
* This document may contain information covered by one or *
* more licenses, copyrights and non-disclosure agreements. *
* Circulation of this document is restricted to holders of *
* a license for the UNIX software system from Western *
* Electric. Such license holders may reproduce this *
* document for uses in conformity with the UNIX license. *
* All other circulation or reproduction is prohibited. *

US User Group Meeting

The first 'letter from America' from Ianj, our roving reporter has arrived and appears later in this issue.

Ian attended the US User Group meeting in Bolder Colorado and apart from getting in some skiing he also got in some good talking with some of the 450 attendees. He is preparing a summary of the proceedings and will send that and the meeting software tape in time for publication and review in the next newsletter.

Ian!! Six dozen eh? And American too!!!

Ian Jackson from Sydney Uni also attended the US meeting, and his summary of what happened arrived in time for this issue.

Last AUUG Meeting

The last AUUG meeting was held in Canberra on 30th January. I am sure we would all like to thank George Gerrity and his staff for arranging an interesting meeting and a very pleasant lunch. The agenda and transcripts of the various talks given appear later in this newsletter.

The panel discussion at the end of the meeting brought out a number of points of general interest and in particular I would like to comment on two of them. The first is the matter of the software. Those present at the meeting were in agreement that "someone should be hired, given a machine with many megabytes of disc space and a tape drive" and turned loose on the multitude of local and overseas distributions to produce a "software catalogue". Contributions of between \$100 and \$200 dollars were mentioned and strangely nobody got up and left. In fact a show of hands appeared to be very financially reassuring.

Well the AGSM is willing to provide the machine and the tape drive, and we know of a few cheap (and skilled) workers, so all that is missing is the money. At the end of this newsletter you will find a number of questionnaire sheets. One of them asks how much your installation is prepared to pay, when, how, and to whom.

The second point is that of site configuration lists. Many of you will have received a copy of the newly revived USENIX association newsletter and accompanying documents. They produced an excellent site summary questionnaire which I have adopted for our own use, and if all you people out there would

like to fill it in I will add the information to my database, and send a copy of the whole lot to USENIX in the US.

Future AUUG Meetings

The next AUUG meeting will be held at Sydney University, hosted by Jeff Cole and Arthur Watson, from the University Computer Centre. A tentative date (2nd July) has been set and we realise that this may conflict with some University term times, but other dates conflict with other peoples session times. Complaints and suggestions should be directed to Jeff.

Robert Elz, from the Uni of Melbourne, was complaining at the January meeting that the proposed World UNIX Meeting, to held in Melbourne around the time of the Eighth World Computer Congress and Exhibition, looked like being cancelled due to lack of interest. We resolved that we should proceed with organisation of such a meeting.

I should like to voice a protest at the apparent lack of enthusiasm about what I consider to be worth while and potentially very informative project. The possibility of cutting through the isolation from the mainstream of computing that we all experience in Australia interests me greatly.

A recent enquiry from Alan Mason of the UKUUG about a 'rumour' of a meeting in Australia gives me hope that people overseas are interested. Hence the loose sheet enclosed with this issue of AUUGN. Please circulate this sheet as widely as possible among people at your site, among other UNIX contacts, and (foreign newsletter editors please note) among your readers.

See you in Australia in October.

USENIX - alive and well ??

As mentioned above, after a long absence USENIX has burst back on the user group scene. Their latest newsletter contains only "reflections of a scribe" (published in AUUGN Vol I no VI) and the covering "login:" rave. As we have already seen most of this issue, I have included only a copy of the rave.

In this you will see mentioned "exchange arrangements" with other user groups. I have already arranged with the UKUUG for exchange of newsletters (both past and present) and software, and Ianj is negotiating for us with the US and Canadian user groups for similar arrangements. We here at UNSW (where the software catalogue will be maintained) are willing to act as a central collection point for information and software from the US, UK and Canada. Full copies of the overseas newsletters will be available on request, although I think most of the interesting stuff will be published in AUUGN.

Quote of the Month

From "Decus RSTS-11 Sig" November 1979, page 13. Channel 13 -- Feedback Sessions, San Francisco, Fall 1978.

Question.

Could a "RSTS/E-PLUS" system be developed which exploits the I and D-space capabilities of the 11/45 and 11/70? Which DEC operating systems use this

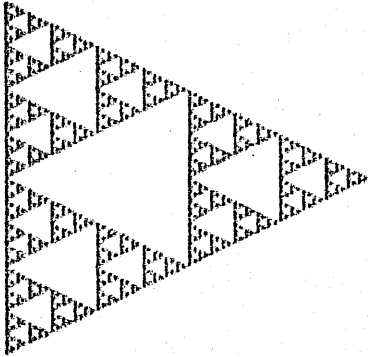
hardware.

Response.

We don't have any plans to change RSTS/E in that way. What benefits do you expect to get from the use of I and D-space? The answer to the second part of the question is that, as far as I know, the only DEC operating system to use it is RSX-11M Plus.

What benefits? Good grief!! We know of another system don't we.....

Some more artwork



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

(02) 662-3781

Thursday January 24, 1980

Pete,

Well I guess it is fair to say that I have settled down at last in New Jersey. It takes a reasonable while to get organised. The weather has been great although a little cold, it has only snowed twice and most days are fine and sunny. Altogether not hard to take except it is a little cold for rafting. All I can say about american beer is that it is cheaper. As Monty Python said when appearing in new York 'Now that I have had a chance to taste your american beer I can see that it is like making love in a canoe; F...ing close to water'.

I am also settling in quite nicely at Bell Labs. I must say it is quite nice to be able to concentrate on doing what I like to do rather than doing what has to be done. If you see what I mean. The group I now work in has the title "UNIX Networking and Special Development". I say now because there has been a reorganization of the lab that my department is part of.

The department's computer room has a reasonable array of hardware. I won't bore you with all the details, but I will say that there are two VAX 11/780s, two PDP 11/70s, one PDP 11/45 and a PDP 11/23. One of the VAXs supports the main timesharing load for the department and typically 30 to 35 users are signed on during the day. The other VAX is currently used solely for system development. One 11/70 runs UNIX/TS and the other UNIX/RT. The 11/70s are used both for production and system testing.

I have spent a deal of time familiarising myself with the VAX 11/780 and UNIX/TS version 1.2 which is the UNIX that is used here. Getting to know the VAX is no easy matter since the Architecture and Hardware books that describe the hardware leave a lot to be desired. What ever happened to the documentation standard that existed for the PDP11 ?? Aside from the fact that they contain numerous errors and omissions they are written for an intended audience that escapes me. For example "Since they are hardware registers within the processor, they provide speed advantages when used for operating on frequently accessed variables" appears in the architecture handbook - I am not sure why !! UNIX/TS on the other hand is a real clean, reliable system. It is a level 7 style system but the code is notably superior to that of the distributed UNIX. A lot of work has gone into it to make it reliable and it owes a lot to the local guru Larry Wehr whose baby it is.

I have constructed a mechanism to profile the UNIX system on the VAX. It is nothing special and uses the usual software techniques of taking an interrupt and using the program counter and processor status saved by the interrupt mechanism to keep a set of counters that record which areas of the system code are being used. I have a program that uses these counters to determine on a procedure by procedure basis where the system spent its time. The usefulness of this approach can be limited by the amount of core used to store the counters and I have a few ideas about using the VAX memory management to greatly reduce the size of the table needed. Now that this tool (buzzword) has been developed I am going to analyse the system during normal running to see what improvements and alterations can be made to improve performance. It will also be used to measure the effect of changes to the system.

I have also got interested in VAX performance from the viewpoint of how long it takes to execute an instruction. While attempting to measure the degradation due to profiling being active in the system I became rapidly aware that it wasn't going to be that easy to do. The VAX has several caches as well as instruction look-ahead and these combine to make it complicated to measure instruction speed. The alignment of instructions can be of crucial importance to the speed of execution. The VAX can be very fast but it can also run very slowly. There is no table such as appears in PDP11 processor handbooks giving formulas and the necessary data to calculate what timing should be.

As part of my interest in UNIX system performance I am also investigating the code that the C compiler produces. Some of the code that currently is produced leaves a lot to be desired. I am going to use similar techniques to Greg Rose (UNSW) when he investigated the PDP11 C compiler. That is I will look at both the static and dynamic instruction usage.

UC Berkley are now distributing a paged version of UNIX, details of which you already have. A tape for our perusal has been received and a few preliminary tests have been made. For the sort of workload that we run here it does perform worse even though they have increased the disk block size to 1024 bytes. Increasing the disk blocksize in this way on a normal UNIX system results in a marked increase in performance. Full evaluation of the larger blocksize and the Berkley system are now being undertaken. The Berkley system is not robust and it has been crashed several times.

