 OF 1979 INT MICRO AND MINI COMPUTER CONF, HOUSTON TX, USA, 14-16 NOV 1979, IEEE, NEW YORK, USA, 1979
LOS ALAMOS SCI LAB, LOS ALAMOS, NM, USA
SAC 1980: 012956

0063 PRELIMINARY STEP TOWARDS A LANGUAGE FOR PICTURE CONTROL IN A REAL-TIME MODE.
KATZ L + ETRA B
SIGPLAN NOTICES 11(6): 73-8 (JUN 1976)
COLUMBIA UNIV, COLLEGE PHYSICIANS SURGEONS, NY, NY, USA

0064 GRAPHICS SATELLITE FOR THE UNIX TIME-SHARING SYSTEM.
KAVANAGH RN + HARDIE PA + VIIK AA
P72-5 OF SYMP ON TRENDS AND APPLICATIONS, 1976, MICRO AND MINI SYSTEMS, IEEE, MAY 1976, GAITHERSBURG, MARYLAND.
SASKATCHEWAN UNIV, SASKATOON, SASKATCHEWAN, CANADA
SAC 1976: 25419

0065 UNIX PROGRAMMING ENVIRONMENT.
KERNIGHAN BW + MASHEY JR
SOFTWARE-PRACT EXPER 9(1): 1-15 (JAN 1979)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1979: 013053

0066 UNIX TIME-SHARING SYSTEM: DOCUMENT PREPARATION.
KERNIGHAN BW + LESK ME + OSSANNA JF
BELL SYST TECH J (USA) 57(6): PT2 2115-35 JULY-AUG 1978
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 031256

0067 SYSTEM FOR TYPESETTING MATHEMATICS.
KERNIGHAN BW + CHERRY LL
COMMUN ACM 18(3): 151-7 (MAR 1975)
BELL LABORATORIES

0068 UNIX MULTI-ACCESS SYSTEM FOR PDP-11 COMPUTERS.
KILGOUR AC
IUCC NEWSL 6(2): 11-14 (SUMMER 1978)
DEPT OF COMPUTING SCI, UNIV OF GLASGOW, GLASGOW, SCOTLAND
SAC 1978: 031269

0069 SOFTWARE FILTERS FOR GRAPHICAL OUTPUT AND INTERACTION.
KILGOUR AC
P 194-200 OF PROC OF 4TH INTERNATL CONF AND EXHIBITION ON COMPUTERS IN DESIGN ENG, MARCH 1980, IPC SCI AND TECHNOL PRESS, GUILDFORD, ENGLAND
DEPT OF COMPUTING SCI, UNIV OF GLASGOW, GLASGOW, SCOTLAND

0070 INTERFACES, SUBROUTINES, AND PROGRAMS FOR THE GRINNELL GRM-27 DISPLAY PROCESSOR ON A PDP-11/45 WITH THE UNIX OPERATING SYSTEM.
KIRBY RL + SMITH R + DONDES PA + RANADE S + KITCHEN L
AD-A086 098/1, OCT 1979, 172P.
MARYLAND UNIV, COMPUTER VISION LAB, COLLEGE PARK, MD

0071 A MODIFICATION REQUEST CONTROL SYSTEM.
KNUDSEN DB + BAROFSKY A + SATZ LR
P187-192 OF PROC IEEE NATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 2ND, 1976, SAN FRANCISCO
BELL LABORATORIES

0072 COMPUTER TYPESETTING OF TECHNICAL JOURNALS ON UNIX.
LESK ME + KERNIGHAN BW
P879-888 OF PROC OF AFIPS NATL COMP CONF, 1977, DALLAS, 13-16 JUN 1977
BELL LABORATORIES

0073 COMPUTER-BASED GROUP DECISION AIDING.
LEVIN S + JOHNSTON S + LEAL A + WELTMAN G
P1396-401 OF PROC OF THE INT CONF ON CYBERNETICS AND SOCIETY, TOKYO-KYOTO, JAPAN, 3-7 NOV 1978, IEEE, NEW YORK, USA, 1978
PERCEPTRONICS, INC, ARLINGTON, VA, USA
SAC 1979: 027596

0074 EXPERIENCES WITH THE UNIX TIME-SHARING SYSTEM.
LIONS J
SOFTWARE-PRACT EXPER 9(9): 701-9 (SEPT 1979)
DEPT OF COMPUTER SCI, UNIV OF NEW SOUTH WALES, KENSINGTON, AUSTRALIA
SAC 1979: 034784

0075 AN OPERATING SYSTEM CASE STUDY.
LIONS J
OPER SYST REV 12(3): 46-53 (JULY 1978)
DEPT OF COMPUTER SCI, UNIV OF NEW SOUTH WALES, KENSINGTON, NSW, AUSTRALIA
SAC 1978: 028353

0076 UNIX TIME-SHARING SYSTEM: THE UNIX OPERATING SYSTEM AS A BASE FOR APPLICATIONS.
LUDERER GWR + MARANZANO JF + TAGUE BA
BELL SYST TECH J 57(6): PT2 2201-7 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 031257

0077 UNIX TIME-SHARING SYSTEM: UNIX ON A MICROPROCESSOR.
LYCKLAMA H
BELL SYST TECH J 57(6): PT2 2087-101 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 031254

0078 UNIX ON A MICROPROCESSOR.
LYCKLAMA H
P237-242 OF PROC OF AFIPS NATL COMP CONF, 1977, DALLAS, 13-16 JUN 1977
BELL LABORATORIES

0079 UNIX TIME-SHARING SYSTEM: THE MERT OPERATING SYSTEM.
LYCKLAMA H + BAYER DL
BELL SYST TECH J 57(6): PT2 2049-86 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 031253

0080 UNIX TIME-SHARING SYSTEM: A MINICOMPUTER SATELLITE PROCESSOR SYSTEM.
LYCKLAMA H + CHRISTENSEN C
BELL SYST TECH J 57(6): PT2 2103-13 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 031255

0081 A UNIX-BASED LOCAL PROCESSOR AND NETWORK ACCESS MACHINE.
MANNING EG + HOWARD R + O'DONNEL CG + PAMMETT K + CHANG E
COMPUT NETWORKS 1: 139-42 (1976)
WATERLOO UNIV, ONTARIO, CANADA

0082 USING A COMMAND LANGUAGE AS A HIGH-LEVEL PROGRAMMING LANGUAGE.
MASHEY JR
P169-176 OF PROC IEEE NATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 2ND, 1976, SAN FRANCISCO
BELL LABORATORIES

0083 DOCUMENTATION TOOLS AND TECHNIQUES.
MASHEY JR + SMITH DW
P177-181 OF PROC IEEE NATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 2ND, 1976, SAN FRANCISCO.
BELL LABORATORIES

0084 SOFTWARE: THE NEXT FIVE YEARS.
MC CLURE RM
P6-7 OF 12 IEEE COMP SOC INT CONF PROC(COMPCON76), FEB 24-26, 1976
PALYN ASSOCIATES

0085 PRELIMINARY DESIGN OF INGRES: PART-4.
MC DONALD N + STONEBRAKER M + WONG G
ERL-M435, APR 1974
CALIFORNIA UNIV, ELECTRONICS RES LAB, BERKELEY, CA

0086 KSOS: A SECURE OPERATING SYSTEM.
MCCAULEY EJ
P35-9 OF PROC OF SPRING COMPCON 79, SAN FRANCISCO, CA, USA 26 FEB - 1 MARCH 1979, IEEE, NEW YORK, USA, 1979
FORD AEROSPACE AND COMMUNICATIONS CORP, PALO ALTO, CA, USA
SAC 1979: 016176

0087 KSOS-THE DESIGN OF A SECURE OPERATING SYSTEM.
MCCAULEY EJ + DRONGOWSKI PJ
P345-53 OF AFIPS CONF PROC, 1979, NATIONAL COMPUTER CONF, NEW YORK, USA, 4-7 JUNE 1979, AFIPS, MONTVALE, NJ, USA, 1979
FORD AEROSPACE AND COMMUNICATIONS CORP, PALO ALTO, CA, USA
SAC 1980: 013073

0088 AN ENHANCEMENT OF THE COMPUTER TYPESETTING CAPABILITY OF UNIX.
MCCORD BS
MASTER'S THESIS, JUN 1977, 139P, AD-A044 183/2ST
NAVAL POSTGRADUATE SCHOOL, MONTEREY, CA

0089 USER EXPERIENCE WITH MODULA FOR PROGRAMMING A REAL-TIME APPLICATION.
MCFADDEN SM
P865-870 OF PROC OF THE DIGITAL EQUIPMENT COMPUTER USERS SOCIETY, TORONTO, ONTARIO, FEB 80

0090 SYNTHETIC ENGLISH SPEECH BY RULE.
MCILROY MD
COMPUTING SCIENCE TECHNICAL REPORT NO 14, 1974
BELL LABORATORIES

0091 HIERARCHICAL SYMBOLIC REPRESENTATION FOR AN IMAGE DATABASE.
MCKEOWN DM + RAJ REDDY D
P40-44 OF PROC OF WORKSHOP ON PICT DATA DESCR AND MANAGE, CHICAGO, APR 21-22, 1977
CARNEGIE-MELLON UNIV, PITTSBURGH, PA

0092 UNIX TIME-SHARING SYSTEM: STATISTICAL TEXT PROCESSING.
MCMAHON LE + CHERRY LL + MORRIS R
BELL SYST TECH J 57(6): PT2 2137-54 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 032027

0093 REDAS-A RELATIONAL DATA ACCESS SYSTEM FOR REAL-TIME APPLICATIONS.
MCSKIMIN JR
P295-300 OF PROC OF COMPSAC 78 COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE, CHICAGO, IL, USA, 13-16 NOV 1978, IEEE, NEW YORK, USA, 1978
BELL LABS, COLUMBUS, OH, USA
SAC 1979: 010166

0094 EVALUATION OF THE UNIX TIME-SHARING SYSTEM.
MELENDEZ KJ + JOHNSON RT
LA-6775-MS, APR 1977, 14P
LOS ALAMOS SCI LAB, NEW MEXICO

0095 UNIX-A PORTABLE OPERATING SYSTEM.
MILLER R
OPER SYST REV 12(3): 32-7 (JULY 1978)
DEPT OF MATH, UNIV OF WOLLONGONG, WOLLONGONG, NSW, AUSTRALIA
SAC 1978: 028351

0096 EASY DOES IT (UNIX SYSTEM).
MORGAN SP
TELEPHONY 196(13): 50,52-3,56 (26 MARCH 1979)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1979: 030496

0097 UNIX SYSTEM: MAKING COMPUTERS EASIER TO USE.
MORGAN SP
BELL LAB REC 56(11): 308-13 (DEC 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1979: 010104

0098 COMPUTER DETECTION OF TYPOGRAPHICAL ERRORS.
MORRIS R + CHERRY LL
IEEE TRANS PROF COMMUN PC-18(1): 54-6 (MAR 1975)
BELL LABORATORIES

0099 METHOD FOR REDUCING MEMORY CONFLICTS CAUSED BY BUSY WAITING IN MULTIPLE-PROCESSOR SYNCHRONISATION.
MUHLEMANN K
IEE PROC E 127(3): 85-7 (MAY 1980)
SWISS FEDERAL INST OF TECHNOL, ZURICH, SWITZERLAND
SAC 1980: 024431

0100 A REAL-TIME SATELLITE SYSTEM BASED ON UNIX.
MURREL S + KOWALSKI T
BEHAV RES METHODS AND INSTRUM 12(2): 126-31 (APRIL 1980)
BELL LABS, MURRAY HILL, NJ, USA

0101 UNIX TIME-SHARING SYSTEM: RBCS/RCMAS-CONVERTING TO THE MERT OPERATING SYSTEM.
NAGELBERG ER + PILLA MA
BELL SYST TECH J 57(6): PT2 2275-87 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 031259

0102 DESIGN OF A USER INTERFACE FOR A COLOR, RASTER SCAN GRAPHICS DEVICE.
NESSLAGE RL
MASTER'S THESIS, 1976
NAVAL POSTGRADUATE SCHOOL, MONTEREY, CA

0103 THE DEVELOPMENT OF A SEGMENTED MEMORY MANAGER FOR THE UNIX OPERATING SYSTEM WITH APPLICATIONS IN A MULTIPORTED MEMORY ENVIRONMENT.
O'DELL JM
MASTER'S THESIS, SEP 1977, 79P, AD-A047 170/6ST.
NAVAL POSTGRADUATE SCHOOL, MONTEREY, CA

0104 UNIX TIME-SHARING SYSTEM: NO.4 ESS DIAGNOSTIC ENVIRONMENT.
PEKARICH SP
BELL SYST TECH J 57(6): PT2 2265-74 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAB 1978: 050805

0105 A LOCAL NETWORK OF MINI AND MICROCOMPUTERS FOR EXPERIMENT SUPPORT.
POHM AV + DAVIS JA + CHRISTIANSEN S + BRIDGES GD + HORTON RE
COMPUT NETWORKS 3(6): 381-7 (DEC 1979)
DEPT OF ELECTRICAL ENGNG AND COMPUTATION CENTER,
IOWA STATE UNIV, AMES, IA
SAB 1980: 030676

0106 UCLA SECURE UNIX.
POPEK GJ + KAMPE M + KLINE CS + STOUGHTON A + URBAN M + WALTON EJ
P355-64 OF AFIPS CONF PROC VOL48 1979 NATIONAL COMPUTER CONF, NEW YORK, USA, 4-7 JUNE 1979
AFIPS, MONTVALE, NJ, USA, 1979
UNIV OF CALIFORNIA, LOS ANGELES, CA, USA
SAC 1980: 013074

0107 USING UNIX IN AN INSTRUCTIONAL ENVIRONMENT.
PRENNER CJ
P143-5 OF COMPUTERS SOC INT CONF ON COMPUTERS; THE NEXT FIVE YEARS, 12TH, IEEE, DIGEST OF PAPERS, FEB 1976, SAN FRANCISCO, CA.
CALIFORNIA UNIV, BERKELEY, CA
SAC 1976: 24972

0108 INSTRUCTIONAL COMPUTER SYSTEMS FOR HIGHER EDUCATION.
PRENNER CJ + SPECTOR AZ
P171-7 OF AFIPS 1976 NAT COMPUT CONF, JUN 1976.
CALIFORNIA UNIV, BERKELEY, CA

0109 COMPARATIVE STUDY OF THE FORTRAN DEVELOPMENT ENVIRONMENT PROVIDED BY THE VAX/VMS AND VAX/UNIX OPERATING SYSTEMS.
RAFFENETTI RC
ANL-AMD-TM-346, NOV 1979, 33P
ARGONNE NATIONAL LAB, IL

0110 PICTURE PAGING FOR EFFICIENT IMAGE PROCESSING.
REUSS JL + CHANG SK + MCCORMICK BH
P437 OF PROC OF THE 1978 CONF ON PATTERN RECOGNITION AND IMAGE PROCESSING, CHICAGO, IL, USA, 31 MAY - 2 JUNE 1978, IEEE, NEW YORK, USA 1978
DEPT OF INFORMATION ENGNG,
UNIV OF ILLINOIS, CHICAGO CIRCLE,
IL, USA
SAC 1978: 023366

0111 EVOLUTION OF THE UNIX TIME-SHARING SYSTEM.
RITCHIE DM
P25-35 OF LANGUAGE DESIGN AND PROGRAMMING METHODOLOGY PROC OF A SYMPOSIUM, SYDNEY, AUSTRALIA, 10-11 SEPT 1979, TOBIAS, JM(ED)
SPRINGER VERLAG, BERLIN, GERMANY, 1980
BELL LABS, MURRAY HILL, NJ, USA
SAC 1980: 027130

0112 UNIX TIME-SHARING SYSTEM: A RETROSPECTIVE.
RITCHIE DM
BELL SYST TECH J 57(6): PT2 1947-69 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 031252

0113 UNIX TIME-SHARING SYSTEM.
RITCHIE DM + THOMPSON K
BELL SYST TECH J 57(6): PT2 1905-29 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 031250

0114 UNIX TIME-SHARING SYSTEM.
RITCHIE DM + THOMPSON K
COMMUN ACM 17(7): 365-75 (JUL 1974)
BELL LABORATORIES
SAC 1975: 2771

0115 UNIX PROGRAMMER'S MANUAL, 6TH EDITION, 1975.
RITCHIE DM + THOMPSON K
BELL LABORATORIES

0116 PROGRAMMER'S WORKBENCH: NEW TOOLS FOR SOFTWARE DEVELOPMENT.
ROOME WD
BELL LAB REC 57(1): 19-25 (JAN 1979)
SAC 1979: 015969

0117 PERFORMANCE EVALUATION UNDER UNIX AND A STUDY OF PDP-11 INSTRUCTION USAGE.
ROSE G
OPER SYST REV 12(3): 38-45 (JULY 1978)
DEPT OF COMPUTING SCI, UNIV OF NEW SOUTH WALES, KENSINGTON, NSW, AUSTRALIA
SAC 1978: 028352

0118 UNIX TIME-SHARING SYSTEM: A SUPPORT ENVIRONMENT FOR MAC-8 SYSTEMS.
ROVEGNO HD
BELL SYST TECH J 57(6): PT2 2251-63 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 031258

0119 SOFTWARE DEVELOPMENT FOR MICROPROCESSORS, A CASE STUDY.
SALOMON FA
P110-13 OF PROC OF COMPSAC 78 COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE, CHICAGO, IL, USA, 13-16 NOV 1978, IEEE, NEW YORK, USA, 1978
BELL LABS, NAPERVILLE, IL, USA
SAC 1979: 009901

0120 IMPLEMENTATION OF INTEGRITY CONSTRAINTS IN THE RELATIONAL DATA-BASE SYSTEM, INGRES.
SCHOENBERG I
MASTER'S THESIS, DEP ELEC ENG COMPUT SCI, 1975
CALIFORNIA UNIV, BERKELEY, CA

0121 UNIX.
STIEFEL ML
MINI-MICRO SYST 11(4): 64-6 (APRIL 1978)
SAC 1978: 023417

0122 RETROSPECTION ON A DATABASE SYSTEM.
STONEBRAKER M
ACM TRANS DATABASE SYST 5(2): 225-40 (JUNE 1980)
UNIV OF CALIFORNIA, CA, USA
SAC 1980: 024463

0123 DISTRIBUTED DATA-BASE VERSION OF INGRES.
STONEBRAKER M + NEUHOLD E
ERL-M612, 11 SEP 1976, 33P.
CALIFORNIA UNIV, BERKELEY, CA

0124 THE INGRES PROTECTION SYSTEM.
STONEBRAKER M + RUBINSTEIN P
PROC 1976 ACM NAT CONF, HOUSTON, TEX, OCT 1976
CALIFORNIA UNIV, BERKELEY, CA

0125 THE DESIGN AND IMPLEMENTATION OF INGRES.
STONEBRAKER M + WONG E + KREPS P + HELD G
ACM TRANS DATABASE SYST 1(3): 189-222 (SEP 1976)
CALIFORNIA UNIV, BERKELEY, CA

0126 INGRES - A RELATIONAL DATABASE SYSTEM, FINAL REPORT.
STONEBRAKER M + WONG E
AD-A082 548/9
CALIFORNIA UNIV, BERKELEY, CA

0127 PROCESS STRUCTURE ALTERNATIVES TOWARDS A DISTRIBUTED INGRES.
THOMAS RAC
P215-27 OF DISTRIBUTED DATA BASES, PROC OF THE INT SYMPOSIUM ON DISTRIBUTED DATA BASES, PARIS, FRANCE, 12-14 MARCH 1980, DELOBEL, C, LITWIN, W(ED), NORTH-HOLLAND, AMSTERDAM, NETHERLANDS, 1980
VRIJE UNIV, AMSTERDAM, NETHERLANDS
SAC 1980: 027158

0128 UNIX NSW FRONT END.
THOMAS RH
AD-A084 088/4, MAR 1980, 108P
BOLT, BERANEK & NEWMAN, CAMBRIDGE, MA

0129 PLANNING FOR ACCAT REMOTE SITE OPERATIONS.
THOMAS RH + JOHNSON P
AD-A058 461/5ST, AUG 1078, 30P
BOLT, BERANEK & NEWMAN, CAMBRIDGE, MA

0130 UNIX TIME-SHARING SYSTEM: UNIX IMPLEMENTATION.
THOMPSON K
BELL SYST TECH J 57(6): PT2 1931-46 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 031251

0131 UNIX COMMAND LANGUAGE.
THOMPSON K
P375-84 OF STRUCTURED PROGRAMMING INT STATE OF THE ART REPORT, INFOTECH INT, MAIDENHEAD, BERKS, ENGLAND
BELL LABORATORIES
SAC 1976: 18944

0132 SOFTWARE DEVELOPMENT CONTROL BASED ON MODULE INTERCONNECTION.
TICHY WF
P29-41 OF PROC OF THE 4TH INT CONF ON SOFTWARE ENGINEERING, MUNICH, GERMANY, 17-19 SEPT 1979, IEEE, NEW YORK, USA, 1979
DEPT OF COMPUTER SCI, CARNEGIE-MELLON UNIV, PITTSBURGH, PA, USA
SAC 1980: 005681

0133 A MODULAR IMPLEMENTATION AND SIMULATION OF THE UNIX OPERATING SYSTEM.
UNGER BW + MUTALIK PR
P892-6 OF PROC OF THE 1978 SUMMER COMPUTER SIMULATION CONF LOS ANGELES, CA, USA, 24-26 JULY 1978, AFIPS PRESS, MONTVALE, NJ, USA 1978
COMPUTER SCI DEPT, UNIV OF CALGARY, CALGARY, ALBERTA, CANADA
SAC 1979: 010114