I and a few other people have been wondering what exactly could be done with a command language that "is compatible with the following operating systems: CDC KRONOS, NOS, SCOPE, NOS/BE, DEC UNIX RSTS/E RSX-11, Univac EXEC8, IBM DOS, OS, MVS, and probably others". This quote was taken from a report entitled "The enhanced network" by Dr. John Gergen of the CSU at UNSW.

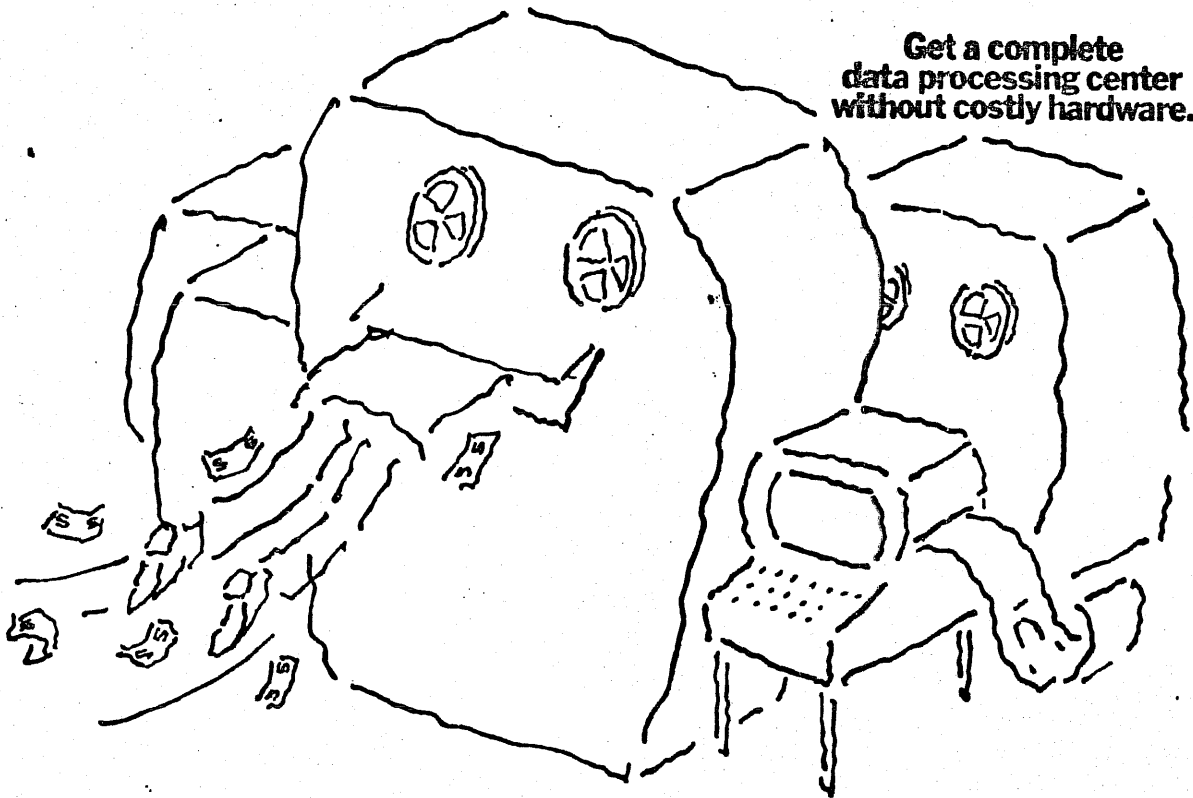
I will now mention a few of the rumours that are circulating. A new version of the DZ11 is about to come out and is to be called a DZ11-H apparently this one handles modem signals correctly. That is to say it uses interrupts to control modems as compared to the polling that was previously necessary with the DZ11-E. A smaller VAX called the 'comet', and a faster one called the 'nebula' are in the pipeline. The 'comet' will have a different instruction set to the 780. The differences are the interlock instructions which are either new or implemented differently. A new instruction set ROM for the 11/780 will be available. The 'comet' will not have a MASSBUS for at least a year, so only UNIBUS devices will be able to be used. The way multiprocessing is to be handled is via multi-ported memory that can be accessed on several SBIs. It is not clear how this will work.

I am going to the next US User Group meeting in Boulder Colorado where it is being held next week. However I am flying over tomorrow so that I can take in some of the wonderful scenery of the rockies.

Well I hope I haven't rambled on too much. Perhaps if I am not so rushed next time I can phototypeset my letter.

IanJ

**Get a complete
data processing center
without costly hardware.**



Now there's a software package that can turn a minicomputer into a small-scale data processing center with from 5 to 40 terminals. The UNIX™ System.

UNIX Systems are time-sharing operating systems that are easy to program and maintain. So easy, in fact, that more than 800 systems are already in use outside the Bell System.

UNIX Systems give fast and efficient data processing. They feature more than 100 user utilities.

UNIX System, Seventh Edition, and UNIX/32V System. The new UNIX System, Seventh Edition, offers greatly enhanced capabilities, including a larger file system and inter-machine communications. The Seventh Edition is designed for PDP-11 minicomputers. For those needing its capabilities on a larger machine, the UNIX/32V System is available for the VAX-11/780. The Seventh Edition's improved portability features allow users to adapt it more easily to other computers.

Both the UNIX System, Seventh Edition, and the UNIX/32V System can support up to 40 users with FORTRAN 77 and high-level "C" languages.

Programmer's Workbench. For large software design projects, the PWB/UNIX System (Programmer's Workbench) allows up to 48 programmers to simultaneously create and maintain software for many computer applications. The PWB/UNIX System features a unique, flexible set of tools, including a Source Code Control System and a remote job entry capability for the System/370.

Developed for our own use, UNIX Systems are available under license from Western Electric and come "as is". With no maintenance agreements, no technical support.

For more information about UNIX Systems or other Bell System software, complete the coupon and mail to Bell System Software, P.O. Box 25000, Greensboro, N.C. 27420. Or call 919-697-6530.

UNIX is a Trademark of Bell Laboratories.

To: Bell System Software,		H1
P.O. Box 25000, Greensboro, N.C. 27420		
Please send me more information about Bell System software packages.		
<input type="checkbox"/> UNIX Systems. <input type="checkbox"/> Other Bell System software packages.		
Name _____		
Title _____		Company _____
Address _____		
City _____		State _____ Zip _____
Telephone _____		Hardware _____



Highlights of the USENIX Conference
Boulder, Colorado
January 1980

Ian Jackson

Basser Department of Computer Science
University of Sydney

Eric Morton of Sanders Technology told me that their Data Media dot matrix printer is being implemented for UNIX, RT11 and other DEC operating systems. If you order one now, be sure to order one that is field upgradable to their new forthcoming midyear model or to delay delivery until this model is available. The new model will have programmable type fonts, can act as a Versatec replacement but of higher quality, and will accept nroff and troff output directly. Drivers for it will be available through DECUS. He incidentally also spoke of using Toronto's RT11 emulator under UNIX and the Whitesmith's C compiler under that to produce 8080 code. Thus \$10000 would yield an albeit somewhat inefficient development system for the Z80.

At the Software Tools conference that immediately preceded the USENIX conference, there was much discussion on the content and form of a Software Tools distribution tape. There seemed almost enough agreement that something will get distributed. The Addison-Wesley tape was seen to be inadequate.

Ian Johnstone feels that throughput on a VAX/UNIX can be doubled by using double page size and paging. VAX appears to take at least 40microsecs to field an interrupt. One day we may see a PWB 2.0 released from Western, although Al Arms later said that Western currently has no plans for such a release.

Bill Munsen and his boss Peter Jessel work in DEC's Telephone Products Division and are very concerned that DEC products work well under UNIX and that UNIX users are happy. Next time your DEC salesman says ...

Brad Cox of Hendrix Electronics is working on newspaper automation tools. He plans to complete a TI990 C compiler in about 2 months and will probably make it available through Whitesmiths. He reckoned it takes about 2 weeks

using LEX and YACC to implement a fast and dirty C compiler for a new machine.

Mark Pearson of Yourdon Inc spoke of his C compiler under RSX11-M which produces P-code for the Western Digital Microengine and in May he expects to produce an EBCDIC version for the IBM370. \$3500 first CPU, \$500 for others. His OMNIX operating system (internally unlike UNIX) works on a Z80. Basic price is \$350. They also sell supported single user UNIX systems.

Ben Woznick is working on a C compiler for the M68000

Bill Munson of DEC admitted that the RL06, RL01 and 11/60 had had their reliability problems. He intends that he should review DEC products for their applicability to UNIX before they are released. His group spends more money on UNIX than RSX11-M and runs internal training courses on UNIX. A US DECUS SIGUNIX has now been permitted. He was surprised that supervisor mode works because DEC never tested it. Mark Bartell is the ersatz chairman of US DECUS SIGUNIX. Bill said that a VAX11/580, a smaller version of the /780, should be announced this year and mentioned the price of \$80-90,000.

Al Arms of Western Electric said that the Justice Dept is of the opinion that the WECO's licensing agreements are compatible with the consent decree and so present licencing arrangements will continue. ADAPSCO's complaint is thus rejected. He announced a new small systems license at \$700 for one user up to \$9400 for some larger number of users that I did not catch. WECO is required to license things as they are being used at some moment in time in the Labs. A license (cost \$0) for the equipment test package can be obtained. No tape or documentation is available from WECO although the tape might be available from Harvard.

Bill Shannon of Case Western is implementing UNIX on a Harris/6 which has such neat features as 3 bytes/word and 336 bytes/sector. Also "bytes/word" cannot be added to a pointer to get a pointer to the next word. When forking, each page is made read only and only copied for the new process if it is written to.

Paul Jalics of Cleveland State U is implementing UNIX on an IBM Series/1 which he made sound like a nice machine.

Mark Kreiger of Whitesmiths spoke of the IDRIS operating system which has UNIX compatible interfaces to the os and file system. They have a one user system for LSI/11, 8080 and the Z80 and multiuser systems on machines without memory management. They plan to move IDRIS to M68000, Z8000 and 8086 when plans for memory management for these machines have firmed up. Their system does not have such utilities

as nroff, yacc, diff and sps. They are not aiming their marketing at the personal computer market.

Gordon Kass of Amdahl has put up a UNIX environment on an Amdahl 470 for inhouse use only at present.

Walt Lazear of Advanced Business Communications has documentation for beginning users.

Jim Kulp of IIASA, Austria has done some very interesting things with signal processing so that UNIX signals can be withheld but not lost.

Lou Katz said that he expects to start distribution of USENIX tapes on April 1st but to take several months to complete distribution. Back issues of ;Login are available. USENIX employs a half time secretary to answer the phone etc and one half time computer type person. Things should thus run more smoothly.

Next meeting will be at the U of Delaware June 18-21.

Herb Edelstein of International Data Base Systems Inc. spoke of his SEED system which is a Transportable CODASYL DBMS

Dennis Ritchie spoke.

Ron Morford of the Justice Dept's Drug Enforcement Administration who is blind described and demonstrated his hardware for being an effective programmer.

James Ellis of Duke University NC described his mail system and offered his machine as a central mail exchange.

AWA has won some ATT contracts and is thinking of becoming a UNIX licensee.

Bill Joy described his work on implementing VAX/UNIX paging. He seems to use a Second Chance page replacement algorithm. Page allocation for large processes is done in chunks larger than a page. He wants to run MAXIMA which has 2Mbytes of text and >1Mbyte of data. OS size increased by 32Kbytes of which 12Kbytes was text. He changed the page and disc block size to 1Kbytes. He spoke of tools for working on the microcode.

Al Egan of Standard Memories sells VAX memory which is cheaper than DEC's and does the job in half the control logic that DEC does.

More details will be forthcoming.

AUUG Meeting - Canberra - January 30

What follows is a collection of the written versions of what was said at the last AUUG meeting.

Two talks are missing; those by Andrew Hume (AGSM) on SCCS and Richard Miller (Uni of Wollongong) on his experiences in conversion from V6 to V7 UNIX, on an Interdata 7/32. Andrew has promised to supply a copy of his talk for the next newsletter and I am sure Richard, when he reads this, will also.

UNIX USER GROUP MEETING

30 JANUARY 1980

AGENDA

- 09:30 Morning Tea
- 10:00 Introduction
- 10:10 John Lions (UNSW) *UNIX Manuals.*
- 10:15 Richard Miller (Wollongong) *Converting to 7th Edition UNIX:
One User's Experience.*
- 10:45 Piers Lauder (Sydney Univ.) *Vax UNIX.*
- 11:00 Robert Elz (Melbourne Univ.) *The Latest TTY Driver.*
- 11:15 Andrew Hume (UNSW) *SCC'S.*
- 11:30 Andrew Hume (UNSW) *Scheduling.*
- 11:45 Adrian Freed (UNSW) *The Second Pass of the Portable C Compiler.*
- 12:00 LUNCH (ANU Union)
- 14:00 Brian Lederer (CSIRO) *A Timetable Editor.*
- 14:30 Panel Discussion on UNIX Software.
- 15.30 Afternoon Tea

UNIX USER'S GROUP MEETING, 30 JAN 1980

LIST OF ATTENDEES & AFFILIATION

ARMSTRONG, John	CSIRO Division of Plant Industry
BAXTER, Dr R.I.	CSIRO Division of Mathematics & Statistics
BILSON, J.R.	University of Tasmania Computing Centre
BROWN, Roger	ANU RS Phys S
COLE, Geoff	Sydney University Computing Centre
DAWSON, Kevin	UNSW Department of Computer Science
DUNN, Paul A	University of Melbourne Computer Science Department
ELLEM, B.A.	CSIRO Division of Mathematics & Statistics
ELSWORTHY, Graeme	UNSW AGSM
ELZ, Robert	University of Melbourne Computer Science Department
FREED, Adrian	UNSW AGSM
FRIS, Dr I.	University of New England Department of Computer Science
GERRITY, George W	UNSW Faculty of Military Studies
GREVIS, Richard	UNSW Department of Computer Science
HARRIS, Lindsay	UNSW Department of Computer Science
HILL, Kevin	UNSW AGSM
HORSFALL, David	UNSW Department of Computer Science
HUME, Andrew	UNSW AGSM
IVANOV, Peter	UNSW Department of Computer Science
JARVIS, Dennis	South Australian Institute of Technology
LAUDER, Piers	Sydney University Basser Department of Computer Science
LEDERER, Brian	CSIRO Division of Computing Research
LIONS, John	UNSW Department of Computer Science
LLOYD, John	CCAIE Department of Information Science
LONGO, Joseph	University of Melbourne Department of Computer Science
MCGREGOR, Craig	UNSW Faculty of Architecture
MALAFANT, Kim	CSIRO Division of Mathematics & Statistics
MALTBY, Chris	Sydney University Basser Department of Computer Science
MILLER, Richard	University of Wollongong Department of Computer Science
MILWAY, David	UNSW Department of Computer Science
MURRAY, Susan	ANU RS Phys S
NOLAN, Geogg	Sydney University Basser Department of Computer Science
ORSZANSKI, Roman	University of Adelaide Department of Computer Science

RYAN, Doug	CSIRO Division of Computing Research
SCHAFFER, Christine	CSIRO Division of Mathematics & Statistics
SWAIN, Peter	UNSW Department of Computer Science
SAUNDERS, Munro	UNSW Department of Computer Science
TOBIAS, Geoff	AAEC, Lucas Heights
TSANG, C.P.	University of W.A. Department of Computer Science
WATSON, Arthur	Sydney University Computing Centre
YAP, Ken	Sydney University Basser Department of Computer Science

Peter Ivanov,
Editor, AUUGN,
EECF,
UNSW.

Basser, Dept. of Comp. Sci.,
University of Sydney.

25/1/80

Dear Peter,

As of today, Unix is the chosen operating system for the computing facility at the Department of Computer Science. The competition was tough, but the opposition was last seen trailing a tape containing a memory dump bound for the USA. This brief note documents our experiences with Unix on the VAX 11/780 at Basser to date.

The VAX finally became operable on Christmas Eve, and we managed to boot Unix before VMS (as a point of honour). This turned out to be an augury for the future, as VMS is still having problems at this moment, whereas Unix goes from strength to strength. Our first inspection of Unix/32V showed that it needed a bit of work to bring it up to the sort of standards to which we have become accustomed. The disk driver optimisation was positively unhelpful for instance.

Your well beloved AUSAM system has now been ported to Unix/32V, and students will be able enjoy to the fruits of an even more carefully restricted environment! Limits are now also enforced for memory usage, and we have added a few extra features such as a per-user nice for scheduling and proper real time disk quotas.

We have currently configured the system for 80 simultaneous users, 180 disk buffers, 300 texts, 400 processes, 550 inodes, 650 file structures, etc. Benchmarks have been run to simulate 64 simultaneous students, with encouraging results (the real thing of course will always be different). A 65 line pascal program runs through "pi" in 0.6 secs. (real time). A script containing two edits, two Pascal compilations of the same 65 line program (but different copies), and one run of a resulting object file, complete with random "think" delays was run: 1 took 2m19s, 64 took 7m00s.