0134 INTERPROCESS COMMUNICATION EXTENSIONS FOR THE UNIX OPERATING SYSTEM, PART-1. DESIGN CONSIDERATIONS.
SUNSHINE, CA
AD-A044 200/4ST, JUN 1977, 36P
RAND CORP

0135 PRACTICAL COURSES OF STUDY ON THE BASIC PROGRAMMING LANGUAGE IN THE DATA TRAINING GROUP OF THE FREE UNIVERSITY.
VAN DE RIET RP
INFORMATIE 20(10): 578-86 (OCT 1978) (IN DUTCH)
SAC 1979: 002569

0136 IMPLEMENTATION OF A SECURE DATA MANAGEMENT SYSTEM FOR THE SECURE UNIX OPERATING SYSTEM.
WAGNER BN
AD-A056 902/0ST, JUL 1978, 41P
MITRE CORP, BEDFORD MA

0137 SPECIFICATION AND VERIFICATION OF THE UCLA UNIX SECURITY KERNEL.
WALKER BJ + KEMMERER RA + POPEK GJ
P64-5 OF PROC OF THE SEVENTH SYMP ON OPERATING SYSTEMS PRINCIPLES, PACIFIC GROVE, CA, USA, 10-12 DEC 1979, ACM, NEW YORK, USA 1979
UNIV OF CALIFORNIA, LOS ANGELES, CA, USA
SAC 1980: 021739

0138 SPECIFICATION AND VERIFICATION OF THE UCLA UNIX SECURITY KERNEL.
WALKER BJ + KEMMERER RA + POPEK GJ
COMMUN ACM 23(2): 118-31 (FEB 1980)
UNIV OF CALIFORNIA, LOS ANGELES, CA, USA
SAC 1980: 016416

0139 EXPERT ASSISTANCE SYSTEM: ONE APPROACH TOWARDS PEOPLE-ORIENTED SYSTEMS.
WATKINS SW
SIGSOC BULL 11(1): 7-8 (JULY 1979)
NAT BUR OF STAND, BOULDER, CO, USA
SAC 1980: 005755

0140 BACKGROUND AND STATUS OF THE EPC-11 EXPERIMENT.
WHITBY OW
IEEE TRANS PROF COMMUN PC-20(1): 6-12 (1977)
STANFORD RES INST, INFO SCI LAB, MENLO PARK, CA

0141 UNIX TIME-SHARING SYSTEM: MICROCOMPUTER CONTROL OF APPARATUS, MACHINERY, AND EXPERIMENTS.
WONSIEWICZ BC + STORM AR + SIEBER JD
BELL SYST TECH J 57(6): PT2 2209-32 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 031862

0142 A KERNEL-BASED SECURE UNIX DESIGN.
WOODWARD JPL + NIBALDI GA
AD-A073 173/7ST, MAY 1979, 95P
MITRE CORP, BEDFORD, MA

0143 ADVANCED TEXT PROCESSING USING UNIX.
YORMARK B
P22/1/1-3 OF 1978 MIDCON TECHNICAL PAPERS, DALLAS, TX, USA, 12-14 DEC 1978, WESTERN PERIODICALS CO, NORTH HOLLYWOOD, CA, USA, 1978
INTERACTIVE SYSTEMS CORP, SANTA MONICA, CA
SAC 1979: 033968

0144 INGRES REFERENCE MANUAL, 5.
ZOOK W, ET AL
MEMORANDUM ERL-M585, APR 1976
CALIFORNIA UNIV, ELECTRONICS RES LAB, BERKELEY, CA

0145 INTERPROCESS COMMUNICATION EXTENSIONS FOR THE UNIX OPERATING SYSTEM, PART-2. IMPLEMENTATION.
ZUCKER S
AD-A044 201/4ST, JUN 1977, 24P
RAND CORP

AUUGN

0146 EMBEDDING A RELATIONAL DATA SUBLANGUAGE IN A GENERAL PURPOSE PROGRAMMING LANGUAGE. (INGRES, EQUEL, QUEL)
ALLMAN E + STONEBRAKER M + HELD G
SIGPLAN NOTICES 11 (SPEC ISS): 25-35 (1976)
CALIFORNIA UNIV, BERKELEY, CA

0147 TYPE SYNTAX IN THE LANGUAGE C, AN OBJECT LESSON IN SYNTACTIC INNOVATION.
ANDERSON B
SIGPLAN NOT 15(3): 21-7 (MARCH 1980)
MAN-MACHINE LAB, UNIV OF ESSEX, COLCHESTER, ENGLAND
SAC 1980: 018930

0148 A COROUTINE PACKAGE FOR C.
BAILES PAC
AUST COMPUT SCI COMMUN 1(4): 306-9 (DEC 1979)
DEPT OF COMPUTER SCI, UNIV OF QUEENSLAND, ST LUCIA, AUSTRALIA
SAC 1980: 024312

0149 AN ALGORITHM FOR STRUCTURING FLOW GRAPHS.
BAKER BS
J ACM 24: 98-120 (JAN 1977)
BELL LABORATORIES

0150 A PARSER GENERATION TOOL FOR MICRO-COMPUTERS.
BURGER WF
P631-4 OF PROC OF COMPSAC THE IEEE COMPUTER SOCIETY S THIRD INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONF, CHICAGO, IL USA, 6-8 NOV 1979, IEEE, NEW YORK, USA, 1979
DEPT OF COMPUTER SCI, UNIV OF TEXAS, AUSTIN, TX, USA
SAC 1980: 012916

0151 A SMALL C COMPILER FOR THE 8080'S.
CAIN R
DR DOBB'S J COMPUT CALISTHENICS AND ORTHOD
5(5): 5-19 (MAY 1980)

0152 PROGRAMMING LANGUAGES AND STANDARDS.
CHAPMAN D
DR DOBB'S J COMPUT CALISTHENICS AND ORTHOD
3(10): 18-21 (NOV-DEC 1978)
WESTERN RESERVE ACAD, HUDSON, OH, USA
SAC 1979: 016122

0153 A UNIFIED HARDWARE DESCRIPTION LANGUAGE FOR CAD PROGRAMS.
CRAWFORD JD + NEWTON AR + PEDERSON DO + BOYLE GR
P151-4 OF PROC OF THE 4TH INT SYMP ON COMPUTER HARDWARE DESCRIPTION LANGUAGES, PALO ALTO, CA, USA, 8-9 OCT 1979, IEEE NEW YORK, USA, 1979
DEPT OF ELECTRICAL ENGNG, COMPUTER SCI, UNIV OF CALIFORNIA, BERKELEY, CA, USA
SAC 1980: 015800

0154 ADAPTATION OF THE HERSHEY DIGITIZED CHARACTER SET FOR USE IN COMPUTER GRAPHICS AND TYPESETTING.
DOYLE PM
JUNE 1977, AD-A042 291/5WC

0155 CAUTION: STRUCTURED PROGRAMMING CAN BE HABIT-FORMING.
GIBSON TA
CREATIVE COMPUT 5(1): 68-71 (JAN 1979)
TINY C ASSOCIATES, HOLMDEL, NJ, USA
SAC 1979: 027513

0156 STRUCTURED PROGRAMMING, C AND TINY C.
GIBSON TA + GUTHERY SB
DR DOBB'S J COMPUT CALISTHENICS AND ORTHOD
5(5): 30-33 (MAY 1980)
TINY-C ASSOC, HOLMDEL, NJ, USA

0157 'FLOWBLOCKS'-A TECHNIQUE FOR STRUCTURED PROGRAMMING.
GROUSE P
SIGPLAN NOT 13(2): 46-56 (FEB 1978)
SCHOOL OF ACCOUNTANCY, UNIV OF NEW SOUTH WALES, SYDNEY, AUSTRALIA
SAC 1978: 014886

0158 AN OPTIMIZER FOR A C COMPILER FOR THE SERIES/1.
HAMMOND RA
P85-91 OF PROC OF MICRO-DELCON 80 THE DELAWARE BAY MICROCOMPUTER CONFERENCE, NEWARK, DE, USA, 11 MARCH 1980, IEEE, NEW YORK, USA, 1980
DEPT OF ELECTRICAL ENGNG, UNIV OF DELAWARE, NEWARK, DE, USA
SAC 1980: 021679

0159 IMPLEMENTING A TINY INTERPRETER WITH A CP/M-FLAVORED C.
HANCOCK L
DR DOBB'S J COMPUT CALISTHENICS AND ORTHOD
5(1): 20-8 (JAN 1980)
YOURDON INC, NEW YORK, NY, USA
SAC 1980: 027064

0160 DECISION LOGIC TABLE PREPROCESSOR.
KELLER JF + ROESCH RW
AD-A041 154/6WC, JUN 1977
NAVAL POSTGRADUATE SCHOOL, MONTEREY, CA

0161 A GUIDE TO NED: A NEW ON-LINE COMPUTER EDITOR.
KELLEY J
AD-A045 157/5WC, 44P. (JULY 1977)
RAND CORP

0162 A USER'S LOOK AT TINY-C.
KERN CO
BYTE 4(12): 196,198,200-2,204-6 (DEC 1979)
SAC 1980: 016344

0163 THE C PROGRAMMING LANGUAGE.
KERNIGHAN BW + RITCHIE DM
PRENTICE-HALL, 1978, 230P.
BELL LABORATORIES

0164 C LANGUAGE'S GRIP ON HARDWARE MAKES SENSE FOR SMALL COMPUTERS.
KRIEGER MS + PLAUGER PJ
ELECTRONICS 53(11): 129-33 (8 MAY 1980)
WHITESMITHS LTD, NEW YORK, NY, USA
SAC 1980: 021639

0165 A SYSTEM FOR RESOURCE-SHARING IN A DISTRIBUTED ENVIRONMENT: RIDE.
LU PM
P427-33 OF PROC OF COMPSAC THE IEEE COMPUTER SOCIETY S THIRD INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONF, CHICAGO, IL USA, 6-8 NOV 1979, IEEE, NEW YORK, USA, 1979
BELL LABS, NAPERVILLE, IL, USA
SAC 1980: 013080

0166 C: A LANGUAGE FOR MICROPROCESSORS .
MADDEN JG
BYTE 2(10): 130,132,134,136,138 (OCT 1977)
JGM DEV LABS, WEST LAFAYETTE, IN, USA
SAC 1978: 012138

0167 PASCAL VERSUS C: A SUBJECTIVE COMPARISON.
MATETI P
P37-69 OF LANGUAGE DESIGN AND PROGRAMMING METHODOLOGY PROC OF A SYMPOSIUM, SYDNEY, AUSTRALIA, 10-11 SEPT 1979, TOBIAS, JM(ED)
SPRINGER VERLAG, BERLIN, GERMANY, 1980
DEPT OF COMPUTER SCI, UNIV OF MELBOURNE, PA
SAC 1980: 027050

0168 PASCAL OR C: DETAILS DECIDING FACTOR.
MICHAUD EE
COMP BUSINESS NEWS 3(34):1,5-6 (25 AUG 1980)

0169 A BLUE COLLAR LANGUAGE FOR CAD.
NEWTON AR
P81-2 OF COMPCON SPRING 80 VLSI' NEW ARCHITECTURAL HORIZONS SAN FRANCISCO, CA, USA, 25-28 FEB 1980, IEEE, NEW YORK, USA, 1980
DEPT OF ELECTRICAL ENGNG AND COMPUTER SCI, UNIV OF CALIFORNIA, BERKELEY, CA, USA
SAC 1980: 024364

0170 PASCAL VS. C DEBATE GETS HOT IN THE SMALL CPU ENVIRONMENT.
O'CONNOR RJ
COMP BUSINESS NEWS 3(34):1,4 (25 AUG 1980)

0171 UNIX TIME-SHARING SYSTEM: THE C PROGRAMMING LANGUAGE.
RITCHIE DM + JOHNSON SC + LESK ME + KERNIGHAN BW
BELL SYST TECH J 57(6): PT2 1991-2019 (JULY-AUG 1978)
BELL LABS, MURRAY HILL, NJ, USA
SAC 1978: 031196

0172 THE C PROGRAMMING LANGUAGE.
RITCHIE DM + JOHNSON SC + LESK ME + KERNIGHAN BW
DR DOBB'S J COMPUT CALISTHENICS AND ORTHOD
5(5): 20-9 (MAY 1980)
BELL LABS, MURRAY HILL, NJ, USA

0173 AN EXTENDED BASIC COMPILER WITH GRAPHICS INTERFACE FOR THE PDP-11/50 COMPUTER.
ROBERTSON MD
MASTER'S THESIS, JUN 1977, 208P, AD-A044 366/3ST
NAVAL POSTGRADUATE SCHOOL, MONTEREY, CA

0174 USE OF THE C LANGUAGE FOR MICROPROCESSORS.
ROVEGNO HD
P24/2/1-3 OF 1977 ELECTRO CONF RECORD, NEW YORK, USA 19-20 APRIL 1977, ELECTRO, EL SEGUNDO, CALIF, USA, 1977
BELL LABS INC, HOLMDEL, NJ, USA
SAC 1978: 009315

0175 PICK A COMPUTER LANGUAGE THAT FITS THE JOB.
SCHINDLER M
ELECTRON DES 28(15): 62-70,72,74,76,78 (JULY 1980)

0176 REPORT ON THE PROGRAMMING LANGUAGE PLZ/SYS.
SNOOK T + BASS C + ROBERTS J + NAHAPETIAN A + FAY M
SPRINGER-VERLAG, BERLIN, GERMANY, 1978, VI+90P.
SAC 1979: 033918

0177 SOFTWARE TOOLS PROJECT.
SNOW CR
SOFTWARE-PRACT EXPER 8(5): 585-99 (SEPT-OCT 1978)
COMPUTING LAB, UNIV OF NEWCASTLE UPON TYNE, TYNE, ENGLAND
SAC 1979: 001897

0178 THE DESIGN AND IMPLEMENTATION OF A GENERAL PURPOSE INTERACTIVE GRAPHICS SUBROUTINE LIBRARY.
STANKOWSKI BJ
MASTERS THESIS, SEP 1976, AD-A03232416
NAVAL POSTGRADUATE SCHOOL, MONTEREY, CA

0179 MODULARISATION. II. THE MODULAR LANGUAGES.
STIRZALKOWSKI P
INFORMATYKA 14(7): 14-18 (JULY 1979) (IN POLISH)
INST BADAN JADROWYCH (CYFRONET), SWIERK, POLAND
SAC 1980: 009121

0180 PARTIAL DERIVATIVE GENERATOR.
WARNER DD
COMPUTING SCIENCE TECHNICAL REPORT NO 28, 1975
BELL LABORATORIES

0181 C NOTES: A GUIDE TO THE C PROGRAMMING LANGUAGE.
ZAHN CT
YOURDON PRESS, 1979, 102P.

ALEXANDER ST	0036	IMPLEMENTATION AND PERFORMANCE OF A UNIX LINK.
ALLMAN E	0146	EMBEDDING A RELATIONAL DATA SUBLANGUAGE IN A GENERAL PURPOSE PROGRAMMING LANGUAGE.
ANDERSON B	0147	TYPE SYNTAX IN THE LANGUAGE C, AN OBJECT LESSON IN SYNTACTIC INNOVATION.
BAILES PAC	0148	A COROUTINE PACKAGE FOR C.
BAKER BS	0149	AN ALGORITHM FOR STRUCTURING FLOW GRAPHS.
BALOCCA R	0002	NETWORKING AND THE PROCESS STRUCTURE OF UNIX: A CASE STUDY.
BARAK AB	0003	UNIX WITH SATELLITE PROCESSORS.
BAROFSKY A	0071	A MODIFICATION REQUEST CONTROL SYSTEM.
BASS C	0176	REPORT ON THE PROGRAMMING LANGUAGE PLZ/SYS.
BAYER DL	0079	UNIX TIME-SHARING SYSTEM: THE MERT OPERATING SYSTEM.
BELL LABORATORIES, 7TH E	0004	UNIX TIME-SHARING SYSTEM: UNIX PROGRAMMER'S MANUAL.
BELL LABORATORIES, VERSI	0005	UNIX/32V TIME-SHARING SYSTEM: UNIX PROGRAMMER'S MANUAL.
BIANCHI MH	0006	A USER'S VIEWPOINT ON THE PROGRAMMER'S WORKBENCH.
BIGGER JT	0007	A SHAPE ORIENTED SYSTEM FOR AUTOMATED HOLTER ECG ANALYSIS.
BIRMAN KP	0007	A SHAPE ORIENTED SYSTEM FOR AUTOMATED HOLTER ECG ANALYSIS.
BISIANI R	0008	SOFTWARE DEVELOPMENT FOR TASK-ORIENTED MULTIPROCESSOR ARCHITECTURES.
BLOOMFIELD P	0009	AN INTERACTIVE STATISTICAL PROCESSOR FOR THE UNIX TIME-SHARING SYSTEM.
BOEHM APW	0010	THE INSTALLATION OF ALICE ON THE PDP 11/45 UNDER UNIX.
BORDAGE G	0011	AN INTRODUCTORY COURSE IN THE APPLICATIONS OF COMPUTER TECHNOLOGY IN THE HEALTH SC
BORDEN BS	0012	THE MH MESSAGE HANDLING SYSTEM: USER'S MANUAL.
BOURNE SR	0013	UNIX TIME-SHARING SYSTEM: THE UNIX SHELL.
BOYLE GR	0153	A UNIFIED HARDWARE DESCRIPTION LANGUAGE FOR CAD PROGRAMS.
BRACKETT JW	0027	AN ENVIRONMENT FOR PRODUCING WELL-ENGINEERED MICROCOMPUTER SOFTWARE.
BRESSLET RD	0014	COMBINED QUARTERLY TECHNICAL REPORT NO. 10: PACKET BROADCAST BY SATELLITE, PLURIBU
BRIDGES GD	0105	A LOCAL NETWORK OF MINI AND MICROCOMPUTERS FOR EXPERIMENT SUPPORT.
BUNT RB	0015	SCHEDULING TECHNIQUES FOR OPERATING SYSTEMS.
BURGER WF	0150	A PARSER GENERATION TOOL FOR MICRO-COMPUTERS.
CAIN R	0151	A SMALL C COMPILER FOR THE 8080'S.
CERMAK IA	0016	AN INTEGRATED APPROACH TO MICROCOMPUTER SUPPORT TOOLS.
CHANG E	0017	DISTRIBUTED MEDICAL DATA-BASE: NETWORK SOFTWARE DESIGN.
	0081	A UNIX-BASED LOCAL PROCESSOR AND NETWORK ACCESS MACHINE.
CHANG SK	0110	PICTURE PAGING FOR EFFICIENT IMAGE PROCESSING.
CHAPMAN D	0152	PROGRAMMING LANGUAGES AND STANDARDS.
CHERRY LL	0067	SYSTEM FOR TYPESETTING MATHEMATICS.
	0092	UNIX TIME-SHARING SYSTEM: STATISTICAL TEXT PROCESSING.
	0098	COMPUTER DETECTION OF TYPOGRAPHICAL ERRORS.
CHESSON GL	0018	NETWORK UNIX SYSTEM.
CHRISTENSEN C	0080	UNIX TIME-SHARING SYSTEM: A MINICOMPUTER SATELLITE PROCESSOR SYSTEM.
CHRISTIANSEN S	0105	A LOCAL NETWORK OF MINI AND MICROCOMPUTERS FOR EXPERIMENT SUPPORT.
COHEN H	0019	UNIX TIME-SHARING SYSTEM: THE NETWORK OPERATIONS CENTER SYSTEM.
COTRELL J	0020	MULTILEVEL SECURITY FOR INTELLIGENCE DATA PROCESSING SYSTEMS.
CRAWFORD JD	0153	A UNIFIED HARDWARE DESCRIPTION LANGUAGE FOR CAD PROGRAMS.
CURTIS P	0021	PLOT: A UNIX PROGRAM FOR INCLUDING GRAPHICS IN DOCUMENTS.
CUSTEAD LR	0022	NUCLEAR PHYSICS DATA ACQUISITION WITH THE UNIX TIME-SHARING SYSTEM.
DAVIS JA	0105	A LOCAL NETWORK OF MINI AND MICROCOMPUTERS FOR EXPERIMENT SUPPORT.
DEPT OF COMP SCI, ILLINO	0053	SOFTWARE SYSTEMS RESEARCH: REPORT FROM NOVEMBER 16, 1977, TO NOVEMBER 15, 1978.
DOLOTTA TA	0023	UNIX TIME-SHARING SYSTEM: THE PROGRAMMER'S WORKBENCH.
	0024	LEAP LOAD AND TEST DRIVER.
	0025	INTRODUCTION TO THE PROGRAMMER'S WORKBENCH.
	0026	USING A COMMAND LANGUAGE AS THE PRIMARY PROGRAMMING TOOL.
DONDES PA	0070	INTERFACES, SUBROUTINES, AND PROGRAMS FOR THE GRINNELL GRM-27 DISPLAY PROCESSOR ON
DOYLE PM	0154	ADAPTATION OF THE HERSHEY DIGITIZED CHARACTER SET FOR USE IN COMPUTER GRAPHICS AND
DRONGOWSKI PJ	0087	KSOS-THE DESIGN OF A SECURE OPERATING SYSTEM.
EANES RS	0027	AN ENVIRONMENT FOR PRODUCING WELL-ENGINEERED MICROCOMPUTER SOFTWARE.
EBELING C	0038	DESIGN OF A USER MICROPROGRAMMING SUPPORT SYSTEM.
ELLIS JR	0028	A LISP SHELL.
EMERY HW	0029	THE DEVELOPMENT OF A PARTITIONED SEGMENTED MEMORY MANAGER FOR THE UNIX OPERATING S
ETRA B	0063	PRELIMINARY STEP TOWARDS A LANGUAGE FOR PICTURE CONTROL IN A REAL-TIME MODE.
FAY M	0176	REPORT ON THE PROGRAMMING LANGUAGE PLZ/SYS.
FELDMAN SI	0030	MAKE-A PROGRAM FOR MAINTAINING COMPUTER PROGRAMS.
FITZGERALD, ML	0031	COMPUTER SCIENCE AND TECHNOLOGY: COMMON COMMAND LANGUAGE FOR FILE MANIPULATION AND
FORGACS T	0032	MESS-A MACROLANGUAGE FOR STRUCTURED SYSTEMS PROGRAMMING.
FOURTANIER J-L	0033	OPERATING SYSTEMS IN SHARED TIME-THE UNIX PHENOMENON.
FRASER AG	0034	UNIX TIME-SHARING SYSTEM: CIRCUIT DESIGN AIDS.
GAINES RS	0012	THE MH MESSAGE HANDLING SYSTEM: USER'S MANUAL.
GIBSON TA	0155	CAUTION: STRUCTURED PROGRAMMING CAN BE HABIT-FORMING.
	0156	STRUCTURED PROGRAMMING, C AND TINY C.
GILLOGLY JJ	0035	WORD PROCESSING WITH UNIX.
GREEN SL	0036	IMPLEMENTATION AND PERFORMANCE OF A UNIX LINK.
GROUSE P	0157	'FLOWBLOCKS'-A TECHNIQUE FOR STRUCTURED PROGRAMMING.
GUHA RK	0037	DYNAMIC MICROPROGRAMMING IN A TIME SHARING ENVIRONMENT.
	0038	DESIGN OF A USER MICROPROGRAMMING SUPPORT SYSTEM.
GUTHERY SB	0156	STRUCTURED PROGRAMMING, C AND TINY C.
HAFEMAN DR	0039	DESIGN DESCRIPTION OF THE NOVA 3 CARTRIDGE DISK EMULATOR ON THE STANFORD EMMY SYST
HAIGHT RC	0023	UNIX TIME-SHARING SYSTEM: THE PROGRAMMER'S WORKBENCH.
HAMMOND RA	0158	AN OPTIMIZER FOR A C COMPILER FOR THE SERIES/1.
HAMPEL VE	0040	USE OF SCIENTIFIC DATA WITH THE MASTER CONTROL AND INGRES DATA MANAGEMENT SYSTEMS.
HANCOCK L	0159	IMPLEMENTING A TINY INTERPRETER WITH A CP/M-FLAVORED C.
HANSON DR	0041	A PORTABLE FILE DIRECTORY SYSTEM.
HARDIE PA	0064	GRAPHICS SATELLITE FOR THE UNIX TIME-SHARING SYSTEM.
HARLAND DM	0042	HIGH SPEED DATA ACQUISITION: RUNNING A REALTIME PROCESS AND A TIME-SHARED SYSTEM (
HAVERTY JF	0043	INTER-PROCESS COMMUNICATIONS FOR A SERVER IN UNIX.
HAWLEY JA	0044	MUNIX, A MULTIPROCESSING VERSION OF UNIX.
HAYES KC	0045	PDP 11 IMAGE PROCESSING SOFTWARE.
HELD G	0046	STORAGE STRUCTURES AND ACCESS METHODS IN THE RELATIONAL DATA-BASE MANAGEMENT SYSTE
	0047	INGRES: A RELATIONAL DATA-BASE SYSTEM.
	0125	THE DESIGN AND IMPLEMENTATION OF INGRES.
	0146	EMBEDDING A RELATIONAL DATA SUBLANGUAGE IN A GENERAL PURPOSE PROGRAMMING LANGUAGE.
HENNEGAN NM	0048	GIML REFERENCE MANUAL.
HERMAN M	0045	PDP 11 IMAGE PROCESSING SOFTWARE.
HITCHON CK	0027	AN ENVIRONMENT FOR PRODUCING WELL-ENGINEERED MICROCOMPUTER SOFTWARE.
HOLMGREN SF	0049	RESOURCE SHARING UNIX.
HORTON RE	0050	USING PERSONAL COMPUTERS AS TERMINALS IN COMPUTER NETWORKS.
	0105	A LOCAL NETWORK OF MINI AND MICROCOMPUTERS FOR EXPERIMENT SUPPORT.
HOWARD JE	0051	AN IMPLEMENTATION OF A CODASYL BASED DATA-BASE MANAGEMENT SYSTEM UNDER THE UNIX OP
HOWARD R	0081	A UNIX-BASED LOCAL PROCESSOR AND NETWORK ACCESS MACHINE.
HUCK J	0052	I/O DEVICE EMULATION IN THE STANFORD EMULATION LABORATORY.
ISHIDA H	0054	UNIX-AN EASY-TO-USE OPERATING SYSTEM DEVELOPED BY BELL TELEPHONE LABORATORIES.