Perhaps we should mention the 3rd Berkeley Distribution which arrived in the nick of time containing the Pascal interpreter and many other wonders. No compiler yet, but rumours of its delivery later this year are rife. However, it turns out that a program interpreted on the VAX runs only 15 times slower than one compiled on the Cyber 170/73, so we are not holding our breaths. The Berkeley Distribution also contained a new screen oriented editor called "Vi", another mail program ("Mail"), and a Lisp system with a gargantuan appetite for real memory. There is also a "terminal capabilities database" which we are incorporating into appropriate programs that should know about such things.

The VAX appears to be only 16% faster than an 11/70 on such jobs as compilation, but twice as fast on floating point, and four times faster calculating polynomials. Pity we don't do much floating point work teaching students, eh? What seems to help is having 2 megabytes of memory free for users, and scatter/gather for swapping (no memory fragmentation). Program sizes are marginally bigger due to larger per-user areas and initial stack allocations, but text sizes are generally smaller. It helps to move read

only data into the text areas (the default load flags give shared text - hence the 300 "text"s!) In fact most of the make files have elaborate schemes to do this. The largest program is therefore 2 megabytes, but god help the swap space fragmentation, and running two such processes simultaneously is worse than you can imagine. Maybe we will implement the Berkeley paging scheme after all. However, overall we are much more satisfied with the VAX's performance than we were initially led to expect, and a 64 user configuration seems to present no problems.

A few technical details for those interested:-

We have implemented hashed buffer headers, allocate buffers during initialisation, and have removed the hidden restriction to 128 buffers. We have also got inodes working properly! An Inode allocated at boot time for root solves all sorts of problems. The dynamic disk limit checking is the height of simplicity. Everyone gets charged, all the time, for allocation. We solve the problem of "lpr", "mail" etc., by linking them to root's Inode before creating the files. Memory limits are in, a simple change to a new level 7 routine "chksize". Two numbers, one limits the max size of a process (text+data+stack), the other limits the total of non-shared parts ({Sum of} p_size). We have got "async nice" and "async kill" implemented too. Other system changes are to keep free files and texts in a linked list to speed allocation, and a very simple one - for shared texts - that removes nearly all searches through the "text" table. We are thinking of hashing inodes incore also. "Least recently used" helps here, but a search of 550 inodes to see if the one you want is there is a trifle overboard! The usual problem - Bell just doesn't have 2000 angry students to placate. We have a KMC11-a to drive output to our 30 or so fast terminals, which we believe cuts system overhead by a factor of 15 or so, and allows students to use the Vdu's at 4800 baud, (VMS - which doesn't support KMC11's - limits Vdu's to 2400 baud, which is still a trifle optimistic for 30 terminals).

A few other points come to mind. The file system is of course different from level 6, in particular, data representing integers has bytes in a different order, which affects the way one looks at inodes and freelist blocks. Init behaves a little differently as there are no console switches, and comes up single user after a boot. It now invokes "rc" with 3 parameters to tell it the state of the game, and one changes init's stat by invoking it with an argument, inherently safer i think. The "ttys" file is still the same though.

In the future we are going to implement something like Berkeley's paging scheme and in the process go over to 1024 byte blocks and 7-address inodes (yet another file system change); one can then have files larger than can be fitted in a 32 bit size!

Piers Lauder
Chris Maltby

Terminal Interfaces (Part 2)

This is the sequel to an earlier preliminary comment on terminal interfaces that found its way into the pages of AUUGN (see AUUGN I/V). Since the time of that paper and the discussion that followed at the subsequent UUG meeting at AGSM, I have designed an interface between UNIX* processes and the terminal driver. This interface is the subject of this talk.

There is also (naturally) an interface from humans to the terminal driver, and this will be mentioned where appropriate, however, it is less important that this interface be standardised, as it only affects humans (who can sometimes be persuaded to alter their habits, especially when given no alternative.)

These interfaces have been implemented on the VAX and Interdata, and could easily be implemented on a PDP-11 (unlike the previous Melbourne terminal driver). However, the design makes minimal assumptions as to the implementation method, and the suggestion that most of the terminal driver be implemented in a micro-processor attached to each (or a group) of terminals is not precluded. At the current date, no progress has been made on a prototype implementation of such a scheme.

The interface from the terminal driver to a UNIX process is through the 32V and V7 system call 'ioctl'. This function has 3 arguments, a file descriptor (designating the terminal line to be affected), a command (designating the function to be performed), and the address of a data block (to or from which data may be moved).

The use of this system call relieves all the problems mentioned in the earlier paper concerning extensions to the 'stty' and 'gtty' system calls. There is now room for (almost) infinite expansion while retaining good compatibility with earlier versions. The problem of local implementation modifications or extensions is also solved, since all a site need do is to define a few of its own 'ioctl' commands and use those.

The Melbourne terminal driver (version 2) is an extension of the V7 terminal driver (which itself is an extension of the 32V driver). Eleven new 'ioctl' commands have been added (with one more on the way) to the 13 that are standard. The only incompatibilities concern the use of delays (which are no longer considered to be flags), and the redefinitions of the bits in the flags word that used to indicate delays for

*UNIX is a Trademark of Bell Laboratories.

other purposes. Note, that as the expansion of tabs to spaces is considered to be a 'tab delay' in V7 (and 32V), that the method to request tab expansion has also altered.

Flags available:

The standard V7 flags (tandem, cbreak, lcase, echo, crmod, raw, oddp, and evenp) are augmented by the following:

- ucase which causes lower case input to be converted to upper case (and which is only really useful in conjunction with 'lcase' in order to handle inverted case terminals)
- kn1 which causes a newline to be echoed after a kill character (this is always done in 32V and V7, and I am open to suggestion that this behaviour should not be optional, and that the bit could be put to better use.)
- wrap which causes the terminal driver to wrap long lines (not yet actually implemented)
- page which will (eventually) enable some form of page mode.
- dcase which allows users with upper case input only terminals to have output and echoing sent in upper or lower case, (for terminals that can display lower case, but not send it)

There are also two flag bits that together select one of four methods of echoing an erase character. These are

- 1) to simply echo the character the user typed (that is: the erase character)
- 2) to echo the character string '\b' ' ' '\b'
- 3) to echo the character erased (with a string of erasures being delimited by the users erase character)
- 4) unassigned (currently behaves like 1.)

At the current time there is one spare bit.

Apart from these alterations, none of the 'ioctl' commands

TIOCGTD TIOCSETD TIOCHPCL TIOCSETP TIOCSETN TIOCGETP

TIOCEXCL TIOCNXCL TIOCFLUSH TIOCSETC or TIOCGETC

have been altered (nor naturally have any that do not pertain to ttys).

The 'ioctl' commands TIOCMODS and TIOCMODG remain a mystery to me, if anyone has any information of their history or usage, I would be pleased to learn of it.

New ioctl commands

TIOCXBRK to send a 250 ms (approx) long space contition (line break) to the terminal (but which is not guaranteed to be available for all terminals).

TIOCGCHR

TIOCSCHR which obtain and set (resp) the following terminal special characters:

redisp which causes the current line to be redisplayed

escp which is the terminal escape character

disc which causes output sent to the terminal to be discarded (^O on RSTS)

dqueue which causes the entire input queue not yet read by some process to be displayed

delq which causes the above queue to be deleted

brkin which is the character the is input when the user hits the 'break' key (this may be anything at all, and the driver behaves just as if the user had used the pseudonym key on the terminal, or it may be 'nothing').

TIOCGDEL

TIOCSDEL which obtain and set the delays to be used when one of the following characters is sent to the terminal

newline
carriage return
tab
form feed (or vertical tab)

These delays are in clock ticks if less than 128, or specify some special delay algorithm if greater than 128 (eg: for '\r' based on current column). A delay of exactly 128 indicates simulation of the indicated function (at present this is only available for tabs and form feeds).

These commands also get/set the delay to be used when a specified terminal dependant character is output, and that

character itself. A delay of less than 128 transmits the character followed by the delay (with a zero delay being treated as you would expect). A delay of greater than 128 causes a delay computed as the delay specified minus 128 to be sent to the terminal, without the character that caused it. A delay of exactly 128 simply causes the character to be deleted from output.

Delays are not available in raw mode (which transmits 8 bit data).

TIOCGETS

TIOCSETS get / set the terminal state, that is the terminal type, its width and depth. TIOCGETS also obtains the internal driver status so a sufficiently intelligent process can find out what is going on down there (currently only used for 'stty(1)' to print 'hup' or '-hup' - which is a state in V7 not a flag).