WONG E	0126	INGRES - A RELATIONAL DATABASE SYSTEM, FINAL REPORT.
WONG G	0085	PRELIMINARY DESIGN OF INGRES: PART-4.
WONSIEWICZ BC	0141	UNIX TIME-SHARING SYSTEM: MICROCOMPUTER CONTROL OF APPARATUS, MACHINERY, AND EXPER
WOOD JL	0006	A USER'S VIEWPOINT ON THE PROGRAMMER'S WORKBENCH.
WOODWARD JPL	0142	A KERNEL-BASED SECURE UNIX DESIGN.
YORMARK B	0143	ADVANCED TEXT PROCESSING USING UNIX.
ZAHN CT	0181	C NOTES: A GUIDE TO THE C PROGRAMMING LANGUAGE.
ZOOK W, ET AL	0144	INGRES REFERENCE MANUAL, 5.
ZUCKER S	0145	INTERPROCESS COMMUNICATION EXTENSIONS FOR THE UNIX OPERATING SYSTEM, PART-2. IMPLE

THE SAN FRANCISCO SOFTWARE TOOLS AND USENIX MEETINGS IN SUMMARY

SOFTWARE TOOLS and USENIX Meetings
San Francisco, California
January 20 -- January 23, 1981

This report is a summary of a double winter meeting held in San Francisco. It is based on my notes and memories, and as such reflects my personal bias and knowledge. Extensive detail has been deliberately avoided, in the hope of keeping these notes down to a reasonable size.

In general, there is a rapid growth in the size of both these users groups; there were 670 attendees at the USENIX meeting, which is the largest attendance yet. The Software Tools group was also much larger than previous meetings, although I do not have the actual number of attendees.

I cannot guarantee that what is reported here was actually said. If you want to be SURE, or need more information, check with the speaker in question. My apologies to anyone who has been misquoted.

My thanks to the many persons who made informative presentations at the meetings. Further thanks to David Sherman, whose notes and macros from the June '79 conference made easy the production of these notes, to Mike Tilson of HCR who supplied additional information, and to Martin Tourl of DCIEM, who helped clean up some of the spelling and grammatical errors.

David Legg
Defence and Civil Institute of Environmental Medicine
(DCIEM)
PO Box 2000
Downsview, Canada
M3M 3B9
(416) 633-4240 Ext 300

Table of Contents

37	A New Text Formatting Package	M. Kampe
38	Terminal Independent CRT Software	M. Horton
39	TROLL: A Compact Relational System	A. Wasserman
40	A Database Application Design System	M. Meyer
41	INGRES: Status and Directions	E. Allman
42	Environmental Technical Info. System	C. Corben
43	The "Draw" Circuit Design System	S. Bourne
44	Unix as a Large Application Base	M. Tilson
45	Compiler Error Analysis for Fast Debugging	R. Henry
46	A Pascal Compiler for the VAX	P. Kessler
47	Tcheck: a file system tree checker	J. Thompson
48	UNIX Aides for English Courses	J. Joyce
49	Nroff macros package	G. Watt, jr

SOFTWARE TOOLS USERS GROUP MEETING
TUESDAY MORNING
Chair: Debbie Scherrer, Lawrence Berkeley Laboratory

Speaker 1 9:00 a.m. **Opening Remarks**

Debbie Scherrer
Lawrence Berkeley Laboratory

Debbie called the meeting to order, welcomed the attendees, and then introduced the first speaker after making a few orgaizational comments.

Speaker 2 9:13 am **The Next Generation of Software Tools**

Tony I. Wasserman
UCSF and UCB

Tony first described briefly the common problems of software design, development and maintainence experienced by most installations, regardless of the system in use. These include the ideas of compatability, capability, and uniformity between different software products. To overcome these problems, he suggested a set of tools, which would employ a common methodology, and encompass the entire software life-cycle. The tools would be combined to form a "tool kit". This "tool kit" could also include a database, which would allow the tools to be "customized" to individual preference, while still maintaining the the desired methodology. The database could also be used to capture information about program structure, design decisions, and the software's life cycle its self, all on a real time basis. The tools themselves could provide for sophisticated human interfaces, such as graphical and voice oriented I/O operations.

Some of the desired characteristics of a tool could be: singularity of use - a text editor would not include text formatter function, ease of use - not obscure for a novice, and not frustrating for a wizard. self documenting - will try to help if possible, consistent - each tool will not be totally unique to approach.

The guidelines suggested to achieve this are as follows. First, the documentation must be online. Second, at least a two level interface, novice and wizard. Third, a syntax driven menu command interface. Fourth, the 10% *basic capability* should be identified, while the remaining 90% should be *extended capability*.

Mr. Wasserman suggested that in the future, we might see PPHS (professional programmer based systems), which would have approximately the power and space of an 11/70 including a floppy drive, for every programmer. These could provide for a work at home situation with dial up or network connections for file transfer, mail, etc.

Speaker 3 9:52 am **Virtual Operating System**

Dennis Hall
Lawrence Berkeley Laboratory

Dennis gave a very brief introduction to Virtual Operating Systems, and then proceeded to list a number of ideas which he felt should be followed in designing a virtual system. The two basic ideas, which should be done in parallel, were to design and develop utilities top down, and to design and develop the virtual machine bottom up. A couple of suggested rules were to "imitate success", and to "inovate with caution".

Speaker 4 10:07 am **Virtual Aethers**

Joseph Sventek
Lawrence Berkeley Laboratory

Joseph described the concept and use of virtual aethers. Virtual aethers are basically an implementation of the Ether net concept in software, and used for communication between a number of processes, rather than a number of processors. In such a scheme all processes tapped into the aether see all of the messages. It would include a protection scheme to prevent unauthorized access to the aether, but would allow any process to communicate a message to the owner process of the aether to request the key required to tap into the aether. If the owner process terminates, the aether survives, but no new process can connect, since there is no one to request the key from. A paper describing virtual aethers is available from Joseph, if you send him a self addressed stamped envelope.

Speaker 5 10:29 am **Portability Layers of Graphics Software**

Mike O'Dell
Lawrence Berkeley Laboratory

Mike proposed extending the software tools notion to the area of graphics. This could provide some insulation from the underlying graphics system, making the software more portable, and giving a common interface to differing types of hardware. The largest impact would be felt at the "middle layer" of software, the utility programs. For instance, there could be routines for drawing bar charts, pie sections, and other common sub-components of graphics output. This level could also set conventions for interchange of data both to higher and to lower level routines. Obviously setting this up would be a great deal of work, but each individual routine would be reasonably simple. Mike's opinon is that it is still too early to start saying that one convention is THE STANDARD, and that we should instead be looking at throwaway test implementations to determine from experience what is, and what is not, good.

Speaker 6 10:49 am **IEEE Micro Operating Systms Interface Standards**

Sam Kirk
TRW/VIDAR

Sam described the efforts of the IEEE to define a standard operating system for micro computers, one which would be implementable on different micro's, but which provide the same interface to the outside world and to other systems. He felt that the standards should be defined on the basis of experience, and not defined out of thin air by a commitee.

---- BREAK ----

Chair: David Martin, Hughes Aircraft Corporation

Speaker 7 11:34 am **Toward a More Interactive Shell**

David Martin
Hughes Aircraft Co.

David described some enhancements made to the shell to make it easier to use, especially for poor typists. These enhancements attemp to obtain the following results: reduce unnecessary keystrokes, reduce file name spelling errors, provide for command resubmittal, and command editing. There were only a few additions made to achieve the above goal. One of the most interesting was the addition of a ^F function which allows for the automatic completion of a partially specified filename by the shell. The shell will pick try to match the partially typed filename to that of a file in the directory. If no match is achieved, then this is reported. If multiple matches are possible, it chooses the longest possible match, and then asks for confirmation of its choice. A negative responce to the request for confirmation result in all the possible matches being displayed in a menu format, with the user being able to select a menu item by number. Other features include retrieval of the last command entered, and the ability to edit and re-enter the command without the need to completely retype the command. The editing feature uses VI type commands, but works only on a single line.

Speaker 8 12:04 pm **A Virtual Terminal Handler**

Allen Aiken
Georgia Institute of Technology

The description of a virtual terminal handler was very interesting. The basic concept is to hide differences in individual terminal types, while not losing the ability to access specific features of a given terminal. In other words, a very dumb terminal with minimum capability would seem to be smarter, while special features of a very smart terminal would still be accessable. The required minimum would include a non-destructive way of moving the cursor, and some basic clear capability. The advantages are clear -- the ability to write software which is terminal independant, and the ability to run old software on new equipment with a minimum of effort. It will even handle hard copy terminals, by treating them as a one line CRT type terminal.

Speaker 9 12:37 pm **Product Development and the Software Tools**

Edward G. Happ
Interactive Data Corp.

Edward described the implementation of an investment analysis system written in Ratfor. His company chose Ratfor for a number of reasons, such as compatability with existing fortran system, portability, as well as product oriented control structures. The existing Ratfor subroutine library also provided a good base from which to build. Although this system is not a small system, it was completed in only a few months from the start of the project, much faster than if it had been written from scratch.

---- LUNCH ----

Chair: Joseph Sventek, Lawrence Berkeley Laboraory

Speaker 10 2:00 pm **Implementation Issues**

This session consisted of a number of short discussions on the implementation of the software tools on a number of different machines and operating systems, and the problems encountered by the installers. Since the length of each talk was very limited, the speakers packed a lot of detail into a very short space, and it was impossible to get any notes without missing large portions of the rest of the talk. the list of speakers, and their topics were as follows.

LTSS, CTSS on CDC7600, CRAY-1
Margaret Hug, Los Alamos Scientific Laboratory
MAX4 on Modcomp IV
Bob Upshaw, Lawrence Berkeley Laboratory
CP/M on Z80
Philip Scherrer, Unicorn Systems
MPX on SEL
Walt Donovan, NASA/Ames Research Center
RSX-11M on VAX-11
Joseph Sventek, Lawrence Berkeley Laboratory
VMS on VAX-11
Joseph Sventek, Lawrence Berkeley Laboratory

---- BREAK ----

Speaker 11 4:00 pm **Panel Discussion**

Allen Akin, Georgia Institute of Technology
Skip Egdorf, Los Alamos Scientific Laboratory
Mike O'Dell, Lawrence Berkeley Laboratory
Debbie Scherrer, Lawrence Berkeley Laboratory

The above members of the panel raised a number of points for discussion. These included such topics as the future of the group itself, the direction the group should take in the future, whether or not to standardize some of the primitives, etc.

On the question of standardization, it was decided that the time was not yet right, in fact there was some discussion as to whether standards were a good thing, and whether it might actually be detrimental to standardize. In support of standards, it was pointed out how much easier many things were because of the ASCII and RS232 standards. It was agreed however, that consistency should be maintained in the primitives.

It was also suggested that the group act as a clearing house on information on how to bring up the primitives on different systems, but it was pointed out that to do this required the co-operation of the members to supply this information in the first place. The idea was suggested that new and improved versions of the tools be supplied because what might be perfect for one application, might be very poor for another, and vice versa.

It was also proposed to look into making the mailing list available on microfiche because of the size and cost of the hardcopy version. It was noted that most people would be able to get acess to microfiche equipment.

WEDNESDAY JANUARY 21 MORNING SESSION
Chair: Thomas Ferrin, UCSF

Speaker 1 10:03 am **Opening Comments**

Thomas Ferrin
UCSF

Thomas called the meeting to order, welcomed the attendees to San Francisco, and then made a number of announcements. The only announcement of interest here, was a call for a show of hands on whether to include attendees phone numbers with their addresses in the list of attendees to be published, the result was a unanimous "yes".

Speaker 2 10:12 am **Evolution of the UNIX Timesharing System**

D. Ritchie
Bell Telephone Labs

Dennis gave a talk on the evolution of UNIX, from its inception on a PDP 7, to the present. Since the talk was a repetition of the history of UNIX, and did not contain any new information, I did not take many notes. What was of more interest, was the question and answer period which followed.

Q: What is planned for the future design of UNIX?.
A: There is some interest in designing a network to make several machines appear as one, however this will probably use our internal network and hardware, and will not be released.
Q: What was the origin of the name of the "dsw" program?
A: "dsw" stands for delete from switches, and is a carry over from a program which was originally typed in on the switches of the PDP 7.
Q: What year was UNIX first moved onto a PDP 11, and when was it written in C?
A: UNIX was moved to the PDP 11's in the winter of 1971, and was translated to C in the summer of 1973, between June and August.
Q: Did you influence the RSX idea of inodes?
A: Possibly, although I had no direct input to the design of the system.
Q: Is there going to be a new standard for "C"?
A: Yes, there is one currently in draft form. Any changes will not be too suprising.
Q: When did the various versions occur?
A: I cannot give exact dates, they were produced as and when there was a reason to release a new version.
Q: It appears that five years from now, everyone will be running some version of UNIX, will there be some sort of standard to adhere to?
A: No.
Q: Can people use the V7 C compiler without a V7 license.
A: Please wait until Larry Isley speaks, he will deal with that question.

Speaker 3 11:18 am **What's Happening at BTL/Western Electric**

L. Isley
Western Electric

Larry had very little to say, but what he did say was very important to a number of sites. He made only one announcement, that was that Western Electric had decided to allow any particular site to standardize on a single version of the C compiler for which that site was licensed. That is, a site with licenses for a number of differing UNIX systems, could pick a single version of the C compiler and run it on all of the systems, not just its original system. What is *NOT* permissible is to run a v7 or phototypesetter compiler at a site with licenses for only v8 systems. He also stated that Western would be increasing the amount of auditing they would be carrying out in the future.

The total number of licenses and installations is as follows:

	Licenses	Installations
Commercial	170	287
Governmental	90	197
Educational	608	1524
Administrative	17	51
Total	879	2059

There are also between 200 and 300 internal Bell sites.

The committee to review software packages has been revised, and should prove to be much more effective than its predecessor. It will meet as required, but at least once a month, with the first meeting scheduled for the 29 of January, 1901. It will be looking at a total of eight items at this time, and should be able to finish up the backlog at the next meeting. Hopefully it will also be able to speed up the issuing of licenses, and cut the delay to two to three weeks.

There was no time for a question and answer period, which disappointed many people. This was raised again in the final business meeting, and Western should get a bigger time slice at the next meeting.

Speaker 4 11:34 am **What's Happening at DEC**

Bill Munson
DEC

Bill announced that people are still buying UNIX for DEC machines in spite of the DEC sales force, (a fact we already know), and that DEC is finally starting to realize the advantage of having inhouse knowledge of UNIX. Bill is part of a group pushing UNIX from within DEC. It is now possible for field service personnel to recieve training on UNIX, and he urged that we tell our field service representatives to contact his group to find out more about what is available to them. He said he could not talk about the C compiler they don't have, and that a tape may be announced on March 2 of this year. He also read from a sales folder for an ML-11, the description of the hardware, and how it could be used by a UNIX system. He said that the sales pitch did not mention a DEC operating system once, only the UNIX operating sytem. (one for UNIX).

Speaker 5 11:47 am **DARPA VAX/UNIX support effort**

R. Fabry
UCB

Mr. Fabry gave an brief report on the current status of the VAX UNIX support project. He stated that almost every site using UNIX on a VAX has a Berkeley license, and that the budget for the group was 2/3 of a million dollars. Some of the features of the Berkeley VAX UNIX are the ability to give fast access to a large (30 Mb) file with very little computation, the ability to couple files to the address space including the ability to share files, and improved interprocess communication. A couple of other goals include interfacing a BBN type ARPAnet with Ethernet type local networks, and improved performance. They also intend to preserve the simplicity and elegance of the original UNIX throughout all of this work. For information on the 4BSD distribution contact:

Laura Tong
C.S.R.G
Computer Science
University of California at Berkeley,
California,
U.S.A.
CA 94720

phone: (415) 642-7780
network: CSRG@BERKELEY

Other Efforts

There are a number of other areas in which UCB is interested, although the major effort is the one described above. Some of the things that are being attempted are:

- improved FORTRAN speed, (currently 2-2.5 times slower than VMS FORTRAN)
- the UUCP "network"
- the kernel configuration
- eliminating the linear search of directories
- screen management, a window shell
- SDB, a symbolic debugger
- line discipline, need a stackable one
- kernel buffering
- software controls over networks
- looking for software from other sources.

There followed a number of questions from the audience:

Q: Do you have UNIX on the VAX 11/750?
A: Yes.
Q: Can people using 11/40's and 11/70's get these things?
A: Yes some of these are available through 2BSD which has been updated.
Q: How much of this is available for other versions of UNIX?
A: This is available for ONYX and for the C machine. For IDRIS and COHERENT, I failed to catch the answer (Dave).

A new version of INGRES will be on the 4BSD tape. The time frame for the network system will be, hopefully initial release by the end of January, and a formal release by the end of the year.

---- LUNCH ----

Speaker 6 1:08 pm **Business Meeting 1**

Lou Katz
USENIX

The following are a list of the major points of the first of the two business meetings.