TIOCGCTL

TIOCSCTL get / set control character echoing specifications. This includes a long (32 bits), one bit for each of the ASCII control characters, which if set indicates that whenever the corresponding character is echoed, it should be represented as ^X (with the obvious meaning). There are also two characters that give the character to be echoed when ESC (octal 033) or DEL (octal 0177) is input (as these characters are frequently used for control purposes).

Note: none of this causes anything to be sent to the terminal when it wouldn't have been before (eg: DEL is not echoed if it is the interrupt character), nor does it have any effect on data sent from some process, only on input typed from the user.

TIOCRSTO causes output to resume being sent to the terminal (negating the effect of the 'disc' control character)

TIOCMPTY which returns (via the argument address) the unsigned data 1 or 0 depending on whether the terminal input queue is empty or not (ie: 1 if a read started now would have to wait).

TIOCDELAY accepts an unsigned integer from the process, and causes output to the terminal to be delayed for that period of milliseconds (NB: not clock ticks). This is intended to be used to replace fill characters in device dependant but system independant graphics (etc). (NOT YET IMPLEMENTED)

For anyone who doesn't yet know, the following are all possible using the existing features of V7 (but not all in 32V):

setting the following control characters - interrupt, quit, stop, start, erase, kill, and an extra input line delimiter.

flushing the input and output queues

causing the terminal to hangup on last close (but NOT the inverse)

preventing terminal input flushing when flags are altered

Duplication of control characters:

The following relationships are defined to hold if the same character is used for more than one control function. Relationships not mentioned are undefined.

interrupt and quit - the character will be interpreted as quit, except that if 'break' is input as the same character, then in these conditions depression of the 'break' key will signal interrupt. (By setting the three characters equal, depression of the actual key will give quit, and depression of 'break' will give interrupt (and will be the only way to do it).

NB: if break is input as the character which would generate an interrupt or quit, then it remains effective in 'raw' mode.

start and stop - the character will become a toggle, each depression will reverse the current state (this is standard in V7, and mandatory in 32V).

Other relationships:

The character input immediately after the escp character has no special meaning, except if it is interrupt, quit, stop, or start. Except if it is escaped in this way itself, the 'escp' character is deleted from input.

Setting a control character to octal 0200 or greater will effectively disable it (there will be no way to obtain that function). Setting 'break' to be input as exactly 0200 will cause line breaks to be ignored.

Terminal independence.

In an effort to attain some degree of terminal independence, the

driver should be configured at each site, to cause a backspace sent from a process to cause the cursor to move to the left (if at all possible), and to cause a form feed to clear the screen or advance to the top of a new page. Other terminal dependant functions must be supported by user processes.

User interface

In the current implementation (which is subject to change) the following features are provided to the user.

On certain known terminal types, when the user enters a 'kill' character the line is removed from the screen, and the cursor returned to the left margin.

It is possible to erase a kill character, upon doing so the line reinstated is redisplayed (if appropriate) but only if it is done before the next data character is entered. Several of the command characters will also prevent a kill from being erased, 'kill' itself is a notable and deliberate example of this.

On VDU's characters generally vanish when erased.

Displaying the current line (or input queue) will show any line that has been killed, but might be reinstated.

Costs:

- 1) The tty structure is bigger, there is now lots more info to keep.
- 2) The tty driver itself is bigger.
- 3) Input is not currently handled particularly efficiently. For the Interdata and the VAX, there is a comparatively simple fix, that will probably cause things to be even better than they are now (particularly as less clist space is used, since erasing is done as the erase character is typed). (I know of no comparable fix for PDP-11's).

Uncertainties:

The following issues are currently undecided.

- 1) Should any character input disable 'stop' mode, or only the 'start' character (and interrupt/quit)? Currently the latter is implemented, as that is the way it always was on the Interdata and our users are used to it. 32V and V7 adopted the former.
- 2) Should 'escp' be able to escape the stop and start characters, (and even interrupt and quit) ?
- 3) Should there be a way to disable HUPCLS?
- 4) What numbering scheme should be chosen for terminal types? (Currently a set of highly irrational and meaningless numbers have been issued.)
- 5) Which of the user command keys (erase, interrupt, etc.) should be echoed, and which not? Currently erase, kill, eof, and discard are echoed, the others are not. This choice is (mostly) purely arbitrary and in some cases is clearly wrong (Eg: why should kill (one line) be echoed, but not delq (kill many lines)). Should this be switchable, or built in (as now) ?
- 6) Should processes be prevented from outputting the start and stop command characters in tandem mode? In V7, there is no restriction, in 32V there is. The Melbourne driver imposes the restriction, but my feeling is that this is incorrect.

Robert Elz,
Computer Science,
University of Melbourne.

January 1980.

1. Introduction

This paper describes an application of the "SHARE" scheduling philosophy of John Larmouth to the UNIX¹ Timesharing system. Larmouth's system has been developed in two stages: the first² described a scheduling system for a batch processing oriented system. The second³ addresses itself more to the problems of scheduling interactive computing resources. It should also be pointed out that the philosophy behind the "SHARE" system is applicable to all scheduling problems, whether they be computer jobs or patrons of a canteen.

The major aims of the "SHARE" system are

- a. an individual's usage (in the long term) of the system shall be commensurate with his allotted share of the system's resources.
- b. an individual may request, and receive, an arbitrary turnaround for his jobs, provided his usage is comparable to his share of the system.
- c. the system should be proof against attempts to 'play' the scheduling. In other words, the system should be fair (to everyone).

The basis of this system can be described very briefly as

$$T = \frac{kU}{PxS}$$

where U is the user's usage, S is his share of the machine, P is the user-assigned priority for a particular job and T is the turnaround time for that job.

When a job is finished, the usage for its owner is updated

$$U_{\text{new}} = U_{\text{old}} + PxR$$

1. UNIX is a trademark of Bell Laboratories.
2. J.Larmouth, 'Scheduling for a share of the machine', Software-Practice and Experience, 5, 29-49 (1975).
3. J.Larmouth, 'Scheduling for Immediate Turnaround', Software-Practice and Experience, 8, 559-578 (1978).

Unfortunately, the implementations discussed by Larmouth are inapplicable to UNIX in fundamental ways.

One critical parameter of a job is R_E , the user-estimated amount of resources the job will require. Running on UNIX, there is no feasible way to estimate any relevant parameter for process before it runs. Memory requirements can, and do, grow as the program starts executing. This is done by expanding stacks and 'alloc's of more memory space via 'break'⁴. The only control currently available is a time limit facility. This facility is not useful for interactive processing and in fact, its only use is in a batch processing monitor.

Another problem lies in the differing size of the quantum of scheduling acceptable to a batch and an interactive system. Scheduling of whole jobs is adequate in a batch processing environment but obviously not in an interactive environment. A process's execution profile may vary wildly and in an unpredictable fashion. UNIX needs a scheduling algorithm which is more or less continuous in nature, rather than the discrete (with large lumps) algorithm Larmouth uses.

Before the UNIX situation is discussed, it is important to realise that the SHARE scheduling system is inherently a high level scheduler. There is a need for a lower level scheduler (probably scheduling on the basis of a single parameter) which does the dirty work of swapping and activating processes. Thus, implementing the SHARE scheduler on UNIX, as on any system, will consist of two steps:

1. implementing a low level scheduler working on one parameter (call this 'priority')
2. designing an algorithm for setting a process's priority so that the low level scheduler will do the 'right thing'.

4. break is a system call that expands a program's usable data space.

2. The UNIX Scheduler

This section looks closely at the UNIX scheduler and attempts to understand its aims. To avoid complications introduced by local optimisations, the word 'UNIX' shall refer to the original Level 6 UNIX as distributed by Bell Laboratories.

There are three major parts to the UNIX scheduler which I shall denote by low, middle and high level scheduling respectively. While each interacts with each other to some extent, they can, and will be, considered as separate activities.

2.1 Low level scheduling

Low level scheduling essentially involves the system gaining control of the CPU and then passing control to another process. This activity is done by the procedure 'swtch'⁵.

'swtch' is called from within UNIX when the current process wishes to surrender control of the CPU, or at least UNIX thinks it should. In addition, 'swtch' is called after most interrupts. The process's environment is saved and the environment is switched to that of process 0, hereafter referred to as the kernel. 'swtch' then selects the executable process with the best⁶ priority and switches control to it. If no executable process is found, 'swtch' executes a 'wait'. This is quite safe as any interrupt that occurs will cause 'swtch' to be set running again iff one of the following has occurred since it 'wait'ed:

- a. a process has been made runnable (via the routine 'setrun')
 - b. a process has been given a better a priority than the one currently running (via the routine 'setrun')
 - c. the system's idea of the time reaches N seconds, where N is divisible by 4
- If any of these conditions hold, the flag 'runrun' is set.