- a referendum on recruitment at meetings will be issued with the dues notices.
- the association has been incorporated, and has an office in New York, the address is:

USENIX
Box 8
Rockefellar University,
1230 York Ave.
N.Y., NY 10021

phone: (213) 360-1102 preferably 9:00 am - noon, 1:00 pm - 5:00 pm Eastern time

- 158 distribution tapes have been mailed.
- the dues are the same for 81 as for 80.
- renewal forms will be mailed in January.
- 2 tapes authorized, bugeted for 3 tapes in 81
- Harvard will be a source for manuals for V6, V7, and PWB, and will have one for 4BSD in a couple of months.
- There will be 10 newletters in 81, mail 1st class in North America, and airmail overseas.
- we need articles for the newletter, please deliver by electronic means if possible. Send to Wally Wedel at the University of Texas. It is possible to dial in to their system and use a login specifically for the login newsletter. The login name is "login", and the password is "usenix". The phone number is (512) 474-5511.

The next meeting is scheduled for the 24-26 June 1981, at the University of Texas. The following one will be in Santa Monica, and the one after that will probably be in the Boston area. For information on the Austin conference, contact Wally Wedel, at the Computation Centre, University of Texas, Austin, Tx 78712 or on the network as wedel@utexas-11. His phone number is (512) 471-3241.

It is hoped that starting with the Santa Monica conference, that the location of the conference will alternate between the east and west coasts, to try to distribute the travel expenses evenly.

Chair: Sam Leffler, Sytek Inc.

Speaker 1 1:35 pm **C/70 Micro-machine Hardware Overview**

A. Nemeth
BBN

Mr. Nemeth discussed the background of the development of the C machine, including the requirements and goals of the design, from its inception to its current state. Some of the requirements were that it had to support existing applications, it had to be a base for a standalone system, and it should be a source of inexpensive computer cycles. From the hardware point of view, it had to be flexible and easy to adapt. It also had to be simple, in architecture, in programming, and in construction. The ability to extend its capability in the future was also a major factor in its design. The design incorporates a large address space (20 bit addresses), and microprogramming. The word size is also 20 bits, and the process space also uses the full 20 bit address space.

Speaker 2 1:52 pm **C/70 Macro Architecture**

C. Howe
BBN

Mr. Howe described how the instruction set of the C/70 was designed, and what its characteristics are. Some of the basic requirements were an address space greater than 16 bits, the ability to efficiently handle "C" data types, the ability to handle more register variables, fast and effecient subroutine and function linking, direct access to basic machine functions from "C", and compact encoding of instructions.

The C/70 is a stack machine, with the top of the stack in registers, and automatic procedures for moving the bottom of stack in and out of memory as required. An instruction can take from 1 to 4 words, with 19 addressing modes.

Speaker 3 2:10 pm **Porting UNIX to the C/70**

A. Nemeth
BBN

Mr. Nemeth described the implementation of UNIX V7, with a couple of additions, on a C/70. The additions consisted of a cursor oriented editor, and some network capability, including file transfers, and a message system. The major areas of concentration were the kernel, memory management, rapid switching, and swapping as opposed to paging. Interrupts are directed to "C" subroutines for handling with the arguments giving the trap type or system call id, the program counter, and the stack pointer. Allowance also had to be made for the 20 bit disk addresses, which is room for approximately a 635 Mbyte drive, and for the differences in byte size, 10 bit as opposed to 8 bit. Changes were made in the initialization process, and in communications with other users. In total only three modules were rewriten (mchdep.c, trap.c, and ureg.c), three modules had extensive (10-20 lines) modified, nine had minor changes (2-3 lines)

made, twelve modules were unchanged, five drivers were written, and conf.c had to be rewritten, for a total of thirty six modules. There was no need of a mch.s module. A number of steps were taken to debug the system, and then the user programs were ported to the new system.

In the future BBN will be looking at moving more critical routines to the hardware, supporting local networks, bit map displays, multiprocessor UNIX, and a high relibility UNIX.

On the question of building other machines using the same concept, it was noted that FORTRAN 77 uses C intermediate code under UNIX, and that PASCAL might be a possible candidate, but that it really depends on the language under consideration as to whether or not it is feasible. It was noted that work was under way on supporting tape drives for future addition to the system.

Speaker 4 2:33 pm **LOCUS: The UCLA Distrubuted System**

Bruce Walker
UCLA

The LOCUS system is a distributed UNIX system architecture, on co-operating local UNIX systems. This is not an addon, but is imbedded directly in the kernel. It is theoretically possible to run a UNIX system with no disks, using the network to supply the file storage capability in place of local disks. The architecture goals were as follows: compatability with existing UNIX application code, locally autonomous, heterogenous, extensible, highly reliable, good performance, and network transparency. A process need not know the location of files and other processes. For instance, files have the same pathname from all location within the network, therefore their location is transparent to applications, users, and to the systems themselves. This was achieved through the use of global file system numbers, and globally unique low level file names (made up of file system number and inode number). The directories were unchanged, and a system wide mount table was added, in addition to (and not replacing) the local mount table. Synchronization plays a very important part in the success of the system.

---- BREAK ----

Chair: William Joy, UCB

Speaker 5 3:31 pm **C on the FPS-164**

K. Wilson
Cornell University

Mr. Wilson gave some statistic on the FPS AP-190C using an IBM 370-168 as host. The array processor is from one half (FORTRAN) to 8 (bit twiddling) times as fast as the IBM. It has a thirty eight bit word length, ninety six thousand words of memory, and costs about one hundred and thirty thousand dollars. It can handle either FORTRAN or assembler. Its major uses are in galaxy dynamics, monte carlo simulation, molecular dynamics, and band structure analysis.

The FPS-164 will be released soon, it will have a sixty four bit word, up to one and a half million words storage, and will cost from two hundred to six hundred thousand dollars. Its first host will be a VAX, but an IBM host is promised. It will have a full FORTRAN 77 compiler. In responce to questions Mr. Wilson stated that the FPS-120B is the same internally as the FPS-190C, and that they have done nothing about getting drivers for the array processor, as the first step is to get a cross compiler for it.

Speaker 6 3:46 pm **A Truly Portable I/O Library**

D. Strict
University of Pittsburg

Mr. Strict gave a very brief overview of the different portable I/O libraries under the various versions of UNIX, now in common use. Under V6, the assembler routines are in libc, and the C routines are in libp. Under V7, libs has been merged into libc. Some desirable features of the portable I/O library are to have I/O access via a single I/O stream, and to have concurrent independent sequential access to a file.

Speaker 7 4:00 pm **IBM Front End to Hasp Multileaving**

Peter Hardie
University of Saskatchewan

Peter described the DEUS system at the University of Saskatchewan. DEUS is a front end to HASP for the submission of student jobs. The hardware involved is a PDP11/70 with three DH-11's, two RM03's, a DQ-11 and a dual density tape drive. There are forty student terminals, four data entry operators, an optical scanner, a line printer, and two other terminals attached to the system. Communication to the IBM system is via a 1600 baud synchronous line from the DQ-11. The students enter the jobs via the terminals, it send them down to the IBM for execution, HASP returns the jobs to the 70, and DEUS takes care of the printing. DEUS also takes care of limiting logon time and disk storage space.

Speaker 8 4:24 pm **DEAFnet**

K. Harrenstein
SRI

Mr. Harrenstein gave us an insight to some of the problems of the deaf community. For instance, the common telephone is of no direct use -- something must be attached to allow visual data to be transmitted. The advantage's to the deaf of a network are clear: the ability to communicate with other people easily, and to leave messages for people who are not "home" when someone trys to contact them.

UNIX was chosen for a number of reasons, one being that the PDP 11 family is a good model for a nationwide network, another is that UNIX is flexible, it can be modified easily, and a lot of software was already available to make the base of the system.

One of the problems encountered was that the user community was generally very computer naive. Another was that many deaf people have a telex type terminal which

uses the five bit Baudot code, and cannot afford to buy a new ASCII terminal; therefore the system had to be capable of accepting Baudot code as well as ASCII.

The final result is that the network is up and running to everyone's delight, and there are plans to extend it. As a point of interest, the state of California has approved a law to the effect that the phone company must supply a terminal capable of both ASCII and Baudot to all deaf customers at no cost to the customer. This is a major achievement, which will be of great benifit to the deaf community.

Speaker 9 4:50 pm **ANGUS**

David Tilbrook
BNR

The foundation of ANGUS is taken from UNIX, MASCOT, and TIPS. ANGUS provides a standard representation for data, and a library of routines for the manipulation and extraction of that data. It can also provide menu files for enhancement of the shell to make it more "friendly" to novice users.

THURSDAY JANUARY 22 1981
Chair: Michael O'Dell, Lawrence Berkeley Laboratory

Speaker 1 9:00 am **Compact and Simple Kernel Overlays**

K. Harrenstein
SRI

Mr. Harrenstein described his implementation of kernel overlays, which allows all but the process switching module to be overlayed. The individual overlays are created by the loader, under user control. The user must specify to the loader what to put into each overlay. This is done by means of a "-k" flag, which tells the loader to start a new overlay. This will be public domain and released if possible. It is possible to use this on a 11/70.

Speaker 2 9:21 am **V7 Conversion Tools**

Mike Tilson
HCR

Mike described some of the problem encountered when trying to bring up V7 of UNIX. These included the kernel size if the host is a small machine, I and D space differnces, utilities, floating point, and user migration and code conversion. There are a number of ways of making the kernel fit on a small machine. The one chosen by Mike was to use kernel overlays. The features of this overlay scheme are as follows. Data is never mapped, only text is mapped, no restriction on the calling order of routines, and it is totally automatic. To implement this scheme required changing three lines of "C" code, and about sixty lines of assembler code. There are a few restrictions inherent in this scheme. One is that data plus the bss section must be less than thirty two kbytes long. No object file may be more than 8k bytes long, and the machine assist and main routines must be resident in memory. The overhead involved is increased about two and a half times for mapped function, or six bytes per mapped function.

Speaker 3 9:38 am **USGS/UCB V7 System**

W. Jolitz
USGS

Bill listed a number of improvements and changes they had made to the regular V7 system. These included using a one kbyte filestystem, which gave very much improved file system performance. The kernel was changed to include an eight to sixteen k byte clist, and more buffer space. Both the buffer and the inode tables were hashed, which also added to performance. New features of the system include local averages of system use taken at one, three, and five minute intervals, text overlays, a TENEX styled tty driver,and multiple line disciplines. On the administrative side, the changes include dynamic file quotas, special /tmp file protection, expanded accounting, performance monitoring, and privileged accounts. One final change was made to limit the number of process by process group. There is a number of further efforts underway.

Speaker 4 9:52 am **Multi-controller disk driver**

P. Staubach
University of Oklahoma

Mr. Staubach gave a very quick description of a way of using multilple controllers for multiple devices, in which the required extra information is contained in the minor device number. The first two digits of the minor device number specify the controller, the next two specify the device, and the last four specify the filesystem on that device. In responce to a question, Mr. Staubach replied that they had not implemented any bad sector recovery, but that they were working on it.

Speaker 5 10:01 am **Floating point save problems**

J. Reeds
University of Oklahoma

Mr. Reeds described some problems discovered in relation to the floating point processor. The fixes for these problems are included on the UCB tape. They involve changes to mch.s, user.h, trap.c, sig.c, sysent.c, and syslocal.c. The basic problem is that the kernel floating point exceptions (FPE) are mis-scheduled, and sometimes even get lost. This is due to two problems, one is that the floating point error registers are read only, and cannot be saved and restored, and the second is that only one signal is serviced for each entry to the kernel. Users using stst type instructions can expect to see crosstalk between processes. The solution is to force an extra step via the job scheduler whenever a FPE occurs.

Speaker 6 10:17 am **Real time I/O using LPA-11**

A. Romberger
UCB

Mr. Romberger described the operation of an LPA-11 to handle the data buffering of raw data coming in in real time, and passing it upon request to the host CPU. The device always has at least two buffers active for a process, the one recieving the data, and the one to fill up when the current one is full. Although the device only sees two buffers at any one time, it is possible to have up to eight buffers in the queue of buffers. The device is flexible according to Mr. Romberger, and can have an address space of eighteen bits. [Sites using LPA-11's are advised that there were bugs in earlier versions of the LPA microcode, and ECO's exist to fix some problems.]

---- BREAK ----

Speaker 7 11:00 am **Terminal Linking Line discipline**

S. Leffler
Sytek Inc.

The line linking disiplines described by Sam allow the connection of one terminal I/O queues to those of another terminal, in a number of configurations. This could be used in a number of ways, a simple link, where the input of one terminal is also copied to the input of another terminal, and vice versa for the output. This set up allows for the monitoring on one terminal the actions of a user on another terminal, very useful in a teaching situation, since each student could see on their own screens whatever the instructor is typing. It can be used across a network, so the input to the local terminal is diverted to the remote system, and the output is returned. This is hard to describe without the use of diagrams, but a paper describing it will be available on the UUCP network.

This is a V7 line discipline, which requires mods to tty.c to expand definitions, and a bug fix to dh.c. It also requires the additions for the line disiplines themselves. One fact that should be noted is that all of the code is in the local system when working with a remote processor, so that it is not necessary for the system you are talking to to know the line disipline in use.

Speaker 8 11:17 am **Low cost Terminal MUX**

Mike Tilson
HCR

A low cost terminal multiplexor based upon a 6800 microprocessor was described. The 6800 communicates to UNIX via a single 9600 baud line. Up to 8 serial ports may be connected to the 6800. A special UNIX driver cooperates with the 6800 to provide line multiplesxxing and data compression. Future plans include movement of UNIX TTY handling code into the 6800.

Speaker 9 11:30 am **Multi-host Terminal Front End**

R. Broersma
NOSC

Mr. Broersma described a method of connecting many terminals to many hosts. This system will accomodate user preferences, and will keep terminal profiles. It is adaptable to new computers, new terminal types, and to increased numbers of both terminals and host computers. It uses the idea of a microport. A microport is really a microprocessor (Z80), and a UART, which communicates via a sixteen bit data bus with a modem which communicates to the appropiate host. The advantages of this system, are fewer cables, more available ports, and reduced load from the terminals on the CPU. The microport can handle character translation, echo, tab expansion, and line gathering and editing. The microport also handle's speed control, and knows about the differing terminal characteristics. The cost to make a microport board is around three hundred dollars, and there are plans to release this.

Speaker 10 11:49 am **A Crash-resistant UNIX File system**

William Joy
UCB

Bill described the modification necessary to insure better file system reliability. The current problems are a lack of error checking in V6, and a lack of full ECC in V7. These problems are aggravated by using the 1k byte filesystem described above. Delayed I/O also causes problems and can lead to inconsistencies. For example, if indexed data is created after the indexing data, then for a short time the indexing is pointing to garbage, and if a crash should occur at just this time, well .... The improved version provides for automatic repair and for interactive repair. The automatic repair tend to the conservative side when it has to make a decision, but the associated checks are run in parallel to increase speed.

Some of the remaining problems are: the system continues after a disk failure, which can lead to a spreading "cancer". Unmounting and remounting a different volume can load to the new volume being logically "smashed" if the logical filesystems are not unmounted before unmounting the volume. This is due to the fact that the disk cache isn't flushed when a file system is unmounted. Performance is really unchanged, except to remove a file takes longer than it did before, otherewise there is no real degredation. Some of these problems were fixed in V7, and the rest in the Berkeley VAX system.

In conclusion, one need no longer say that UNIX needs babysitting, or that it has an unreliable file system. George Gable (Purdue, EECS) has the appropriate fixes for V6.

Chair: Michael O'Brien, RAND corp.

Speaker 1 13:02 am **Eunice; UNIX emulation on VMS**

D. Kashtan
SRI

Eunice is a way of porting UNIX programs to the VAX VMS system. It provides an efficient emulation of UNIX system calls. It minimizes the amount of source code modification to user programs. It does require the re-compilation and relinking of programs because the loader is realy a translator which call the VMS loader. As to the file system, true UNIX file names can be used, but "/" and "/dev" are faked, and cannot be chdir'ed to. All of the signals are by SIGPIPE, and the system calls are from 4BSD. The files are stored as VMS files, with the associated variation in types, but all appear the same to UNIX users. The vfork call has been implemented. Just as a note, ls and lseek work correctly without any modification.

Speaker 2 1:22 pm **Vax/UNIX Enhancements and Directions**

W. Joy
UCB

Bill described a way of making seeks shorter by distributing the inodes throughout the file system; a section of inodes followed by a section of data blocks make up a unit called a control group, and a number of control groups make up a file system. There is work underway on user level block allocation schemes, ETHERNET technology, ARPAnet work, network line disciplines, better buffering facilities, and other areas.

Speaker 3 1:50 pm **11/750: Comet Haley or Kohoutek?**

R. Pike
BTL

The VAX 11/750, or the Comet as it is commonly known, was described from a user point of view by Rob. Some of his observations were: there are/were some microcode problems, microcode will loop if a write error occurs when the translation buffer is empty, power control is noisy, RP07's should be avoided, the servo system is unstable, they only have a sixty six byte buffer, and the TS-11 is hard to mount tapes on and has a skew problem. On the whole, he felt he liked the system despite the problems.

Speaker 4 2:02 pm **VAX News From DEC**

A. Stettner
DEC

Mr. Stettner made a few comments about the VAX and Comet systems.

Speaker 5 2:10 pm **VAX Roundtable**

D. Kashtan, SRI
W. Joy, UCB
R. Pike, BTL
A. Stettner, DEC
R. Kridle

The comments at the start of the panel discussion basically said most people were fairly happy with DEC as a whole, but would like to be able to mix and match equipment more easily. The other interesting fact to come out was that one VAX has blown up (power supply problem), one had a core melt down (memory board fire/failure), and one suffered from a chronic reboot problem for a while (hardware failure). The floor was then opened for a question and answer period, but there were no question, so we took an early break.

--- BREAK ---

Chair: Bruce Borden, 3Com Corp.

Speaker 6 3:00 pm **VAX/UNIX ARPA-net Support Project**

R. Gurwitz
BBN

Mr. Gurwitz described the work going on to connect VAX/UNIX to packet switched networks, in particular the ARPA-net. They have implemented a TCP transmission protocol, and an IP internet protocol. The TCP handles the handshaking, the sequence numbers, and the checksums, while the IP handles the internet addressing, the fragmentation and reassembly of the messages, and the internet or gateway connection. It has been implemented as a part of the UNIX kernel, and the interface consists of the standard UNIX file I/O processes. The design goals were to maximize network throughput, minimize queueing between levels, minimize the copying of data, have a low process overhead, minimize changes to kernel modules. The size is eight to ten k bytes without the page tables.

Speaker 7 3:35 pm **An IP/TCP Network Front End**

J. Mullen
Mitre Corp.

The idea of the front end is to reduce the drain on the host resources, and to make it simpler to connect a host to a network. This is accomplished by moving the network specific protocols to the front end and use a "network access protocol" to communicate between the front end and the host. The network access protocol is not network dependent.

Speaker 8 3:55 pm **The RAND Network Front End**

M. Wahrman
RAND Corp.

Mr. Wahrman spoke for only five minutes, so the information came thick and fast. What he basically said was that in their implementation of FENCP, all of the modifications made to the system were to the device drivers, and that the results were successful. For further information, contact Mr. Wahrman, or if using the network contact:

mike@rand-unix
or group@rand-unix

Speaker 9 4:17 pm **Porting UNIX to the Series 1**

P. J. Jalics
CSU

There were a number of reasons that UNIX was taken to the Series 1, to verify the claim of transportability, to add to the base of software transportability, to gain familiarity with the C compiler and UNIX, and to make use of a Series 1 which was available. The characteristics of a Series 1 are: it is a sixteen bit mini, with eight registers, of which only three are really useful, it is byte addressable, it can handle up to two hundred fifty six k bytes of memory, it has separate I and D spaces. It has "reasonable" disks, diskettes, and printer, although it doesn't know about full duplex communications. Of the available operating systems, EDX is a qualified disaster, and RPS is an unqualified disaster. Mr. Jalics then went on to describe the steps he followed to bring up UNIX on the Series 1.

FRIDAY, JANUARY 23, 1981

Chair: Eric Allman, UCB

Speaker 1 9:05 am **Device Independent Screen Editing**

G. Aikens
Owl Associates

Mr. Aikens outlined the desirable characteristics of a screen editor. They are:
- It should process clear text files
- It should preserve spatial relationships
- No other tools should be needed for most applications
- It should use the minimum number of special keys
- It should be portable
- It should be device independent
- It should be easy to enhance and to debug
- It should not be a complete text formatter
- It should not be a catch-all software package

The product of this project was an editor which uses the screen as a window onto the file, and is capable of handling multiple files. The equipment description file is pure ascii text, no "funny" characters, it contains such information as the display size, the formatting characteristics, and input and output micro substitutions. The biggest problem were one of inadequate hardware, and operating systems which insist on processing the data they handle. It does not have any online help features, as yet. It requires a terminal with some minimal capabilities: an addressable cursor, and the ability to move up, down, left, and right non-destructively.

Speaker 2 9:28 am **Network Independent Message System**

K. Auerbach
Interactive Systems Corp.

Mr. Auerbach described a network mail system, which is independent of the network type, and seems to provide a lot of functionality. Some of the points made were:
- can use a wide variety of communications facilities
- may be configured to meet local and/or personal needs
- may be suplemented by local software - easily portable to new hosts and operating systems.
- has a feature to catalogue and redistribute incoming mail
- uses asynchronous delivery
- uses full name addressing
- can use aliases for recipients
- automatic redirection to recipients host
- delivery mode may be redefined
- implements urgency specifications
- interactive error correction and return to sender features
- has carbon copy and blind copy capabilities
- From and sender header lists
- can have automatic file copies made

- public and private mailing lists
- nested mailing lists
- elimination of most duplicate deliveries

It is currently running on VAX, 11/70, 11/60, 11/34, and ONYX computers, using X-25, INcard, and other protocols under IS-1, VMS, and ONYX operating systems. This is a rather heterogeneous network, as you can see.