5. the procedure 'swtch' lives in slp.c and starts at line 2178. All line references refer to the book 'UNIX Operating System Source Code Book Level Six' by John Lions.

6. UNIX considers that the more negative a priority is, the better it is.

The low level scheduling works well and in a way that is fair to all processes. The only problem that has been observed is that when the system is heavily loaded, the CPU spends most of its time in switching processes. This is because the flag 'runrun' is set most of the time due to the large number of competing processes and thus, most interrupts will cause 'swtch' to run. This problem has been cured to a great extent by making the interrupt handler for the more common interrupts not call 'swtch' at all.

2.2 Middle level scheduling

The middle level scheduling in UNIX consists of scheduling processes to be swapped in and out of core. The routine that does this scheduling is called 'sched'⁷. It consists of two intertwined loops: one is a short loop that looks to see if anything should be swapped in or out and the other handles the swapping proper.

The first is controlled by the flag 'runout'. This flag is set whenever 'sched' can find no runnable process that is swapped out. After setting the flag, 'sched' goes to sleep. It is woken up when a process has been made ready to run or after a process has been swapped out.

The other loop is controlled by the flag 'runin'. After deciding which process should be swapped in, and if there is not enough memory available for the swapping operation, 'sched' finds a process to be swapped out. Firstly, 'sched' looks for any process in core that is either waiting or stopped. If there are none of these, 'sched' checks to see if the process to be swapped in is worthy enough. The criteria for this is that it has been swapped out for at least 3 seconds⁸. If this is the case, 'sched' looks for the process which has been in core the longest. If this period is less than 2 seconds⁹, 'sched' will try again later.

7. line 1940 in 'slp.c'.

8. 3 is a fairly arbitrary figure.

9. as above, 2 is an arbitrary constant.

Other pertinent facts to note about swapping on UNIX include

- a. generally, a process image is swapped in and out as a whole. The exceptions are processes with either shared text or shared data segments. The former account for about 40% of the swapping and the latter less than 1-2%. Even when processes do not share their text or data segments, it would be advantageous to swap in their text and data segments separately. This is because of the fragmentation of main memory.
- b. swapping appears to be the bottleneck on large UNIX systems. This is especially the case if the swap device lives on the UNIBUS. The importance of the UNIBUS is due to the fact that all data accessed via the UNIBUS updates the cache. This is fine for normal operation but during a swapping operation, it is a disaster.

2.2.1 High level scheduling As we have seen, the low level scheduling in UNIX consists of finding the highest priority process in core and passing control to it. How is this priority assigned? There are three places where the priority of a process is altered:

- a. 'sleep'¹⁰. The process priority is altered here by the system as it waits for a future event such as the completion of an I/O request.
- b. 'clock'¹¹ Two things are done within 'clock'. Firstly, at every clock tick (in Australia 1/50 second) the current process has its 'p_cpu' value increased with the restriction that will not be increased beyond 255¹². Secondly, 'clock' adjusts the 'p_cpu' figure every second. This is meant to simulate ageing. The actual method used is

```

if((pp->p_cpu & 0377) > SCHMAG)
    pp->p_cpu -= SCHMAG;
else
    pp->p_cpu = 0;

```

10. lines 2078 and 2091.

11. clock starts at 3725 and pri is altered at 3817.

12. 'p_cpu' is a byte quantity and 255 is the maximum value a byte can hold.

This is the way UNIX penalises CPU bound jobs (see `setpri`). If the process has a priority greater than 'PUSER', it gets reset by 'setpri'.

- c. 'setpri'¹³ The calculation done by 'setpri' is very straightforward and needs little comment. Some comment is necessary about the infamous line 2165, viz

```

2165         if(p > curpri)
2166             runrun++;

```

This says that if the new priority 'p' we have just worked out is greater than 'curpri' (which is the priority of the current process), arrange for 'swtch' to swap processes. All seems well until you realise that priorities in UNIX are reversed in the sense that the more negative, the better. Thus the test is saying, if the new priority is worse than the current process then reschedule the CPU.

The dilemma is not too difficult to resolve if you remember (!?) that 'setpri' is called in only three situations:

1. from 'trap' after it has serviced the trap. The point of the test here is that if the current process has now a worse priority than it had before, reschedule.
2. from clock as it recalculates all the new process priorities. The test here should be the other way as we want to say that if we calculate a new priority better than curpri (i.e. $p < \text{curpri}$), reschedule.
3. from the last lines in clock, viz

```

3824         if((ps&UMODE) == UMODE) {
3825             u.u_ar0 = &r0;
3826             if(issig())
3827                 psig();
3828             setpri(u.u_procp);
3829         }

```

This code is due to an optimisation in the loop mentioned in 2). In this loop, the priority is only recalculated if it is greater than

¹³ setpri starts at line 2156 in file slp.c.

PUSER, the default user priority. This saves recalculating the priority for user processes who use little CPU time. However, as the current process, if it is a user process, is quite likely to have a priority of PUSER (because it is less than all the other runnable processes), it should have its priority recalculated as it has just scored a CPU 'tick'.

We can see, then, that the test in setpri is dependent on where it is called from. Thus, in order to clarify the issue, I would remove the test from setpri altogether and put the test in the code for trap and clock.

There is an additional bug in clock. The problem is that as we run through the proc array testing against curpri, we do not know whether or not curpri is the new priority of the current process. This is easily fixed by evaluating the new curpri (after ageing of 'p_cpu' etc) and then running through the array.

1. The Portable C Compiler

This is an advertisement for another Level 7 utility, the portable C compiler. It should interest people in the following areas:

- It forms the basis of Lint, whose virtues seem well known to the group.
- It is the production compiler for those running UNIX on VAX's.
- It is the basis of the task of running UNIX on a new machine.
- It successfully accepts the new C standard.
- It will easily form the basis of a compiler for microprocessors.
- It works.

A description of the compiler may be found in "A Tour Through the Portable C compiler" by S. C. Johnson, which is in the Level 7 manual. An interesting design feature is that it does not follow the traditional method for achieving portability, the use of an intermediate code. Instead the machine dependent tasks in each pass have been factored out into small, readily understandable functions. These are collected in suitably named files, making the bulk of the required changes purely mechanical.

The only tricky parts are the data structures required to generate good quality code, making full use of machine registers.

Adrian Freed

2. The Portable C Compiler for 8-bit Microprocessors

After a bad experience modifying someones C compiler for 6800's we decided that waiting for the Level 7 portable C compiler would be the best approach. Particularly as it supports the full C language with such goodies(?) as enumerated data types, unsigned char's, castes etc.

C is a sufficiently complicated language to make the lexical and syntax analysis a non-trivial task. The symbol table maintenance code has to be seen to be believed. Happily, this is all done in the mainly machine-independent first-pass and may be ignored for the most part.

The main task to adapt the compiler to a new machine is the code generation. Unfortunately, the 8085 has very few useable machine registers and its only saving grace is that it is almost a stack machine. The portable C compiler's code generation is based on a general (?) register machine model and some subtle tree-fiddling using things called Sethi-Ullman numbers. We suspect this model will be particularly appropriate for the new 16-bit microprocessors (8086, Z8000, 68000). It looked difficult to make the second pass believe the 8085 was a register machine, so we decided to rewrite the entire second pass for a stack machine model. The 8085 was then made to look like a stack machine with a run-time support package. The code thus produced is quite concise but not very fast. We have built in the facility to let certain operations be done in-line. However we have only been able to make use of this for very few operations.

In summary the Portable C compiler supports the full C language and whilst the task of adapting to different machines is not simple it is certainly quicker to use it than re-invent the wheel. A figure of one full weeks work seems reasonable for adapting the compiler to a new machine.

Incidentally, the Portable C compiler is written in Level 7 C and a bootstrapping problem arises. Someone else got the production level 7 C compiler going for us on Level six UNIX.

Adrian Freed and Graeme Elsworthy

Timetable Editor

Brian Lederer

CSIRO, Division of Computing Research, Canberra

University departments periodically have the job of scheduling their teaching activity. A department is required to teach a range of subjects to students in several degree courses and has available a staff of lecturers etc and a number of rooms. In essence the scheduling task is to assign time slots and rooms to meetings between lecturers and classes (the latter perhaps attended by students from several courses) so that no meetings conflict with one another or with outside constraints. If the core of a viable timetable exists from a previous session then one can make the additions/deletions/changes to produce an up to date timetable. This is a trial and error process since changes lead to conflicts requiring further changes to resolve them. The timetable editor is intended to aid this process by providing commands to enter/modify data, edit/validate data, sort, check for conflicts, produce reports, etc. There is an associated program (written by Dr J. Beale, CSU/UNSW) to provide batch sorting/conflict checking/reporting.