Speaker 3 9:47 am **A New Text Formatting Package**

M. Kampe
Interactive Systems Corp.

Mr. Kampe outlined the operation of a new text formatter which took a year and a half to complete, and looks nothing like nroff or troff. It is device independent, covers a broader range of formatting needs than do nroff and troff, and provides an escape from fundemental nroff and troff problems. It was decided that extending troff was not feasible, due to the problems of performance, maintainability, and extensibility. The complexity of the major macro packages, and the difficulty in mastering the input language of troff were also major reasons for not trying to fix it.

The new formatter uses a completely different approach to the problem. It does not look at the characters individually, but rather treats an abstraction of the text as object to be manipulated, for example to do multi column output, the page width is first defined to be narrow, and then the page formatter routine decides to cut and paste to fill up a page of output, but the text is treated as just a rectangle at this level. The symbolic form of the text is stored as a tree, and operations are carried out on the tree. The last step is to translate the tree back into text for the output. The device deppendent physical output is handled by a postprocessor.

Claims of speed increases of six to fifty times were made for the processor on text files.

Speaker 4 10:18 am **Terminal Independent CRT Software**

M. Horton
UCB

Mr. Horton described a pair of libraries which together allow an application program to talk to any terminal through a fixed interface. The libraries contain the required information on about a hundred and fifty different terminals, although a new terminal can be added in about twenty minutes. The names of these libraries are termlib and termcap.
'Curses' is another library which provides a high level interface to these libraries.

---- BREAK ----

Chair: Roland Johnson, Lawrence Berkeley Laboratory

Speaker 5 11:10 am **TROLL: A Compact Relational System**

A. Wasserman
UCSF

Mr. Wasserman gave a status report on the TROLL system. Version 2 is operational, and version 3 will be operational by/during March of this year. Version 4 will be a multiuser version.

Speaker 6 11:30 am **A Database Application Design System**

M. Meyer
UCB

APPLIDE, an application design environment, provides subtasks to do data input, data storage, manual and automatic transformations, and data output. It provides these functions through the use of menus and forms. It is a layer over the INGRES data base system, to provide a better interface.

Speaker 7 11:47 am **INGRES: Status and Directions**

E. Allman
UCB

Version 6.3 runs on the high end PDP-11 family members with separate I and D spaces. It runs under V7 and is functionally the same as version 6.2. It is in the public domain, and is in the final stages before release. Will probably be released under 2BSD or Bob Kridle's unnumbered V7 distribution. It may possibly be available from the INGRES project at a cost of $150. The VAX version is scheduled for release on March 1, and should appear on 4BSD or 5BSD. The documentation will cost $5 for the intro packet, and $30 for the complete packet, cash in advance.

Future directions:
The PDP-11 versions will be frozen, with no more development done on them. Future VAX version will no longer emphasize compatability with existing versions, and they may not be available outside of the research community. Future research topics may include all or some of the following:
- distributed databases
- hypothetical databases
- exports - to deal with specific data types, ie. FATHER TIME
- artificial intelligence enhancements (eventually)

For availability information and documentation contact:

Tandy Warhow, Project INGRES
Electronics Research Lab,
Cory Hall,
University of California,
Berkeley,
CA 94720

(415) 642-2344

Summary of a Meeting held in Toronto, Feb 6, 1981
Under the Auspices of the Canadian DECUS UNIX SIG

DECUS SOFTWARE TAPE LIBRARY REPORT

Mike Tilson spoke for Steve Pozgaj, who has been representing our SIG at library board meetings. The DECUS tape distribution facilities have always been able to handle non-licenced software; anyone who wants to send in C programs can do so now. Licenced software presents special problems, unless DECUS is licenced for the product in question (or at least authorized to make straight copies). This includes kernel enhancements and modifications to UNIX.

USENIX, on the other hand, *is* licenced to distribute licenced UNIX software, and have a large investment in understanding the existing software submissions, and how they should be organized. USENIX really has been sending out tapes (cumulative distribution number 4). Steve's suggestion is that we let USENIX struggle under the mountain of material and red tape for the time being, rather than setting up an alternative to their efforts.

The problems which some sites have experienced with shipment over the border seem to be less prevalent recently, so the need for a central distribution point in Canada is no longer so pressing.

NEWSLETTER NEWS

Mike Tilson, our editor, invited submissions of material for the newsletter, and pointed out that there have been fewer issues than promised because there were insufficient submissions. There is still in effect a $10. reward for finding a name for our newsletter. Mike would like to see articles of a technical nature submitted. Short articles on new saleable products are OK, provided they aren't simply advertisements. They should be styled along press releases, or better, giving technical insights into the nature of the software. Keep in mind that this is a DECUS UNIX SIG -- articles about UNIX on other CPU's are not entirely welcome.

REPORTS FROM THE USENIX MEETING IN SAN FRANCISCO

Reports were given by Henry Spencer, Mike Tilson, and Dave Lagg. I won't try to repeat everything they said, just some very brief highlights; a copy of notes prepared by Dave and Mike will be published separately.

Bill Munson's group at DEC has been keeping busy; they have a distribution tape which includes 11/44 support for V7 UNIX, and drivers for some of the newer DEC peripherals. Network activity seems to be picking up considerably; many sites are connected to Duke's USENET, and news groups for SCI-FI, Micro-info, etc. are growing.

Mike mentioned that a commercial UNIX group, called '/usr/group', met prior to the USENIX meeting. He suggested that the nature of USENIX meetings has changed -- "We have seen the last hurrah of university hackers, who are becoming an increasingly minor part of the UNIX scene. There are greater signs of vendors and commercial activity; hardware/software demos and hospitality suites are becoming increasingly common."

Other news includes USENIX' plans for two software tapes in 1981, and Rob Pike's experience with a VAX core meltdown, caused by a faulty power supply.

VERSION 7 CONVERSION TOOLS

Mike Tilson outlined tools being used in the conversion of Version 6 UNIX systems to Version 7. (This is the same talk given at the USENIX meeting). The first important point is that an emulation of V6 system calls is possible within a V7 kernel, so that V6 binaries can be run on a V7 system without change. This is important to sites with large unconvertible software (binaries, etc.), and it allows conversion of user programs to take place gradually, rather than overnight. Secondly, Mike has developed an oversized kernel capability, similar to the one distributed by UNSW, but with several advantages. It requires no planning on the part of the user; the system is compiled for separate I&D, and the 'overlayer' takes over from there. It does not require that the user determine the number of arguments to each non-resident subroutine, or decide whether a data item need be in the always-resident segment -- all data is there. Mike Blake-Knox pointed out that there is yet another overlayer available on the Delaware tape -- one which can be used for the kernel, and user programs as well. It is agreed that the slight increase in overhead incurred by these schemes is well worth the ability to run a reasonably large kernel on the non-separate I&D machines.

Martin Tuori
SIG Scribe

For technical information contact:

Eric Allman
(same as for Tandy Warhow)
eric@berkeley
ucbvaxleric

--- LUNCH ---

Chair: Peter Kreps, Lawrence Berkeley Laboratory

Speaker 8 **Environmental Technical Info. System**

C. Corben
USACE

Sorry, I missed this one.

Speaker 9 1:00 pm **The 'Draw' Circuit Design System**

S. Bourne
BTI.

I missed the start of this talk; but apparently, MIT is using the system, and it may be available to universities.

Speaker 10 1:54 pm **UNIX as a Large Application Base**

M. Tilson
HCR

A case study described in which an existing large system under RSX-11 was re-implemented under UNIX. The size of the new implenemmentation (measured in inches of listing thickness) was 10% of the old one. The system supports 60 users accessing a 200MB database with good response (better than RSX) and much higher reliability. The system is now easy to maintain and is portable.

Speaker 11 2:17 pm **Compiler Error Analysis for Fast Debugging**

R. Henry
UCB

This discussion was about the program "error" on the 4BSD tape. It is built to aid in the iterative approach to software development, and is built to run in cooperation with the VI editor. It filters the error messages from a compilation, and inserts them as comments into the source code at the point where the error occured according to the

compiler. It makes some assumptions and allows certain types of messages to the terminal directly, but filters the normal compile errors and is somewhat interactive in deciding what to insert and what to ignore. It will run into trouble if a cascade type error occurs, a human would recognize that certain errors were probably caused by an effect of an earlier error, and ignore the incorrect message, "error" cannot do this.

Speaker 12 **A Pascal Compiler for the VAX**

P. Kessler
UCB

Sorry, no notes for this session.

Speaker 13 **Tcheck: a file system tree checker**

J. Thompson
University of Oklahoma

Tcheck is a program to check the integrity of a file system tree structure. It can detect some errors which dcheck cannot, such as a loop in the tree.

--- BREAK ---

Chair: Michail Wahrman, RAND

Speaker 14 3:30 pm **UNIX Aides for English Courses**

J. Joyce
ITS

The ability of UNIX to aid in the teaching of English lies mainly in the fact that it has a reasonably friendly and straight forward text editing package, as well as text formatters. Mr. Joyce felt that since a student did not have to retype a complete page to correct an erroneous phrase, the student is more apt to take the time to correct it.

Speaker 15 **Nroff macros package**

G. Watt
Air Force Data Service Center

Mr. Watt described how he took the "mm" macros available for both nroff and troff, as separate packages, and combined them into one package, which was about the same size as one of the originals. The macro package is about 5000 lines long.

ADEC

Advanced Digital Engineering Corporation

Box 327 Sub 6, Saskatoon, Canada, S7N 0W0

(306)374-1118

December 15, 1980

The Editor
Canadian UNIX Users Group Newsletter
Human Computing Resources Corporation
10 St. Mary Street
Toronto, Ontario

Dear Sir,

Thank you for the mention in your October issue. I thought I would write to correct the name and supply an address.

We distribute Unix version 7 with a kernel tailored to LSI 11/23 processors. The kernel has an expanded number of disk buffers (20) and provides split I/D emulation (which allows 'lint' and 'f77' to run on the small 11's).

We also distribute an LSI-11 software development package similar to the Satellite Processor System described in the UNIX issue of the BSTJ (Aug 79). In this system, programs running in satellite LSI-11 nodes are fully integrated into the operating system allowing pipes, redirected i/o, interactive debugging with ADB, etc. The LSI-11 programs also can be stand alone (self booting from TU58 tapes), and can use the LSI-11/2 floating-point instructions in-line.

I hope these items will be of interest to your readers.

Yours truly,

Rod Gilchrist

Rod Gilchrist

Biosciences Data Centre
The University of British Columbia
2204 Main Mall
Vancouver, B.C., Canada V6T 1W5

(604) 228-6527

February 3,1981.

The Editor
Canadian UNIX Users Group
Human Computing Resources Corporation
10 St. Mary Street
Toronto
Ontario M4Y 1P9

Dear Group:

I thought that I would mention some of the things that I have been doing since I last wrote to you:

1 I have implemented a program called "hack" that does the same thing as the program described by P. Henry at the SF Usenix meeting (I could't wait to get his version). It switches one back and forth between a compiler and the visual editor, putting the error diagnostics from the compiler into the source file as comments. It will be on the UBC distribution tape.

2 We have a file "/etc/log" on our system, whose contents are (approximately)

```
(echo " "; date) > /etc/boot
echo "enter reason for re-boot, then ^D (control D)"
echo "please note your UNIX id at the end of the message"
cat /dev/tty0 >> /etc/boot &
set a = $P
(sleep 60; kill $a) &
wait $a
cat /etc/boot >> /etc/bootlog
```

This has the effect of logging the reasons for all boots of the system into the file /etc/bootlog and the latest reason into /etc/boot. There is a line in /etc/rc that will invoke /etc/log on each boot.

The non-obvious process manipulation in the above file is to bring the system up if there is no reason for the boot entered within 60 seconds. This is just in case the console isn't working.

3 I have changed DF so that it reads the default disks from the file /etc/disks. This makes it much easier for me to have one set of sources for a number of machines with different disks. Our DF also recognizes a -m switch which means "do a DF on all mounted disks". Our /etc/disks file normally contains

/dev/hm0 (root)
-m

Which causes a DF on all our file systems. The "(root)" is a comment that is printed for the user's convenience.

4 I have re-written "mkdir" in C, fixing various bugs and inconveniences at the same time. It also recognizes the "sticky" bit as meaning "Other" access is zero for making directories. Useful for preventing users from creating sub-directories in /tmp, /usr/tmp, etc. while still letting them create ordinary files.

5 I changed the link system call to allow up to 255 links in a file. The change in link is simple, but if you do it don't forget to change the code in iput so that the file is unlinked only when the link count is zero.

6 We have a useful feature in our text processing language (not related to nroff) that might be useful in nroff or other processors: in addition to allowing access to the current date and time, it can also get the modification date of the current input file. This allows one to generate documentation or letters (such as this one) whose date is the last time the source was modified.

7 I have a program called "mtime" that behaves like -mtime in find, and is useful in "if" commands. We use it to limit the file system purge (of old core files etc.) to once a day, regardless of the number of boots. The file looks like:

```
cd /usr/lib
if { mtime -1 flag } exit
find /usr/preserve -name "Ex*" -a -mtime +7 -a -print ^ long rm -f
find /mnt -name "core" -a -mtime +21 -a -print ^ long rm -f
find /u -name "core" -a -mtime +21 -a -print ^ long rm -f
touch flag
```

HELP WANTED

Does anyone out there have an interactive statistics and data plotting package running under UNIX that doesn't cost a lot of money? We need one.

Sincerely,

W. E. Webb.
Systems Analyst

WEW:fmt

Henry's Gossip Hotline

or

Highlights of the San Francisco USENIX Conference

Henry Spencer
29 Jan 1981

Some people have asked for an explanation of the rating system I use for items in the Gossip Hotline. Here it is. "Rumor" means I think it might be true. "Solid" means I got it from a reliable source and would be surprised if it were false. "Fact" means I got it from a source I consider entirely trustworthy, and would be shocked if it were false. If you are looking for absolute certainty, look elsewhere.

[Fact] There is a new commercially-oriented Unix-users' group, named '/usr/group'. So far most of their activities are directed towards getting answers from Western Electric on fine points of commercial licensing, although they plan a newsletter and other things.

[Rumor] Texas Instruments is starting to think it would be a good idea to have Unix available on their minis; they are informally sniffing around for someone who might be interested in doing it.

[Fact] Onyx is getting out of the software business: they have made a deal with Interactive Systems to handle the software end of their machines.

[Fact] BBN's C/70 ("C machine") exists; I saw one and played with it briefly. Seemed OK. I have detailed notes on the architecture etc. if anyone is interested.

[Rumor] A Unix version of TEX is being worked on; no clear indication of who, or whether it will run on 11's or only on VAXes.

[Solid] 3COM will soon be selling Ethernet transceivers ($360 in quantity) and PDP11 Ethernet interfaces ($8k with software). Intel is buying transceivers from them.

[Solid] Licensing the Ethernet patents from Xerox is trivial: a $1000 flat fee.

[Fact] The earlier rumors that all C compilers were being made interchangeable for licensing purposes were slightly wrong. Western Electric has now officially stated that if you are licensed for more than one kind of Unix, you may standardize internally on one C compiler – but you must have come by that compiler legitimately (i.e. no using the V7 compiler if you are only licensed for V6). A letter to this effect was supposedly mailed to all affected licensees in fall.

[Solid] WE is routinely investigating Unix lookalikes for signs of software theft, and is also alert to unlicensed people advertising for Unix hacks.

[Fact] WE has officially stated that Mark Williams Co's "Coherent" system was not derived from Bell code and is therefore legit. (Rumor: they had Dennis Ritchie himself spend a week looking at it.)

[Fact] As of mid-January, there are 900 Unix licensees with 2000 installations, not counting Bell itself. About 60% of these are educational licenses.

[Solid] Transfers of software from source-licensed people to binary-licensed people are legally hazardous, even if the software is stuff the binary-licensed people legitimately have.

[Solid] There is little interest at Murray Hill in further straight-line development of Unix, although miscellaneous minor enhancements are a different matter.

[Fact] WE customer auditing (for improper use of licensed software) is on the rise. Beware.

[Fact] There are several new software packages winding their way through WE licensing, probably including newer Bell-internal versions of Unix.

[Quote] "You buy Unix systems in spite of our sales force, not because of them." – Bill Munson of Dec.

[Note] A recent Dec blurb on some new and wonderful piece of hardware gives benchmarks for performance improvement of Unix but does not even mention RSX.

[Fact] Berkeley now has contracts from ARPA for Unix support work, including further performance improvement and other things. All this is being done on VAXes, Berkeley has essentially given up on 16-bit machines because it's too hard to make things fit.

[Fact] UCLA's distributed Unix is starting to function. Much work remains.

[Quote] "Nothing is easy on RSX." – Dan Strick at U of Pittsburgh.

[Fact] U of Saskatchewan has managed to stamp out keypunches completely by using a large Unix with RJE to their 370. They have done substantial work on making it student-proof; in particular, all students ever see is a specialized variant of the editor. The system handled 1000 students this fall. The software will be available to universities on an as-is basis.

[Quote] "Northern Telecom is the third biggest telephone-company supplier, i.e. the Chrysler of teleo suppliers." – David Tilbrook of DNR.

[Solid] A few working Perq's have been seen, with no software at all. People have been cancelling orders due to lack of deliveries.

[Fact] The 4th Berkeley Software Distribution ("4BSD") for VAX Unix contains modifications to the Unix filesystem code to make it largely crashproof, plus better repair software. The combination eliminates human filesystem fixing.

[Fact] The new small VAX (the 750) has been seen, tried, tested etc. It has a good Unibus, although no Massbus yet. Dec's speed assessments are reasonable. Software changes from the 780 are trivial. The big nuisance is that you get it only as a package, with RK07's and a TS11 magtape; both these peripherals are pretty awful. In particular, the RK07 is unusable in the presence of any other DMA activity. Berkeley suggests buying the package and selling the peripherals.

[Solid] For a VAX 780, it is possible to convince Dec that Dec disks are not needed for maintenance (reasonable since all the diagnostics run off the console floppy). This may be harder for early 750's.

[Solid] The purported necessity of having a Massbus on a VAX is bullshit, according to Berkeley. They say that for most configurations and most loads, Massbuses will do nothing for you that Unibuses won't do just as well.

[Solid] Fujitsu's Winchester disk drives get rave reviews from everyone. The current ones are 160-meg, with a 480-meg coming. Fast, reliable, *very* cheap. Two Fujitsu 160's are cheaper and much better than an industry-standard 300. But don't buy your *cables* from them.

[Solid] The only good Dec tape drives are the TU16 and the TU77.

[Fact] Emulex controllers get high marks from Berkeley.

[Solid] Emulex has a new DH lookalike coming, a single interface that looks like *four* DH's (64 lines total). This may be a viable solution to a problem Berkeley complains about: multiple DH's have serious problems with cable crowding.

[Solid] The new Versatec V80 has production problems but looks very good.

[Rumor] Canon has a $12K laser printer under test. It will be widely publicized on arrival; this is not expected soon. It is rumored to need good ventilation because of fumes.

[Solid] A 6250-bpi 125-ips tape drive pretty much wants a VAX Unibus adapter to itself; there will be trouble if other DMA devices are also present.

[Fact] "Number of users" is less valid as a performance measure on a VAX than on an 11, because the VAX lacks the 11's 16-bit limitations on how much a single user can load the machine: one superheavy VAX user can want the whole machine.

[Fact] An outfit called "Bedford Computer" has a box to extend the hardware address space on 11's from 18 to 22 bits.

[Solid] Interactive Systems now has a very *untroff*ish text formatter that is both much better and much faster. I have some details if anyone is interested.

[Fact] The next releases of Berkeley's *Ingres* database system will be in the public domain; they have given up on licensing hassles.

[Fact] The new Berkeley VAX Pascal compiler is very good. Doesn't fit on 11's, this may be fixed eventually.

[Solid] The VAX is increasingly the major machine of the big software-generating Unix users. They see 11's as increasingly not worth bothering with.

[Fact] There are now at least two commercial outfits that will take your *troff* output and typeset it for you. I have prices from one; they start at $2 per page.

[Fact] This was easily the biggest Usenix meeting ever; attendance circa 1000.

Converting V6 Drivers to V7

Henry Spencer
12 Feb 1981

This document attempts to sketch out a rough idea of what things need attention in converting a V6 Unix device driver to run under V7. This is based on limited experience, so errors and omissions may be expected. Comments on conversion of character-device drivers are particularly weak, because most of my experience so far has been with block devices.

It is assumed that the reader is pretty familiar with device drivers; this is not a primer on the subject.

Be ye aware that the version of *The UNIX I/O System* in the V7 manual is out-of-date and thus lies now and then.

- Device numbers are no longer manipulated with the kludgey structure arrangement used before; there are now one-parameter macros *major* and *minor* that yield the pieces of a device number. There is also a *makedev* macro which takes a major and a minor number and yields a device number.

- Kernel code, including drivers, now extensively uses predefined type names to enhance readability and portability. The following are notable. Type *physadr* is sometimes used for physical addresses; it is a pointer to a *struct* containing only one member: an array *r* of words. More usually, a physical address is a device address; these are usually declared as pointers to a structure *device* specific to that device. Type *daddr_t* is block numbers on block-type devices. Type *caddr_t* is core addresses. Device numbers are type *dev_t*, manipulated by the macros mentioned above. And offsets within files are type *off_t*. There are a few other predefined types not usually relevant to drivers; consult *types.h* or *param.h* for details.

- Be careful about doing arithmetic on predefined types, especially if for some foolish reason you are trying to write a portable driver. In particular, be wary of overflow.

- To use the kernel include file *user.h*, you must also include *dir.h* because one of the declarations in *user.h* depends on it.

- The driver-relevant items in the *u* structure have changed types; *u.u_base* is *caddr_t*, *u.u_count* is *unsigned int* (beware signed comparisons, assuming the count is under 32767, and trying to subtract from it and then check whether the result is <0!), *u.u_offset* is *off_t* (beware of things that use it as if it were still a two-integer array), and *u.u_segflg* now has *three* possible values (beware of anything that just tests it against zero).

- Disk drivers should beware of doing *int* arithmetic on *u.u_count* to determine whether a transfer is past the end of the disk section; adding *BSIZE-1* to an *unsigned int* just might cause overflow. Better to do it in *off_t* arithmetic by adding it to the value of *u.u_offset* first.

- The parameters to driver *close* routines have changed. There are three of them. First, as before, is the *dev_t* device number. Second is not just a read-write flag but the whole *f_flag* field from the *file* structure. Third is the *f_chan* field from the file structure, relevant only for multiplexed files. The interface to the *open* routine is unchanged, with the second parameter non-zero if the device is being opened for something that involves writing (the second parameter is in fact the *f_flags* field of the relevant *file* structure, anded with *F_WRITE*).

- The interface to the "special-functions" routine has changed totally, to provide for the new *ioctl* user interface. These days the routine, which should be named *xxxioctl* rather than *xxxsgtty* to minimize confusion and disaster potential, takes four parameters. The first is a *dev_t* giving the full device number, the second is an *int* giving the *ioctl* code, the third is a *caddr_t* giving a core address (in *user* space!), and the fourth is the flag field (a *char*) from the relevant *file* structure.

- The *ioctl* codes are different for different devices; by convention the top byte is a letter indicating what kind of device it's for, and the low byte is a number differentiating the individual *ioctl*s from each other. The special-functions routine should check the *ioctl* code it receives to determine whether the *ioctl* is one of the ones relevant to it; if not, it should give an *EINVAL* error and return.

- Passing information in and out of the special-functions routine is no longer so simple, since the address the routine receives is in *user* space. The routines *copyin* and *copyout* are relevant. The "in" and "out" are relative to system space. *Copyin* takes a user-space *caddr_t*, a system-space *caddr_t*, and a byte count (best obtained from *sizeof*). *Copyout* takes a system-space *caddr_t*, a user-space *caddr_t*, and a byte count. Both return 0 for success and non-zero for failure; the driver should generally give an *EFAULT* error in event of a failure here. The system-space *caddr_t*s are usually obtained by casting the address of a structure to *caddr_t*.

- Beware of drivers that assume they know the structure of the *clist*; there are now two possible lengths for a *clist* block, 8 and 16 bytes. The constant *CBSIZE* gives the number of characters in a *clist* block; the constant *CROUND* is a mask suitable for doing *clist* rounding.

- Calls to *clist* routines probably should pass the address as *caddr_t*, although existing V7 code doesn't bother (perhaps because the routines are in assembler; it *does* cast the same addresses to *caddr_t* when it passes them to other routines).

- Sleep priorities are now all positive and range from 0 to 127. The borderline between interruptible and uninterruptible priorities is *PZERO*.

- The address parameter to both *sleep* and *wakeup* is type *caddr_t*, as is the middle parameter to *timeout*.

- Over and above the usual spu-priority routines (*spl7* and the like), there is an *splx* routine which sets the entire PS (not just the priority!) to the value it gets as a parameter. All the spu-priority routines *except splx* return the value of the PS before the priority change. There is a mask *INTPRI* for the interrupt-priority bits of a PS, and a boolean macro *BASEPRI* which tells whether the PS given to it as a parameter has a non-zero interrupt-priority level.

- The size of a disk/tape block is now a defined constant; driver code should use *BSIZE* rather than wiring-in the number. *BMASK* is a mask for masking off the offset-within-a-block from an *off_t*; *BSHIFT* is a shift count for doing block-to-byte conversions and vice-versa without having to explicitly write the binary logarithm of *BSIZE*.

- The null pointer should be expressed as *NULL*, not 0.

- Care should be taken to use *NODEV* rather than -1.

- A number of fields within buffer headers (*struct buf*) have undergone minor changes. The most conspicuous is *b_wcount*, which has been replaced by *b_bcount*: a non-negated *unsigned int* (byte) count. Note that raw-i/o counts therefore now may call for an odd number of bytes, although *physio* still refuses to pass odd counts.

- *b_blkno* is now *daddr_t*; beware unduly-short arithmetic.

- *b_resid* is now, like *b_bcount*, a non-negated (byte) count, declared as *unsigned int*.

- All drivers (yes, even disk drivers!) must now return a proper value in *b_resid*. Even if you just set it to 0, make sure it gets s t.

- *b_addr* is no longer accessible directly; it is one member of the union *b_un*, and so must be accessed as *bp->b_un.b_addr*. It's now *caddr_t*, too.

- *b_xmem* is now *char*; this is no big thing because the largest value in it is only 6 bits, but beware of pointers to it.

- To get an empty buffer, e.g. for a work area for special drivers, use *getcblk*, not *getblk*. Feeding *NODEV* to *getblk* may crash the system!

- Be careful to check *B_PHYS* and request the map if needed; with the increasing proliferation of Unibus DMA on 22-bit systems, it's important.

- *B_AGE* has been added, to indicate that the person releasing the buffer judges that the same block will not soon be used again. A performance heuristic.

- *B_TAPE* has been added, see later.

- The *devtab* structure is no more. The header for blocks on the queue for a device is now a *buf*. Synonyms for some of the fields in *buf* provide roughly the same interface; in particular, *b_actf*, *b_actl*, *b_active*, and *b_errcnt* exist. Note that the names begin with 'b', not 'd'. *b_forw* and *b_back* exist as usual.

- There is an (undocumented) utility routine *disksort* which will sort a buffer (second parameter) into a queue (first parameter) on the basis of the cylinder number (assumed to be in *b_resid*).

- V7 drivers are careful to maintain the *b_actl* field in the queue header. This doesn't mean, however, that the device queues have become a doubly-linked circular list. The *b_actl* pointers in the blocks actually on the queue are not maintained, and the *b_actf* pointer in the last block is still *NULL*. *Disksort* does all this for you.

- *b_active* and *b_errcnt* are both *unsigned int* in the new order of things, so there is more room there than there was.

- The *b_flags* field in the queue header for a magtape-like device should have *B_TAPE* set; this tells *bdwrite* that writes must be sequential. This replaces the old shoddy system of having *bdwrite* know the names of the queue headers of magtape-like devices.

- *deverror* now takes two parameters to print out in octal, rather than just one.

- Disk drivers should use *daddr_t* fields for the size of filesystems. Beware initializing such fields with numbers computed by *int* arithmetic; if the number is between 32767 and 65535, the value in the field won't be right.

- Disk drivers that need to know whether a given i/o is a swap or not should be aware that there are now *two* swap-i/o buffer headers, *swbuf1* and *swbuf2*.

- There are some monitoring features in the Bell disk drivers that I don't understand yet and therefore won't discuss in detail.

- Terminal drivers now interact with the *tty* driver indirectly, via the *linesw* structure, for the most part. The intention of this is so that different kinds of tty handling can be plugged in simply by issuing the proper *ioctl* to change the "line discipline" on a given line. This stuff is very poorly documented and I do not yet understand it very well. The same goes, in spades, for multiplexed files and the hooks they have within drivers. Note in particular that the Bell DH driver has all this stuff more-or-less right but the Bell DZ driver doesn't.

- I may issue an updated version of this document when these last few items are clearer in my mind.

- Good luck.

Putting Unix V7 up on the 11/44

Henry Spencer
12 Feb 1981

Putting Unixes up on the new 11/44 is actually very easy. A V6 configured for a non-ID machine goes up instantly; I am told the same is true for an ID version, although I have not personally tried this. V7, however, does not; this document discusses what needs to be done to make V7 come up on the 44 ("Unix" henceforth means V7). Much of this discussion would also be relevant to bringing V7 up on an 11/70 with Unibus disks.

To software, the 44 actually looks very much like a 70. The resemblance is even stronger for Unix because Unix does not use most of the 11/70 features that are missing or different on the 44. There actually are only three significant differences between the two machines that Unix cares about.

The first, and least significant, is that where an 11/70 parks a Memory System Control Register, the 44 has a Cache Control Register. Unix actually makes very little use of this register; all it does is set it during the boot and startup. The value it is set to is 05; this is not quite right for the 44 because on the 44 the 02 bit is unimplemented. The 01 bit does roughly the same thing it does on the 70. So you may wish to change the value to '1' from '3' in *mch.s* and *M.s* (*M.s* is part of the bootstrap code, located in */usr/src/cmd/standalone*); instructions that set *MSCR* are the ones to change. This doesn't make any practical difference, however, and I haven't bothered with it yet on my 44.

A slightly more significant difference is that the 44's cache registers are not the same as the 70's memory-control registers. The code in *trap.c* that handles trap type 10 attempts to print the values of some of these registers, not all of which exist on the 44. This matters only if you actually get such a trap, which should be uncommon in any event, and is actually impossible in the distributed V7 because trap vector 114 (incorrectly) gives trap type 7 rather than 10. I haven't bothered with this yet either.

The big difference is that the 44 has no Massbus. This means that all DMA goes via the Unibus, and (flourish of trumpets) 22-bit memory addresses must be generated by the Unibus map. The boot turns on the Unibus map but does not initialize it properly. Since the boot has to be done via Unibus DMA on the 44, V7 won't boot on the 44. (V6 works fine because V6 uses a different, cruder boot program.)

There are three levels of fix for this: quick, reasonable, and best. The quick fix can be done without changing the software (i.e. it can be used to boot the distributed V7 tape); the others take software work.

If you have a 44 with one of the disks supported by the standard V7 distribution tape, and have no other Unix handy, you will need to use the quick fix to bring V7 up. Go through the boot sequence until the point where the console has just typed "Boot" and ":". Then get into console mode on the console terminal (control-P), halt the CPU, and examine memory-management control register #3, which is at 17772516. The value in it will be 065. Change it to 05. Continue the CPU and proceed with the boot. You will probably have to modify the register after every occurrence of "Boot". Unix should come up.

What the quick fix does is to turn off both 22-bit addresses and the Unibus map, making the machine look like a 45. Unix itself looks at this register and believes the setting provided by the boot, so your Unix will think it is on a 45. This restricts you to using only the first 18 bits worth of memory, but this is plenty to get you up and running long enough to apply one of the better fixes. This is what I did (although I did not try it from scratch from the distribution tape because I do not have Dec-compatible disks).

The reasonable fix for this problem is to make the Unix think it is on a 70 with Unibus disks, and act accordingly. This requires two things.

First, the relevant disk drivers *must* request use of the bus map when appropriate. Note that the Bell RP driver, for one, does not. The way to do this is to put, at the start of the *strategy* routine, the code:

```
if (bp->b_flags&B_PHYS)
        mapalloc(bp);
```

This will work fine even on a smaller 11, because the *mapalloc* routine returns immediately if it is not relevant to the particular cpu; all drivers for Unibus disks and tapes should have this code.

Second, the *M.s* code in the bootstrap should initialize the Unibus map properly. Note that the Unibus-map initialization in Unix proper is not right for the bootstrap. Edit *M.s*. Look for the instruction that initializes *MSCR* (you may want to change the initialization value as mentioned earlier). The branch that immediately precedes it is "branch on 18-bit machine". (I'm sorry I can't show the code, but it's Bell code and there's this annoying nondisclosure requirement...). Immediately after the instruction that sets *MSCR*, add the following:

```
        mov     $UBMAP,r0    / initialize Unibus map...
        clr     r1
        clr     r2
        mov     $31.,r3
2:
        mov     r1,(r0)+
        mov     r2,(r0)+
        add     $20000,r1
        adc     r2
        sob     r3,2b
```


and put the definition of *UBMAP* down under the definition of *MSCR* near the end of the file:

```
UBMAP       = 017770200  / 11/70 and 11/44 Unibus map
```

Rebuild the bootstrap, and off you go.

The one possible trouble you might have is that drivers for RH-type controllers (RP04/5/6, TU16/TE16) have to know whether the RH is an RH11 (Unibus) or RH70 (Massbus), and they foolishly decide this by simply checking *cputype* to find out whether they're on a 70. They also assume that they never need to do a *mapalloc*, since RH70's don't use the Unibus and we all know nobody runs RH11's on a 22-bit machine... All this is not really right even for a 70 and is dead wrong for a 44, so you will have to change such drivers. Try putting in the *mapalloc* as described above and commenting out the 70-only code (look for the use of *cputype*); I can't say for sure whether this works because I am fortunate enough not to have RH's.

The best fix for the Unibus-map issue is to have Unix know it's on a 44 and act accordingly. The actual setting of *cputype* is easy, just look at *MSCR* after setting it to 3 and find out whether it really *is* 3; if it's 1 instead, you're on a 44. Or you could try the MFPT instruction, but remember it will trap on all but the most recent 11's. The trouble is that this requires running down all the places where *cputype* is checked and deciding what the code should do if the value happens to be '44'. Clearly, the code should really not check *cputype* at all; it should ask some central module "does this particular CPU have feature X?". I haven't done this yet but I intend to; when I do, I'll publish it.

For anyone who wants to try in the meantime, here is a list of places where *cputype* is examined, and why, in the distributed V7:

ht.c,hp.c	To distinguish between RH11 and RH70 controllers (arghl).
machdep.c	To know whether to initialize the Unibus map, and to determine whether *mapalloc* is a no-op.
trap.c	To know what to do about trap type 10 (memory error).
ureg.c	To know whether ID-space is available, for purposes of setting segmentation registers and determining whether ID space a.out's are legal.

Late flash! There is another problem with V7 on the 44: Dec has changed one detail of the integer divide instruction. On all 8 previous models of the PDP-11, when div aborted because the quotient would not fit in a 16-bit signed number, the register-pair forming the dividend was unchanged. This is not true on the 44. This affects the long-int divide and remainder routines, both the C-library ones and the kernel ones in *mch.s*.

Specifically, in *all* of these routines, there is a div immediately followed by a bvc, and the routine assumes that if the bvc fell through (i.e. the div aborted), that r0 and r1 are unchanged. It is necessary, if the bvc falls through, to restore r0 and r1 to their previous values.

Restoring r1 is dead easy, since it's just a copy of r2 at this moment; just add "mov r2,r1" after the bvc. Restoring r0 is harder. In the remainder routines, just add "mov r0,(sp)" before the div, and "mov (sp),r0" after the bvc. In the divide routines, alas, the stack top is in use; add a "mov r0,-(sp)" before the div, a "mov (sp)+,r0" after the bvc, and change the target of the bvc from simply the label "1:" to:

```
        br      2f
1:
        tst     (sp)+           / remove unneeded saved r0
2:
```

These fixes have not yet been exhaustively tested.

UNIVERSITY OF DUBLIN
TRINITY COLLEGE
SCHOOL OF MATHEMATICS

Telephone 772941
Extension: 1949

39 TRINITY COLLEGE
DUBLIN 2
IRELAND

ref: IUUG/8

Dr. Mike Tilson,
Canadian UNIX Sig Newsletter Editor,
Human Computing Resources Corp.,
40 St. Mary Street,
Toronto, Ontario,
M4Y 4P9,
U.S.A.

12th February, 1981.

Dear Editor,

Could I become a subscriber to the Canadian UNIX Newsletter, of which I believe you are editor.

I am the Secretary of the recently formed Irish Unix Users Group; and it was suggested at our inaugural meeting that I should set up a small library of UNIX software and Newsletters. Unfortunately our request to the government for funding for this was rejected, so we have to meet the expenses more or less out of our own pockets. (I mention that in case you have a very expensive category of membership for other UNIX froups!)

Looking forward to hearing from you.

Yours sincerely,

Timothy Murphy

Timothy Murphy.

;login:
THE UNIX NEWSLETTER

Volume 5 Number 6 August 1980

Contents

* UNIX is a trademark of Bell Laboratories

;login:
THE UNIX NEWSLETTER

Volume 5 Number 7 September 1980

CONTENTS

Notice

* UNIX is a trademark of Bell Laboratories

;login:

THE UNIX NEWSLETTER

Volume 5 Number 8 October 1980

CONTENTS

Notice

* UNIX is a trademark of Bell Laboratories

Buffer Deadlock in UNIX

Buffer Deadlock in UNIX

Darwyn Peachey

Hospital Systems Study Group
3337 8th Street East
Saskatoon, Canada
S7L 4J1

As one of the few installations running 7th Edition UNIX on a "small" PDP-11, we have the dubious honor of having one of the lowest numbers of available disk buffers in UNIX history. With a typical number of file systems mounted, we have only 5 or 6 buffers available for I/O. UNIX functions fairly well even under these conditions, and we can support several people doing program development and text editing. However, because of the scarcity of disk buffers in our system, we have been the victims of an interesting deadlock situation.

The reason for the deadlock is a bug or near-bug in "bio.c" in both 6th Edition and 7th Edition UNIX. When a process enters "breada", one buffer is obtained and used to start the necessary I/O operation. Then a second buffer is requested to do a readahead I/O. With few buffers in the system, the process often must go to sleep until a free buffer is available. Unfortunately, the buffer used for the first I/O will never be made available for the second I/O, because the process is sleeping on the address of the free list header, and the disk driver does a wakeup on the buffer address (NOT the free list address) when the first I/O is completed. The buffer is marked BUSY and is not available for use by anyone until the process which grabbed it gets another buffer and gets out of "breada". In a system with very few buffers and several users it is quite possible for 5 or 6 processes to all enter "breada" and grab buffers at roughly the same time, and then go to sleep waiting for more buffers that will never be available. Every active process in the system very quickly reaches a point where a buffer is needed, and goes to sleep. Nothing can be done except to reboot the system.

I know of other places in UNIX where buffer deadlocks can occur (for example, in "bmap" (file "subr.c") when adding to a large file) but the bug in "breada" seems to be the only one with a high probability of happening. Luckily, this deadlock is easily prevented. My fix consists of changing the "getblk" routine in "bio.c" so that it has another parameter, a flag which is nonzero only in the second "getblk" call in "breada". When the flag is zero, "getblk" behaves exactly as it always has. When the flag is nonzero, no waiting is allowed in "getblk" -- if no buffer is available, a NULL pointer is returned. This allows "breada" to skip the readahead I/O if all buffers are busy. Measurements on our system show that over 99% of the readaheads still get done -- readaheads are only omitted when things are very bad in the buffer pool, so bad that the readahead I/O would probably be of no benefit anyway.

;login:
THE UNIX NEWSLETTER

Volume 5 Number 9 November 1980

CONTENTS

* UNIX is a trademark of Bell Laboratories

Tiny C*

Tiny-C Two - The Compiler

Tiny-c two is ten times faster than tiny-c one. It has many extra features, including long (32 bit) integers, lots of new operators, and redirectable and direct access input/output. This version of tiny-c is viable for professional work, either systems programming or business applications.

It comes with a UNIX* style command interpreter called the "tiny-shell"*. With the tiny-shell, every compiled tiny-c program becomes a new shell command. Tiny-shell commands can have arguments, and dash (-) options, just as real UNIX shell commands do. The <and> input/output redirection operators are supported.

There are over fifty standard library functions, and this set is readily extended. The input/output functions are UNIX style, including fopen, fprintf, etc. Both ascii and raw (binary) input/output are supported.

And the entire package is portable. Bringing it up on a new processor or new operating system should take a few days or a few weeks at the most. And as usual with tiny-c products, all the source code is included.

Language Features

- All the features of tiny-c one
- Additional operators: not, complement, address of, postfix and prefix increment and decrement, left and right shift, and, or, exclusive
- UNIX style i/o; redirectable by the tiny-shell or by program, ascii and raw (binary), formatted print and scan, direct access (lseek)
- Program chaining for very large applications
- Dynamic storage allocation (calloc, cfree)
- Improved machine language interfaces

Physical Features

- 32K recommended. This is enough to compile the compiler.
- The compiler is written in tiny-c; all source code is included
- Emits a very compact, stack oriented intermediate code
- Interpreter for the intermediate code uses about 2K bytes
- Standard assembly language portion of the library uses about another 2K bytes. (The tiny-c coded portion of the library is loaded as needed).
- PORTABLE - readily transported to other processors or operating systems. The bootstrap procedure is well documented, and tests are

* UNIX is a trademark of Bell Laboratories, Inc.
Tiny-c and tiny-shell are trademarks of tiny c associates

provided.
- Speed: 500 to 1000 statements per second on typical 2 MHz to 4MHz 8 bit processors

Human Features

- Thorough documentation: over 200 pages. This includes a tutorial walk-through, a reference chapter, a reference with examples on the tiny-shell, lots of sample programs (and they are useful ones), internals describing how the compiler and linker interface to the tiny-shell, and all the details on how to install this system on any computer
- The tiny-shell support multiple commands per line, input/output redirection, and has thorough error control. Most commands have UNIX style dash (-) options.

For more information contact:

tiny-c Associates
Post Office Box 269
Holmdel, New Jersey 07733
(201) 671-2296

From dave Fri Mar 27 09:42:45 1981 netmail from unswcsu
Subject: SCCS revisited

I have since found the problem with SCCS. It turned out that the original STDIO "sprintf" returned the string, whereas the #7 one returned the number of characters !!! This screwed up SCCS good & proper. Once again, so much for standard libraries. It would be appreciated if you would add this after my article.