The timetable data is stored in a meetings file and supporting master, options and defaults files. The meetings file describes conjunctions of subjects, classes, and lecturers. For each subject at level 1 in the hierarchy of the meetings file there may be several parts at level 2 containing several classes at level 3, each class at level three being taught by several lecturers at level 4.

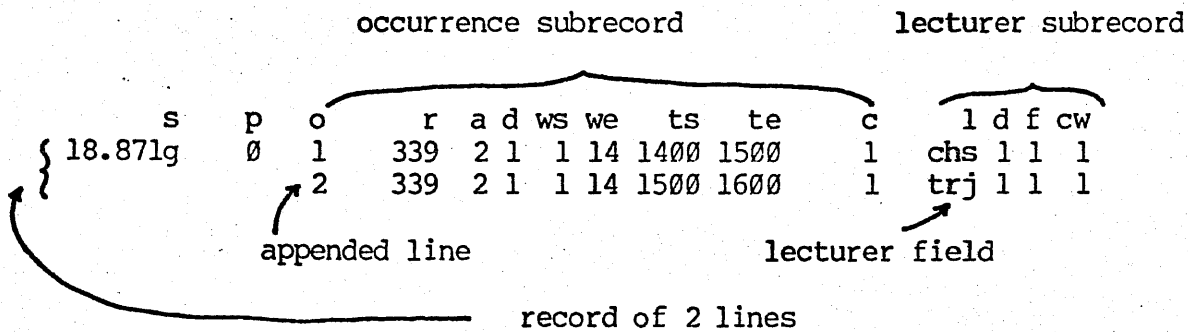


Fig. 1

Fig. 1 shows two lines from the meetings file describing two successive classes (occurrences o=1 and o=2) in a subject (s=18.871g) given on the same day (d=1) and range of weeks (week start (ws) = 1, weekend (we) = 14) by different lecturers (l = chs and l = trj). The mnemonics occurring in the meetings file, such as subject number (s = 18.871g) or lecturer's initials (l = trj), are defined in six master files: subjects, lecturers, courses, rooms, activities, and parts. At the time of data entry to the meetings file these mnemonics may be validated against the master files.

The rooms, lecturers, and courses master files permit association of day, week, and hour data (at level 2) with each mnemonic (at level 1) thus enabling time slot constraints (reservations) to be set in advance and then enforced when data is entered into the meetings file. A defaults file enables data for the meetings file to be taken from selected preset values thereby permitting a degree of specialization and automation of data entry. An options file allows for switching on and off of program options such as validation or constraint checking.

The files in core have the structure of circular linked lists which are built up when the files are read from disc. Reading/writing files is done in toto and all the data in a file must be in core at the one time. There is the concept of a current pointer and an associated pathname prompt: for example, edit commands at level 3 in line 1 of Fig. 1 would be prompted by /18.87lg/0/1. A carriage return would move the pointer to the second occurrence (o = 2) the prompt changing to /18.87lg/0/2. A temporary escape to the subjects master file would change the prompt to /18.55lg (assuming 18.55lg was the first line mnemonic in the subjects file).

The program was written in Pascal. It has been used to help generate a timetable for a department offering about 30 subjects to students in 6 degree courses with a staff of 15 lecturers. For this application it was run on a CDC Cyber 72 under the Telex time sharing system.

ARTIFICIAL INTELLIGENCE LANGUAGES ON UNIX

Claude Sammut,
Dept. of Computer Science,

UNSW

Over the last few years, we have received, on various distribution tapes, two LISP systems and a dialect of POP-2. All three have been installed on our 11/70 at different times. What follows is a description of some experiences with these systems.

LISP/11

This interpreter came to us from Princeton University. It is the version of LISP that has been in use at UNSW for the last three years. (Though I must admit the number of users is rather small - two, maybe!) I have stuck with this version up till now because it is a very neat, clean implementation of LISP, though rather unusual.

Here is a list of advantages and disadvantages of the implementation. First the advantages:

1. Most LISPs have a property list associated with each atom known to the interpreter. One of the properties may be an expression which represents a function. LISP/11 does not use property lists. Instead, an atom may only have one value. This may be a function expression. It makes passing functions as arguments to other functions very clean and easy. It also makes the language more consistent than traditional versions.
2. A small, but versatile set of string manipulation functions is included.
3. Arguments to functions may be passed in a number of different ways depending on how you define the function. The arguments may be evaluated or unevaluated, passed as a list or as individual arguments. The scheme used here is more flexible than the usual "lambda" and "nlambda" calls.
4. The interpreter is well written, and its memory management routines are simple.
5. The user manual is quite thorough.
6. The UNSW version has some local modifications to the standard syntax. These have been an attempt to reduce the need for PROGS. The changes were generally welcomed by the few users that have tried them out.

The disadvantages:

1. Property lists are almost a way of life in LISP. So programs written on other systems may not be portable.
2. I/O is unbuffered i.e. slow!
3. The documentation says that there is a switch which enables you to assemble the interpreter with or without floating point simulation. Unfortunately, the author doesn't seem to have got around to writing the FP routines! An attempt to get LISP/11 running on an 11/40 was not

ARTIFICIAL INTELLIGENCE LANGUAGES ON UNIX

successful.

4. There are no structure editor or pretty print programs which are essential if you want to write reasonably large programs. These are not very difficult to write yourself, but it is a bit time consuming.
5. Worst of all, there is no description of the systems internal workings. I discovered that it is a neat implementation only by hard work! Small additions to the system are not too bad, but major surgery is not recommended for the uninitiated.

Harvard-LISP

This is a fairly standard version of LISP. It was written at Harvard for use in some of their Mathematics courses.

Advantages:

1. As the students themselves were expected to add to the system, the internals are well documented.
2. Among the student contributions are a sophisticated structure editor, pretty printers, and a tracing package, all written in LISP. [The editor is a bit too sophisticated. It is a large and complicated set of programs - probably more than most users really want.]
3. I/O is buffered, so loading programs is not too slow.
4. An attempt has been made to keep the interpreter as close to DEC-10 LISPs as possible.
5. There is a floating point simulator for processors without the necessary hardware.

Disadvantages:

1. The interpreter prints prompts!
2. The implementation is more complicated than LISP/11's, although it is typical of most standard LISP interpreters.
3. Reading the code is fairly difficult despite the good documentation. There are conditional assemblies everywhere! Most of these only apply to Harvard's UNIX environment. (They appear to have a very unusual version of UNIX.) There is a lot of code which would never be used by most of us.

Both LISP systems are written in MACRO. After a reasonable amount of use they appear to be quite stable. As a practical system for general use, the Harvard seems to be better. However, I must admit I have become attached to LISP/11 because it is a neat program. If you are interested in archaeology, digging into LISP/11 is rewarding!

POP/11

This is a dialect of POP-2 developed at Sussex university for students doing a "cognitive studies" course. We used this interpreter for our fourth year artificial intelligence course in 1979. It proved to be quite successful. POP/11 is an excellent vehicle for teaching AI since it comes supplied with

ARTIFICIAL INTELLIGENCE LANGUAGES ON UNIX

a lot of sample programs which Sussex use in their course.

For example, there is a turtle graphics package which allows you to draw simple objects on an ordinary VDU. Another set of programs takes the same turtle commands, but draws on a TEKTRONICS terminal. Any drawing produced by the turtle graphics can be run through an edge detection program which stores its results in a simple database. Another package scans the database to produce a list of labeled corners. There are a number of other tutorial type programs in areas such as natural language programming and problem solving.

Since some of the courses at Sussex were aimed at students who have had no previous contact with a computer, there is a lot of very trivial material. In fact, if you're short of disc space, you may be in trouble. The original distribution requires over 5000 blocks! Most of this is taken up by documentation to be used as handouts for the students. The documentation is thorough, but scattered. You have to know where to look for it.

Another complication is that each subroutine in the system is stored as a separate file. In itself, this is a good idea. But there are thousands of them!!! And none of it is archived! After throwing out a lot of unwanted material and archiving and packing everything, we still need more than 3000 blocks to store it all.

In general POP/11 is a nice language to work with. It is a cross between LISP and ALGOL. This means that you have a good list-processing environment and still keep the syntax you're used to, as well as arrays and other useful things. One feature in POP-2 that is missing from POP-11 is the record data-type. This is unfortunate since records are obviously important. I suspect that records may be added in future releases of the system.