From peteri Wed Mar 18 15:09:23 1981 netmail from elecvax
To: auugn:elec70 kev lindsay:agsm
Subject: the mail below

> From dave Wed Mar 18 14:26:04 1981 netmail from unswcsu
> Subject: auugn
>
> I just mailed a contribution to AUUGN (to auugn:elec70) about my SCCS demo. It is basically what would have appeared on the big screen, had everything been working.
>
> Could you also include a plea somewhere for some decent mag-tape handling software ? Something (say a C library) to read/write ANSI labeled tapes would be a great winner in the argument of UNIX vs. DEC software. I hate to think how many installations turn down UNIX because it cannot handle a simple labeled magtape ! Sure - software can be written to do it but this is just re-inventing the bloody wheel over and over again !!!
>
> I propose routines called something like TMOUNT, TUMOUNT, TREAD, TWRITE etc & corresponding utilities TMOUNT & TUMOUNT. Or perhaps the mag tape dep. stuff could be done inside the kernel instead ?
>
> Anyway - if anyone has written something along these lines would they please tell everyone about it ? If not, I guess I'll have to do it myself so contributory comments would be in order.

From peteri Wed Mar 25 12:53:00 1981 netmail from elecvax
To: auugn:elec70
Subject: the things people do

> From root Sat Mar 21 13:49:19 1981 netmail from elec70
>
> From dave Tue Mar 17 12:47:34 1981
> Subject: connect
>
> I am trying to figure out a way to copy to a file whatever is displayed on a terminal. I find that TTYVIEW doesn't work because it expects a TTY file descriptor, and anyway it can only redirect to a terminal at interrupt time.
>
> I don't want to have to go to the bother of writing a filter program sitting between the shell and me if I can help it because it raises all sorts of problems (need 2 filters, need synchronizing between them etc etc).
>
> Basically I want to create a log of everything I do on a terminal, which in this case is that abortive SCCS demo so I can include it for publication in AUUGN, and I don't feel like fudging a 15 minute

session !

What is really needed is a CONNECT function between 2 arbitrary file descriptors. Any ideas ?

From dave Wed Mar 18 14:06:31 1981
Subject: connect etc

You remember my previous mail about trying to use TTYVIEW to get TTY input/output copied to a file ? Well - I finally did it !

On the MASTER terminal, I view a SLAVE terminal. The VIEW program was my own - it just does a ttyview without exec'ing a shell. I also had to turn off echoing, otherwise the system went berserk echoing & reechoing the first key I typed (the reason becomes obvious in a minute). I also had to disable newline translation, because for some odd reason all newlines tripled ! Then I physically disconnect the MASTER, insert a loop-back plug in its place, set up a process copying MASTER to the file I want, then merrily type away on the SLAVE. The file needs to be edited afterwards to remove extraneous <CR> at the start of each line, remove echoed passwords etc etc.

For the trouble I went to, it was worth it !

Pete: for your amusement - AUUGN material??

kev

From peteri Fri Apr 3 09:10:26 1981 netmail from elecvax
To: auugn:elec70
Subject: whats on at melbourne U

>From kre Tue Mar 31 16:12:51 1981 netmail from basser40
Subject: Adrian Freed's request

What I (we) am (are) doing ...

that just about sums it up!

(in my spare moments, I am looking at putting a rational AUSAM onto 4bsd - will be much the same internally probably, not even similar to the outside world. Also looking at implementing file locking properly (ie: no absurd only one locked file per process garbage). Naturally also trying to make P.E. unix more up to date (ie: progressing beyond v7)

Major initiative is trying to pressure money people to give us some, & let us decide how to spend it)

Of course - I am always looking at new things to do to the tty driver !!!

Regards, Robert

From davidr Tue Mar 3 20:12:15 1981 forwarded by root
hey kev, just have a peep in my directory at a file called

"antarctica" nice to get your letter...
davidr...
(at mawson!!!!)

From peteri Wed Mar 4 09:08:06 1981
The following is the letter from David Robinson, logged in from Antarctica. The letter has not been edited so you can see the difficulties he had to put up with.

pete

greetings from the antarctic!!
this message has been sent over the first know}n unix link from this frozen continent, and
has been made possible by ham radio stations vk2buv and vk0sj, and unsw elec eng unix site.
the remote site is at mawson((australia's premier antarctic base) where i have spent the last year with 30 other expeditioners. we have all enjoyed an excellent year here, the antarctic has a rare beauty.
some time in the next week the last of the 80 party will be leaving here and returning to australia (which we are all looking forward to)
this contact has involved both radio stations, and several other people at unsw in a good deal of work, for which i thank them. i the contact has been a highlight of the year, and i look forward to the time when unix has0a home down here
. david robinson (ipso mawson 1980)

From piers Mon Apr 6 19:04:33 1981 netmail from basservax
To: peteri:elecvax
Subject: a summary of UNIX in Australia to be presented at European U Meet.

This is probably too late! (but here it is anyway).

At Basser we are not doing much development due to the pressure of student support projects, but we are still interested in developing the SUN network. Mainly enhancements are planned for the near future, (rather than any particularly revolutionary work), such as upgrades to the functionality of the virtual host-host links, and error correction.
Some effort is being applied to Micro research, especially support for 68000 C-Compilers etc. There will be some development of an in-house optical fibre link at 2Mbaud using a Cambridge-ring like architecture.
We are considering putting AUSAM into the Berkeley Unix/32V system.

Otherwise, not much!

Piers.

From root Mon Apr 6 10:10:21 1981 netmail from basservax
To: peteri:elecvax auugn:elec70
Subject: a quote for the next AUUGN

"Being written in a high-level language,
Unix doesn't have bugs, so there are no
updates"

From: "The Unix Operating System"
by Eric Foxley
in Computer Age, December 1980.

UNIVERSITY OF DUBLIN

TRINITY COLLEGE

SCHOOL OF MATHEMATICS

Telephone 772941
Extension: 1949

39 TRINITY COLLEGE
DUBLIN 2
IRELAND

ref: IUUG/5

Dr. Peter Ivanov,
Computer Science,
Electrical Engineering,
University of New South Wales,
P.O. Box 1,
Kensington 2033,
Australia.

12th February, 1981.

Dear Dr. Ivanov,

I should like to subscribe to the Australian UNIX Newsletter, of which I believe you are editor.

I am Secretary of the recently formed Irish Unix Users Group; and it was suggested at our inaugural meeting that I should set up a small library of UNIX software and Newsletters. Unfortunately our request to the government for funding for this was rejected, so we have to meet the expenses more or less out of our own pockets. (I mention that in case you have a very expensive category of membership for other UNIX groups!)

We would be very grateful for any back numbers of your Newsletter that are still available; and for any distribution tapes the Australian UNIX group might have produced. Of course, we would be more than willing to send blank tapes, pay the return postage, etc.

Looking forward to hearing from you.

Yours sincerely,

Timothy Murphy

Timothy Murphy.

P.S. I should confess that there are only four centres running UNIX in Ireland at present; but there seems to be very great interest in the system - IUUG has over thirty members, and some fifty people turned up to hear a recent talk on UNIX.

THE UNIVERSITY OF NEW SOUTH WALES

P.O. BOX 1 • KENSINGTON • NEW SOUTH WALES • AUSTRALIA • 2033
TELEX AA26054 • TELEGRAPH: UNITECH, SYDNEY • TELEPHONE 663 0351
EXTN. 3781

SCHOOL OF ELECTRICAL ENGINEERING

PLEASE QUOTE

March 9, 1981

Timothy Murphy,
School of Mathematics,
University of Dublin,
39 Trinity College,
Dublin 2,
IRELAND.

Dear Timothy,

As always I am pleased to hear from another UNIX group, although I thought Ireland would be covered by the European UNIX group. Still I don't suppose that is reason enough not to have an Iuug.

I have enclosed invoices for all issues of AUUGN so far produced, and no I don't have a very expensive subscription rate for other user groups. In fact usually I exchange newsletters free with other groups, but as I already exchange with the Euug and as you don't seem to be ready to produce a newsletter as yet, you will have to pay.

As for setting up a small library of software, I think the words 'small' and 'software' are mutually exclusive. We have more than 50 magnetic tapes of software from overseas and are in the process of compiling a software catalogue. Possibly a copy of the catalogue when it is completed would fulfill your needs. We will send you the three UNSW distributions if you want them and I have enclosed invoices for these. We will also need to see a copy of any UNIX licenses you have.

Yours sincerely,

Peter Ivanov

Newsletter Editor,
Australian UNIX Users Group,
Dept. Computer Science,
University of N.S.W.
P.O. Box 1,
Kensington,
N.S.W. 2033,
AUSTRALIA.

THE UNIVERSITY OF NEW SOUTH WALES

P.O. BOX 1 • KENSINGTON • NEW SOUTH WALES • AUSTRALIA • 2033
TELEX AA26054 • TELEGRAPH: UNITECH, SYDNEY • TELEPHONE 663 0351
EXTN. 3781

PLEASE QUOTE
March 9, 1981

SCHOOL OF ELECTRICAL ENGINEERING
Alan Mason,
Dept. of Computer Engineering,
Heriot-Watt University,
Mountbatten Building,
31-35 Grassmarket,
Edinburgh EH1 2HT,
UNITED KINGDOM.

Dear Alan,

Just a brief note to let you know that I have received a letter from Timothy Murphy, School of Maths, Trinity College, University of Dublin. He says he is the secretary of the Irish Users Group and is setting up a library of UNIX information.

I have sent him subscription details and answered the other questions he asked. I cant help wondering if this is a splinter faction of the euug or is Ireland not counted as part of Europe. You might put him on the euug mailing list if he is not there already.

Yours sincerely,

Peter Ivanov

Newsletter Editor,
Australian UNIX Users Group,
Dept. Computer Science,
University of N.S.W.
P.O. Box 1,
Kensington,
N.S.W. 2033,
AUSTRALIA.

EUUG

EUROPEAN UNIX USER GROUP

COMMITTEE
Chairman : Alan Mason, Heriot-Watt University
Editor : Bruce Anderson, University of Essex
Member(s): Peter Collinson, University of Kent

R.A.Mason
Dept. Computer Engineering
Heriot-Watt University
Mountbatten Building
31-35 Grassmarket
Edinburgh EH1 2HT
(Tel. 031-225-8432 x 155)

17th March 1981

Peter Ivanov,
Newsletter Editor,
Australian UNIX Users Group,
Department of Computer Science,
University of N.S.W.,
P.O. Box 1,
Kensington, N.S.W. 2033,
AUSTRALIA.

Dear Peter,

Thanks for the note. No, Timothy is not forming a splinter fraction of the EUUG. He is in fact a member of a fully constituted 'Local UNIX* User Group' (LUUG) of which we have a number. These subgroups were set up so that geographically close sites could meet more regularly (monthly) than the European Group, and so that more formal (minuted) meetings could be held to satisfy certain administrative bodies within the community.

He should, in fact, have described himself as
'Secretary, Irish Local UNIX User Group'
or
'Secretary, IrLUUG'.

I've explained to him that we have both software and newsletter exchange agreements, and that although he may choose to individually subscribe, the User Groups would prefer to keep this to a minimum, lest our respective editors get overloaded with inter-continental mailings.

I've also pointed out that if he or another of his group were to become committee members of the European Group, then they would 'automatically' receive copies of Australian, Canadian and U.S. Newsletters for evaluation.

Yours,

R.A. Mason

P.S. I believe Timothy was asking about Modula 2 for UNIX V7, could we have a copy when it is ready for distribution?

* UNIX is a trademark of Bell Laboratories.

cc: Timothy Murphy, Secretary, IrLUUG,
School of Mathematics,
Trinity College,
Dublin, Ireland.

THE UNIVERSITY OF NEW SOUTH WALES

P.O. BOX 1 • KENSINGTON • NEW SOUTH WALES • AUSTRALIA • 2033
TELEX AA26054 • TELEGRAPH: UNITECH, SYDNEY • TELEPHONE 663 0351
EXTN. 3781
PLEASE QUOTE

April 13, 1981

SCHOOL OF ELECTRICAL ENGINEERING

Adrian Freed,
Group Informatique,
L.E.R.S. - Synthelabo,
58 Rue de la Glaciere,
Paris 75013,
FRANCE.

Adrian Mate,

I have held off as long as possible waiting for replies to the electronic mail requesting summaries of 'wots on in Aussie land'.

At UNSW this is whats on:

- AGSM have completed the upgrade of their disc system from two DEC RP04s to two CDC 300Mbyte drives on an EMULEX controller (on the cache bus). They are about to upgrade from level 6 to level 7 UNIX. Apart from that the AGSM is doing what they have always done.

- In Elec. Eng. we have upgraded the PDP11/70 with the two RP04 drives from the AGSM, thankfully we are now able to get rid of the AMPEX drives onto two smaller PDP11s where their error rates show a drastic reduction.

  We have the PDP11/70 running and AUSAMised level 7 system which is getting better every day. The VAX system and the 70 system are converging as far as source goes thus reducing our maintenance and development effort. Kev and I now control:

  - PDP11/70 with two RP04s, TE16, 640Kb, and about 50 terminals.
  - VAX11/780 with two RP06s, TU77, 2Mbytes, and about 65 terminals.
  - PDP11/34 (the Digital Systems Lab machine for micro hacking) with an AMPEX 100Mbyte drive, and about 25 terminals.
  - PDP11/40 (Computer Science Departmental machine) with about 20 terminals.

  There are also a PDP11/40, PDP11/34, LSI11/23 and sundry smaller LSIs running various flavours of UNIX within the building.

Dave Milway has the BFI (Bloody Fast Interface - serial bit basher) running between the PDP11/70 and the PDP11/40. He has the newest version well on its way using a 68000 as the on board control. The school has gone totally over to network mode, ie the school views all the machines in the building as one resource into which any user can connect. People need not restrict them selves to one machine.

- At the CSU they have placed orders for two VAXes to replace the CYBER 72, due for delivery in late 1981. Also some money has been spent on upgrading the CYBER 171 substantially. The CSU look like running VMS on the VAXes, but this is not final yet. They plan to offer UNIX as a subsystem under VMS.

- More schools and departments seem to be coming out of the wood-work buying small PDPs etc running UNIX, the latest being psychiatry at Prince Henry hospital.

The Sydney Net is blossoming, with more nodes coming in every day. Access is now available to it via CSIORNET around Australia (see the attached doco) and I regularly get mail etc from Perth, Melbourne etc. Below is a netstate type map of the net now.

```
elec70
        ->      dsl
        ->      elec40
        ->      unswcsu
        ->      agsm
        ->      elecvax
elecvax
        ->      basservax
        ->      elec70
agsm
        ->      elec70
        ->      unswcsu
        ->      basser40
basser40
        ->      sucyber
        ->      basservax
        ->      chemeng
        ->      csiro
        ->      agsm
unswcsu
        ->      comm40
        ->      maths
        ->      mech
        ->      comm
        ->      elec70
        ->      civil
        ->      agsm
elec40
        ->      elec70
        ->      unswpower
        ->      syscon
chemeng
        ->      sucyber
        ->      mhd
        ->      basser40
syscon
        ->      elec40
mhd
        ->      chemeng
csiro
        ->      basser40
basservax
        ->      basser40
        ->      elecvax
unswpower
        ->      elec40
sucyber
        ->      chemeng
        ->      basser40
civil
        ->      unswcsu
mech
        ->      unswcsu
comm
        ->      unswcsu
maths
        ->      unswcsu
comm40
        ->      unswcsu
        ->      comm34
comm34
        ->      comm40
dsl
        ->      elec70
```

I have received some replies to my request for info and these appear attached to this letter.

Robert Elz from Melbourne spent 3 weeks at Berkeley and gave us a summary of what is going on there at the last meeting. He said very little that I have not published in AUUGN already. Ian Johnstone told us about Bell, but swore he did not want to be quoted on it. Ross Gayler from Queensland told us for what the Psychology Dept uses UNIX. Mostly much document processing with many version 7 utilities converted. They also do significant micro work at Elec. Eng. up there, while Rick Stevenson seems to have made a VERY good job of squashing level7 onto the smaller PDPs. Paged kernel overlays and mapped buffers etc make level 7 viable on smaller machines.

There seems to be a trend towards high schools using UNIX as I have had several enquiries about the possibilities and the St Peters Lutheran

College in Queensland run it now.

Unfortunately the summary from Perth did not arrive in time to be included. I will send it as soon as it comes in the hope that I will catch you. Sorry I cant do more but I just don't have time for much these days.

Yours sincerely,

Peter Ivanov

Newsletter Editor,
Australian UNIX Users Group,
Dept. Computer Science,
University of N.S.W.
P.O. Box 1,
Kensington,
N.S.W. 2033,
AUSTRALIA.

The Flinders University of South Australia

BEDFORD PARK SOUTH AUSTRALIA 5042
TELEPHONE: 275 2198

Flinders Institute for Atmospheric and Marine Sciences

25th March, 1981.

Peter Ivanov,
School of Electrical Engineering,
University of N.S.W.,
P.O. Box 1,
KENSINGTON, N.S.W. 2033

Dear Sir,

Thank you for the information regarding UNIX enclosed in your letter of 18th March 1981.

I have one or two questions however, but first let me give you some more details of our proposed system.

Option 1 :
- LSI - 11/23 processor
- 128 Kb memory
- 20 Mb Winchester drive (PERTEC) emulating RL01 or RL02
- 1.25 Mb Floppy drive emulating RX02?
- 4 Serial parts
- 1 Parallel part
- 1 Floating point unit (KEF11)
- 1 Memory management unit.

Option 2 : as above with the following exceptions
- 256 Kb memory
- 7.5 Mb Winchester (DSD 880) emulating RL01

These systems have been offered for nearly identical prices, but option 1 looks most attractive to our requirements. This leads me to the first question - is 128 Kb going to be enough memory? We only anticipate supporting 2-3 users in the first instance.

The second question regards the disk drive. I have been verbally assured that the PERTEC drive emulates an RL01 exactly. I'm afraid that this is an area with which I am not very familiar, and therefore tend to be somewhat 'suspicious". Will UNIX run on such a system, and if not, what modifications would be necessary?

My last query concerns the version of UNIX which would meet our requirements. Several of the features described on page 1 of the UNIX/32V summary would be desirable, especially the Graphics utility. It would appear, however, from your letter to David Woodrow, that UNIX/V7 will only run on the large PDP machines. Can we get the Graphics utility with V6? What are the possibilities of the combined V6/V7 license?

Thank you once again for your cooperation and assistance.

Yours sincerely,

Resolved queries by phone
6/4/81 1600.

Trevor Norman.

CSIRO

DIVISION OF MATHEMATICS AND STATISTICS

P.O. BOX 218, LINDFIELD, NSW, AUSTRALIA 2070. TELEPHONE (02) 4676211. TELEX AA26296.

RIB:unix

March 29, 1981

Peter Ivanov,
School of Electrical Engineering,
University of New South Wales,
PO Box 1,
KENSINGTON NSW

Dear Peter,

Here are some observations on the use of CSIRONET as a UNIX peripheral. I have tried to cover some of the things mentioned at the recent UNIX meeting.

If we consider CSIRONET to be a packet switching network with PDP 11 computers at each node, we need only consider two types of gadget hanging off these nodes, viz. user-terminals and host-computers. User-terminals may connect to host-computers by using a login sequence such as *CYI (for cyber 76 interactive service), *MID (for MIDAS that bloody great host-computer in the sky), or *DIM (the basser40 UNIX). Some of these login sequences require details of CSIRONET accounts. On the other hand host-computers cannot initiate communications with a user-terminal but can communicate with other host-computers.

The *DIM link that connects the basser40 UNIX to CSIRONET as a host-computer depends on software developed by John Gibbons, CSIRO Division of Computing Research (DCR) Sydney, and at this stage it is still under development and will not be released for general use yet.

The three UNIX systems in our Division (DMS) are connected to CSIRONET but look like user-terminals. This allows our UNIX users to connect to a CSIRONET host-computer and transfer files. In particular, we can communicate with basser40 UNIX and transfer files. For this I have used **utalk** (from Piers Lauder) because it does not use checksums as happens with **log** or **con**.

This form of UNIX-UNIX communication has the following limits imposed on it by CSIRONET.

i. We must call basser40, not vice-versa.

ii. If an output record arrives at a CSIRONET terminal when an input record is part-typed, the input is lost and the output is printed. This makes attempted type-ahead frustrating.

iii. CSIRONET nodes inspect input characters from terminals for the <DLE> character (CONTROL-P) and use it together with one or two following characters to perform various commands - mainly setting terminal characteristics. This prevents the transmission of packed files and checksums.

However, files can get transferred and this form of connection may be tried by anyone with a UNIX that can dial out. For those who may try

here are some tips.

a. The login sequnece *DIM may be in upper or lower case but must be terminated by <LINEFEED> or if you like <RETURN>, <LINEFEED>.

b. Once you are connected your terminal behaves as follows. Typed characters are stored in the local node (and echoed to your terminal) until a control character (anything < 040) is typed - the stored characters are then sent as a "packet". Thus the usual UNIX conventions for <RETURN> or <LINEFEED>, for <CONTROL-D> and for backspace are OK. However, <DELETE> is not sent until <RETURN> is typed, but even this sequence is no good for stopping long listings because of limitation (ii) above. However, all is not lost because...

c. The sequence <DLE> A sends <DELETE> to the UNIX host and also aborts the transmission of any output currently in transit within CSIRONET. The sequence <DLE> T terminates the *DIM connection. There are many other <DLE> sequences but I haven't found them necessary.

The attached listing shows a short session using this link.

I think that all these limitations and problems disappear if both UNIX systems are host-computers on CSIRONET. Late last year I wrote to DCR enquiring about the possibility of using John Gibbons modifications to node software for connection of our UNIX systems. The reply is attached and since it arrived I have been told that DCR will definitely make this available.

At first glance it looks expensive, but for us at least it may not be so bad since we already have the DL-11s for our existing connections, and maybe we can find some PDP11 memory in our own PDP 11/10s that are nearing the scrap-heap.

For non-CSIRO UNIX systems the main hurdle will be to find a "friendly" owner of a CSIRONET node to assist you. John Gibbons seems to think that once a node is modified for connection of one UNIX system, it can probably handle several, so this may help.

I expect our Divison will aim for this type of connection on all our UNIX systems (Sydney, Canberra and Melbourne soon; Adelaide later this year; maybe more next year).

Cheers,

Ron B

Ron Baxter.

```
%
%
% utalk
@
UNRECOGNISED FIRST RECORD
:
UNRECOGNISED FIRST RECORD
:*dim
:CSIRONET UNIX  GATEWAY
CSIRONET TERMINAL CONNECTING TO UNIX


Password:

Wrong password.

basser40 login: ronb
ronb
Password: [redacted]

Good afternoon

DISK USAGE: 12 files + 18 blocks = 30 units total
LIMITS: 200 disk units, 10 processes, 20 pages

Sun Mar 29 13:01:14 1981
40% stty -echo
stty -echo
speed 9600 baud
erase = '; kill = '@'
even odd -nl -tabs
40% cp %man/man1/utalk.1 .
40% utalk: receive filename ? utalk.1
cat utalk.1
.TH UTALK I 25/10/77
.SH NAME
utalk \*- talk to foreign operating systems via a tty port
.SH SYNOPSIS
.B utalk
[c] [b] [l] [o] [n] [pname] [sn] [-name] [f filename]
.SH DESCRIPTION
[redacted]
Piers Lauder
( University of Sydney )
.SH BUGS
40% utalk: 656 bytes transferred
utalk: utalk.1 done
rm utalk.1
40%
connect  time      2:51.00
user cpu time         0.40
sys  cpu time         3.10


basser40 login:
DISCONNECT SEEN FROM USER
DISCONNECT SEEN FROM UNIX
LOGOFF FROM CSIRONET UNIX GATEWAY
     1905  CHARACTERS TRANSMITTED
      201  SECONDS CONNECT TIME
:
QUIT
%
%
%
%
%
```

Press <LINEFEED> to see if CSIRONET is really there

Terminate with LINEFEED

Now press <RETURN> a few times

the password is reduced

Switch off UNIX echo because CSIRONET is echoing

CONTROL-R command of utalk

A second CONTROL R to finish file

<DLE> T to disconnect

CONTROL-Q to quit from utalk.

CSIRO

Division of Computing Research
P.O. Box 1800, Canberra City, A.C.T. 2601 Telephone 43 3299 Telex 62145

JEP:LJD

11 March 1981

REF: SP1/6/1

Dr R Baxter
Division of Mathematics & Statistics,
CSIRO
P O Box 218
LINDFIELD N S W 2070

Dear Ron

CSIRONET CONNECTION TO UNIX

I apologise for the delay in replying to your letter of 25 November last, but I understand that you have been having informal discussions with Mark Palandri.

We are currently investigating the possibility of installing the UNIX gateway software, developed by John Gibbons in Sydney, in a new CSIRONET node type. This node type would require 28K words of memory and would use a DL11 asynchronous interface to communicate with the UNIX system.

Several problems need to be overcome before a usable system can be produced. The most major of these is to enhance both the software written by John Gibbons and also the UNIX software to perform some form of error checking and recovery on the communications line. The software used in the experimental connection in Sydney assumes that the communication line is error free and we do not believe that this is a reasonable assumption for a production system. John Gibbons has discussed the need for error recovery with staff at Sydney University and they have agreed to consider adding this option to the UNIX software.

The cost to your division of connecting a UNIX system to CSIRONET would vary according to who owns and operates the node to which you are being connected.

If the node is owned by you (eg. Melbourne and Canberra), it would require a memory upgrade costing about $4500 and a DL11 interface costing about $900. Recurrent software maintenance charges would also increase from $100 to $125 per month.

The cost of establishing a UNIX connection where the node to which it is to be connected is not operated by your division is less clear. Possible options include:

(i) A node hardware and software upgrade with the capital and recurrent costs being met by your division. This option would only be open if it was agreed to by the owner of the node and if the required extra software could be accommodated with the existing node software.

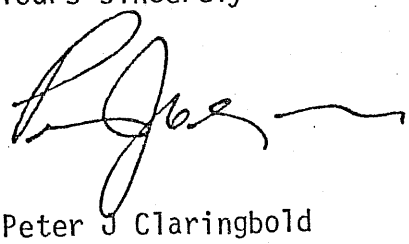
.../2.

(ii) The installation of a new 28K word gateway node to allow the connection.

(iii) Later it may be possible to use a microcomputer gateway machine to achieve the connection instead of a PDP11.

The viability of the above options will become clearer as the software investigations and enhancements currently being undertaken reach completion.

Yours sincerely



Peter J Claringbold
Chief of Division

TELEPHONE:

or: 692 2455

ref: cdr:AUUGN:1

The University of Sydney

THE DEPARTMENT OF CHEMICAL ENGINEERING

N.S.W. 2006

6th. April, 1981.

Peter Ivanov.
School of Electrical Engineering and Computer Science,
University of New South Wales
P.O. Box 1
Kensington , 2033.

Dear Peter,

I have been reading past issues of AUUGN, LOGIN and UKUUGN, and I find that I now am the owner of a very unreliable processor and a very substandard disk system. All reports indicate that PDP-11/60's, RK07s and RL01s are not usable in a UNIX enviroment. If my 11/60 and RK07s had been flakey, I would be truely enlightened as to the reasons behind their poor performance. But, in light of nearly two years of operation, I cannot see the real basis for these poor reports.

The 11/60 has had two failures in the past 20 odd months of it's operation. One was a memory controller failure, the other was a case of dirt and the 11/60 internal processor bus. The second failure was due to dirt building up on the processor board contacts, and disappeared when these were cleaned. Admittedly the symptom of the problem was a very tight loop in the mirocode, following a hardware trap via location 000000.

In the early days of UNIX at Chemical Engineering, we suffered from an inadeqaute RK07 driver and a very poor 11/60 backup routine. With some work, these problems have faded to distant memories. The machine runs 24hrs/day , seven days/week, and has experienced no unexplained crashes in the past year. There currently are 20 terminal lines connected to the 11/60 as well as three network lines, and the AUSAM(UNSW descended) UNIX seems to cope, when the system is heavily loaded.

During 1980, I developed an AUSAM UNIX running on 11/34 with RL01 disk drives.

This system has been installed in six sites (three within CSIRO, two here at the University of Sydney and one at UNSW). These systems, too have proved to be extremely reliable. One CSIRO site also has an RK06 disk system, which gives no trouble at all.

From such scanty experience, I conclude that the bad mouthing of the 11/60, RK07 and RL01 is, if not quite aprocryphal, then at least wildly inaccurate.

I can appreciate the horror and shock that one experiences coming across the RL01 or RK07 cold. The hardware leave a lot to be desired (with respect to alternative disks), but this does not constitute a poor reliability problem. Rather it represents a challenge to produce a reasonable driver to handle some pretty hairy hardware design features.

The RL01 is a real headache. It took several versions to perfect the code for a reasonable driver. This includes the funny method for reporting error conditions, where the hardware loads the silo with up to three status words per error. If these are not flushed out of the multi-purpose register, some data reliability problems develop, due to four bytes of error being used as data in the next write command.

RK07s are a bit that way as well. They are also unbuffered and hence hog the UNIBUS. This is OK, if you have a real UNIBUS, and not so good on a VAX or 11/70. The real objection to both drives is that they tend to be either moderately slow (RK07) or woefully slow (RL01), again you were warned if you bothered to read the peripherals handbook before you bought them.

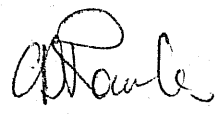
The RK07 and RL01 drivers developed here are available for distribution. I intened to include them, in the (hush-hush) but soon to be announced NSW LEVEL 7 distribution. Both drivers use finite state interrupt routines, and have the following features :

- Optimised file system layout; re. Children's Museum RK05 driver.
- Ability to handle mixed drive configurations (RK06/RK07) or (RL01/RL02) drives on the same controller.
- Cylinder Sweep seek optimisation.
- Rotational optimisation, per cylinder.
- Reasonably efficient level 6 bootstraps (up to 64 k word UNIXs)
- Raw interfaces.
- Overlapped seek.
- Some error recover retires (micro positioning and ECC for RK611 controller)

The machine dependent part of UNIX has also been modified to handle 11/60 backup routines. This works for the floating point option (FP11E) as well as the KU611 writable control store option (solves the classic 11/40 backup problem). Some more effort is required to access the hardware error log (micro-code jam diagnostics), but as the machine does not enter the jam state (why ?), this has a very low priority.

I will change to V7, which presently works for 11/60s and 11/34, when I have converted the RT-11 Fortran and Basic from V6.

Yours Sincerely,



Christopher D. Rowles

PS. This letter was prepared on a Phase 3 Sander's Media 12/7 typographical printer. it uses NROFF and a back end filter to talk to the Sanders. This filter works after a good fashion, and can handle NEQN and TBL output as well as straight text. With some further development, it too will become available.

The font used here is HELVESAN 8 ITALIC, with the mathematical font for the brackets and special symbols.

Several test outputs, in MESSENGER 12/GREEK fonts are attached. They were produced using NEQN and the terminal drive tables -TS under NROFF.

These outputs were used to help debug the Sander's filter as well as the terminal drive tables.

SANDERS TEST

The following is a demonstration of neqn and nroff equation and text formatting software packages driving a Sanders Media 12/7 printer through a filter called sand.

```
J ~=~ {sigma~u~B} over {(1~+~beta sup 2 )}
```

Produces the following equation:

$$J = \frac{\sigma \ u \ B}{(1 + \beta^2)}$$

Partial differential equations:

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} - \frac{\partial^2 f}{\partial x \partial y} = 0$$

$$\frac{\partial}{\partial x} (15r^2 + 5b)$$

Integral equations:

$$\delta^* = \int_0^\infty (1 - \frac{u}{u_\infty}) \, dy$$

Vector equations:

$$\vec{F} = m\vec{a}$$

To Peter Ivanov.

From Rob Freeth, Uni of WA.

Dear Peter,

The Computing Centre here is now hooked into CSIRONET, so we can commniucate at last! I hope this will alleviate the tyranny of distance. For you, this enhances the likelihood of contributions to the AUUGN from us.

I will also mail you a contribution to Adrian Freed's knowledge of UNIX computing in Australia. I don't yet fully understand the wondrous things your `mail' performs, so could you please forward it to him.

Did you ever get your `old-style' Memorex Controller boards from Infosys? I prodded Bert Streppelt's memory several times; he was Infosys, but has now abandoned it to start a new company called `DP Resources'. Infosys is still around, managed now by a Mr Bill Matthews; we've had very satisfactory dealings with him.

We exist at `:basser40' and `:basservax' as `uwa', and have occasionally wandered around your network. Can't think of any reason to ask for accounts on UNSW systems, except that it is fun! We currently log in every couple of days to pick up mail, etc.

Until next time,
regards,

Rob Freeth

April 12, 1981

Ross Gayler
Psychology Department
University of Queensland
ST. LUCIA
QLD 4067

Dear Ross,

I understand that the next UNIX USERS MEETING will be at the University of Queensland some time in August this year. Are you aware that there will be a DECUS conference at Brisbane on Aug 24-28 at Griffith University ? I believe the Unix meeting should be held to coincide with this conference, say on Saturday 29 Aug and perhaps some of the Unix gurus could get together and make a Unix presentation. It will certainly save on air fares for those people who wish to attend both events.

Also, in the recent DECUS NEWS there was a suggestion that "... a session in ADA and UNIX would be popular at the next Decus Australia Symposium". This seems to me to be an ideal opportunity to make such a presentation. Having once presented a paper on UNIX myself at the Townsville conference in 1977, I feel a follow-up would be in order if there were a few more supporters behind me.

I would be willing to present a paper say on SCCS (from a non-partisan point of view - hopefully more successfully this time) if other people would step forward. Could you give this matter your attention, keeping in mind that abstracts must be returned by 16 April ?

Yours faithfully,

Dave Horsfall
Computing Services Unit
University of NSW
KENSINGTON
NSW 2033

CLAYTON VICTORIA AUSTRALIA 3168

TELEPHONE: 03 541 0811 TELEGRAMS: Monashuni Melbourne
TELEX: MONASH 32691

DEPT. OF COMPUTER SCIENCE

SYSTEMS FOR TEACHING DATABASE CONCEPTS

Mini DBMS and Instructional Relational Algebra (IRA) are two systems designed and implemented to support the teaching of database concepts to tertiary level students.

Mini DBMS supports a subset of the 1978 CODASYL proposals and enables students to learn, by first-hand experience what is involved in

- designing, specifying and compiling a database schema
- choosing and defining low-level implementation options via a storage schema
- writing application programs to load, update and interrogate the database
- debugging a database program
- implementation of a DBMS.

Major components of the system include

- a table driven Data Definition Language (DDL) compiler
- a Data Storage Definition Language (DSDL) interpreter
- a Data Manipulation (DM) procedure library
- an interactive DM call interpreter (DMX)
- a symbolic debug utility.

The DDL compiler is written in Fortran 77 (and is largely compatible with earlier Fortran dialects). All other components are written in C.

IRA has been developed to allow students to learn the concepts of relational algebra by "mastery through practice". The program supports a robust, friendly user interface and an interpreter for a simplified relational algebra language.

This program is written in a relatively portable dialect of Fortran 77.

Availability

The software is currently available in source code format for a once-off fee of $50. This fee covers distribution media, documentation, handling charges and postage. The usual disclaimers concerning software correctness and no guaranteed updates apply.

If you would like more information or a distribution tape, please complete the attached form.

SYSTEM FOR TEACHING DATABASE CONCEPTS

Name: .....

Postal address: .....

.....

.....

.....

☐ I would like more information

OR

☐ Enclosed is a cheque for $50 payable to Monash University.

Distribution format: (please complete)

(1) 9 track 1600 bpi ☐

800 bpi ☐

(2) UNIX archive formats tp ☐

3rd BSD tar ☐

4th BSD tar ☐

OR

Multi file reel, fixed record size ..... bytes

fixed block size ..... records

ASCII ☐ or EBCDIC ☐

Please return to:

Dr. Ken J. McDonell,
Dept. of Computer Science,
Monash University,
CLAYTON, VICTORIA, 3168,
AUSTRALIA.

Software

Major firms join Unix parade

Transparent versions of operating system make it available for computers ranging from mainframes down to microsystems

by R. Colin Johnson, Microsystems & Software Editor

Devotees of Unix, the operating system whose responsiveness has been compared to that of a well-tuned sports car, are adding to their number almost daily. This rapid expansion of the user base of Unix, developed at Bell Laboratories and licensed by Western Electric Co., has been spurred by the emergence of user-transparent versions made for computers ranging in size from the likes of IBM System 370 mainframes down to Z80-based 8-bit microcomputer systems.

Item: Texas Instruments Inc., Dallas, long known for its comprehensive software development system, is planning to implement Unix through a subcontract with a third-party software house.

Item: Lifeboat Associates, a leading 8-bit software publisher in New York, has just signed an exclusive marketing contract with Microsoft for end-user sales of its 16-bit Xenix-11 adaptation for PDP-11s.

Item: Intel Corp.'s Ada compiler for the iAPX 432 [*Electronics*, Feb. 24, p. 119] is written in Pascal on a VAX-11/780 under Unix. (When asked why Unix was used when the final compiler release will be under VMS, Nicole Allegre, Ada program manager for the Santa Clara, Calif., company, responds, "The programmers just really wanted to use it.")

Obeys orders. Those programmers at Intel are not alone. Their counterparts across the country have been taken by Unix's responsive software-development environment. Also, the language in which the original Unix is written, C, is one of the most respected of the structured languages extant [*Electronics*, May 8, 1980, p. 129].

Since Unix was developed on Digital Equipment Corp. machines, it has been widely used on PDP-11 minicomputers for some time. However, now that Western Electric allows systems with only a few users to pay a special per-user royalty fee, it has become economical for commercial software houses to configure Unix for even inexpensive systems. An increasing number of original-equipment manufacturers and commercial software houses should start offering Unix for various other computer systems.

Unix is in fact making a strong bid to become a standard among operating systems for the new wave of 16-bit microsystems, though it faces stiff competition from the entrenched operating system family from Digital Research, Pacific Grove, Calif. When that company's 16-bit implementation of its MP/M becomes available, it will include many of the facilities that make Unix so desirable—plus CP/NET, which allows both 16- and 8-bit microsystems to share expensive peripherals. OEMs can look forward to a rich selection of system-level software packages from which to choose. Even the 8-bit microsystems are acquiring Unix-like capabilities without having to sacrifice CP/M capability.

Drawbacks. Unix is not without its critics. They say that the system cannot be used easily by clerical personnel and cite difficult operations, like rebuilding the linked list that describes the hierarchical file structure after a system crash. Some say that Unix does not provide adequate file-protection systems to make it completely trustworthy in commercial uses.

Such criticism stems from Unix's initial target: cooperative multiprogrammer software projects in which most of the users were professional computer specialists. That is why many of the facilities provided by it are specifically aimed at efficient

UNIX AND UNIX-LIKE OPERATING SYSTEMS

Processor or computer	Company	Name	Bell Laboratories' version	Original implementation
Z8000	Zilog Microsoft	Zeus Xenix	√ √	
Z80	Cromemco Morrow Designs	Cromix μNIX		√ √
LSI-11 and PDP-11	Whitesmiths Microsoft Mark Williams Co.	Idris Xenix-11 Coherent	√	√ √
6809 68000	Tech System Consultants	Uniflex		√
C/70	BBN Computer	Unix	√	
470	Amdahl	UTS	√	
All Perkin-Elmer 32-bit Machines	Wollongon Group	Unix	√	

Source: *Electronics*

program development. On the other hand, Unix is probably best known for its document-preparation and -management functions, which are often used by nonprogrammers. And with the addition of a good screen-oriented editor, like Zilog's visual editor, Unix offers a wide avenue of capability for professionals and nonprogrammers alike.

New version. One of the latest Unix versions is the Zeus adaptation by Zilog Inc. Cupertino, Calif., for its Z-Lab software development system using the Z8000 [*Electronics*, March 24, p. 120]. And to be released next month to selected OEMs is the Z8000 version called Xenix from Microsoft in Bellevue, Wash. [*Electronics*, March 24, p. 34]. Among the first of the OEMs is Codata of Sunnyvale, which is working on a floppy- and hard-disk–based microsystem that makes use of a Multibus-compatible central processing unit. Later this year, the 8086 version of Xenix is to be delivered to Altos Computer Systems of Santa Clara for its single-board 8086-based microsystem.

After that, Microsoft plans to release a 68000 version (as does Whitesmiths Ltd. of New York in an original implementation), with an eye to the iAPX-432 and the 16000 in an attempt to establish Xenix as the standard version of Unix for 16-bit microsystems. Not only is Microsoft dedicated to marketing Unix, but it is also dedicated to using it: all product development programming in its Consumer Products division is done in C on a PDP-11/70 under Unix and then transported to the target microsystem.

The first computer to which the operating system was transferred from the one on which it was developed was the Interdata 8/32. The Wollongon Group of Palo Alto, Calif., now offers Unix for the 8/32, as well as for the rest of Perkin-Elmer's 32-bit minicomputers (Perkin-Elmer having bought Interdata).

The same. In the Wollongon offering, a supreme attempt has been made to make this implementation virtually identical to the original as it appears to the user, in the interest

Probing the news

of program portability and of preserving a common command language across Unix systems.

Unix is also available from Amdahl Corp. for its IBM 370 look-alike, the 470 mainframe, and even for a computer that is specially optimized for the C language—the C/70—from BBN Computer Corp. [*Electronics*, Nov. 6, 1980, p. 46]. These, like the others, are licensed by Western Electric.

However, before the licensing procedures were changed to accommodate small systems, several software developers began work on Unix look-alikes. These user-transparent, yet original, implementation projects are now coming to fruition.

One that has been around for more than a year is Whitesmiths' Idris [*Electronics*, March 24, 1981, p. 125]. Some of the newer ones are aiming at the 8-bit market to maintain compatibility with current software bases. Two, for Z80-based microsystems using the S-100 bus, come from Morrow Designs of Richmond, Calif., and Cromemco Inc. of Mountain View, Calif., respectively.

Subtasks. Morrow Designs' version, called μNIX, runs CP/M as one task within its multiuser environment, thereby maintaining compatibility with CP/M software while gaining the conveniences of a user-transparent Unix. The emphasis throughout has been on compatibility and portability; μNIX is written entirely in Whitesmiths' C, which is not supplied with the package. Cromemco's version runs the CDOS operating system as a subtask and maintains compatibility with that already extensive software base, including its new C compiler.

There is even a version, from Technical System Consultants Inc., for Southwest Technical Products Corp.'s 6809-based 128-K-byte microsystem. Called Uniflex, it is written entirely in assembly language and includes most of Unix's features; it supports both floppies and a 20-megabyte hard disk. The West Lafayette, Ind., firm will add a 68000 version soon and is looking to Ada, Pascal, and C for future high-level language projects. □

Mr. Bohdan Durnota

CSIRO

Division of Mathematics and Statistics

P.O. Box 310, South Melbourne, Vic. 3205.
Telephone (03) 699 6711
Telex 35675

9/4/81

Dear Peter,

I was wandering whether you could inform me whether there have been any implementations of

1. Symbolic/Algebraic Manipulation, &
2. Simulation (esp. SIMULA 67)

languages on UNIX systems, or if not, on PDP11/34 computers.

Yours Sincerely

B. Durnota

Any body know of Any?

Peter I

Commonwealth Scientific and Industrial Research Organization, Australia