The interpreter seems to be well written. In keeping with the spirit of POP-2, only the basic language is implemented in assembler (AS), most of the surface features of the language are written in POP itself. This makes it fairly easy to dig in and change things for yourself. The assembler code is well commented.

Of the three AI languages I've used on UNIX POP/11 is the one best developed as a production system, although it is bit more complicated to install because of the huge number of files.

Sydney University recently received a tape from Berkeley which contains a version of MACLISP for the VAX. It is known as Franz Lisp! So far, no one has had the time to look at it.

More bugs in alloc and free

Since level 6 was distributed A.G.S.M. has encountered problems with the storage management routines alloc and free. The three problems were

1. the list of pointers to the allocated areas was corrupted when an attempt to 'sbrk' more memory failed. (BUG)
2. the test for finding a region big enough to allocate from ignored problems of wrap-around. (BUG)
3. 'free' was made to abort if fed a value which had not been 'alloc'ed.

Here is the current source for alloc.c at UNSW:

```
#define BLOK 512 /* sbrk for this many words each time */
#define BUSY 01

char *allocs[2] /*initial empty arena*/
{
    &allocs[1],
    &allocs[0]
};
struct
{
    int word;
};
char **allocp &allocs[1]; /* start point for searches */
char **alltoct &allocs[1]; /* top of arena (last cell) */

alloc(nbytes)
{
    register int nwords;
    register char **p, **q;
    char **t;

    allocs[0].word = | BUSY; /* static initialization */
    allocs[1].word = | BUSY;

    nwords = (nbytes+3)/2; /* round up num. bytes, plus one for pointer */
    p = allocp;
    for (;;)
    {
        /*
         * chain along list of pointers testing each free area
         */
        do
        {
            if ((p->word & BUSY) == 0) /* free */
            {
                while (((q = *p)->word & BUSY) == 0)
                    *p = *q; /* coalesce */
                if (((&p[nwords] >= p) /* end has not wrapped around */
                    && (q >= &p[nwords]))) /* free area found is big enough */
                {
                    /*
                     * if necessary, break the area found
                     * into two smaller bits (first
                     * of which is the area requested)
                     */

```

More bugs in alloc an free

```

    allocp = &p[nwords]; /* end of first bit */
    if (q > allocp) /* must break */
        *allocp = q;
    *p = allocp.word|BUSY;
    return(p+1);
}
}
q = p;
p = p->word & ~BUSY; /* chain along */
}
while (q >= allocp || p < allocp);
/*
 * we have wrapped right around without finding
 * an adequate area - so must sbrk for more core
 */
t = sbrk(BLOK * 2);
if (t.word == -1)
    return(-1);
*allocp = t;
if (t != allocp+1)
    allocp->word = | BUSY;
allocp = (*t = &t[BLOK]-1);
*allocp = allocs;
allocp->word = | BUSY;
}
}

free(p)
register char **p;
{
    register char **r, **s;

    /*
     * consistency check - is it on the list?
     */

    r = allocs;
    do
    {
        s = r->word & ~BUSY;
    }
    while (s && (s != p-1) && ((r = s) != allocs));
    if (s != p-1)
    {
        abort("free error");
    }
    /*
     * yes - turn off busy bit and reset allocp
     * (probably quite pointless)
     */

    allocp = s;
    allocp->word = & ~BUSY;
}

```

Andrew Hume
A.G.S.M.

Two items from Adrian Freed at AGSM

I have just completed implementing (in C) a packing algorithm which has the advantage of being one-pass in operation. This means you can pipe potentially large amounts of data from a program into 'pack' and subsequently 'pcat' them. It does not pack as much as Huffman encoding in most cases, but it manages 10% compression on most textfiles. The actual algorithm is based on a bounded stack.

This item is little esoteric:

A program for the display and plotting of Stefan Grossman style guitar tablature has been written in C for Tektronix terminals.

If anyone is interested in either of these programs contact me at AGSM.

Adrian Freed

UKUUG

UNITED KINGDOM UNIX USER GROUP

COMMITTEE

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

R.A.Mason
Department of Electrical and
Electronic Engineering
Mountbatten Building
Heriot-Watt University
31-35 Grassmarket
Edinburgh

CALL FOR INFORMATION - UNIX LIKE SYSTEMS

The United Kingdom UNIX User Group represents UNIX users throughout Europe with the aim of fostering interest in the UNIX operating system, its ideology, utilities, languages etc. Most of the installations we represent run Bell Laboratories supplied versions of UNIX on DEC hardware.

Recently there has been growth in the number of 'UNIX-like' (not Bell supplied/supported) systems for the equipment of a variety of manufacturers. We as a group would like to collect information on these systems so that we may decide whether or not to give user support and also so that we can pass on information to those installations whose requirements are not directly met by the DEC/BELL combination.

The type of information that we require is not simply restricted to the name of the system and its processor(s). We would like to know about devices and languages supported. In particular we would like to know about the availability of the language 'C' and to what level it is implemented, since this is the most common language for development of UNIX utility programs. Any information which we can collect would be published (subject to restrictions you give) in a condensed, and hopefully comparative, form.

I stress that our interest is not simply based on mini-computer systems but on anything from microprocessors through to mainframes. Systems being or to be developed are not excluded.

If we receive enough response we would like to consider holding a Colloquium here later this year. Representatives of all known suppliers would be invited to give presentations and open invitations to attend would be issued through various user groups and computer press.


Alan Mason

To those people, bodies and organisations whom I or the Group have contacted in the past, I apologise for the lack of personalisation in this letter, but it is being widely circulated.



Peter Ivanov
AUUGN Editor

Dear Peter,

I wrote to you earlier querying the VAX wash list since it had no heading explanation or rationale.

Your note in the latest Newsletter leaves me very concerned. If I read it correctly, the information was provided by its authors for some other purpose and then included in the Newsletter without their permission.

This is, to say the least, highly unethical. I believe that an apology to them on behalf of the Newsletter is called for.

If someone likes to do some comparisons of UNIX with VMS or any other operating system this is one thing but what appears to have been done here is something quite different and in my view reprehensible.

Yours sincerely

John A Lambert
Director

THE UNIVERSITY OF NEW SOUTH WALES

P.O. BOX 1 • KENSINGTON • N.S.W. • AUSTRALIA • 2033

TELEPHONE 663 0351

EXTN 8781

PLEASE QUOTE



February 13, 1980

John A Lambert
Director
Computing Centre
The University of Newcastle
N.S.W. 2308

Dear John,

First let me say that I did not consider the few lines you wrote on the bottom of your subscription form to be a letter requiring a reply. I treated it simply as a comment about content of the newsletter. Indeed you expressed a preference for 'UNIX wish lists may be', and I am keeping that thought in mind.

On the topic of the 'VAX-VMS wish list' I should point out that Ian Johnstone received a copy of the document from a reputable source, expressing no reservations about who might or might not read it. He published it believing it to be freely available, as was his copy.

Should the document have been private, as it later appeared, one is given to wonder at how a 'confidential document' came to lying around a tea-room table at one of our more well known institutions of learning.

Further, as it was published in a newsletter whose readers are all UNIX users, I find it worrying that copies of the newsletter appear to have fallen into the hands of non-licencees. Indeed, I am doing my utmost to ensure that all our present readship are UNIX licence holders.

As time goes on I am sure that comparisons between UNIX and VMS will be made. Until such time I think the only reasonable thing I can do is to offer DEC 'equal time', which I have done. As of this date, they have not replied, despite a few reminders.

In my view, if appologies are in order, they may be necessary from both sides. Ian seems to have come into possession of a private document, as the authors and their friends seem to have come into possession of our legally restricted newsletter.

Yours sincerely,

Peter Ivanov



Department of Computer Science

THE UNIVERSITY OF AUCKLAND

PRIVATE BAG AUCKLAND NEW ZEALAND TELEPHONE 792-300

4 December 1979

Dr Peter Ivanov,
Department of Computer Science,
University of New South Wales,
P.O. Box 1,
Kensington 2033,
AUSTRALIA.

Dear Dr Ivanov,

A contact at the University of Canterbury, New Zealand, tells me that your Department is the headquarters for the Australian UNIX Users' Group and that you have recently set up a UNIX newsletter, AUGGN. I have been highly interested in UNIX for several years now, but until recently have had neither the equipment nor the justification for it. Now at last we are setting up a Computer Science Department at the University of Auckland and we are hoping to get a PDP-11 on which we can run UNIX. We are still at the stage of applying for grants, however, and even if everything progresses smoothly we would not have it before September 1980 at the earliest. Which brings me to my first query: is it possible for us to subscribe to AUGGN, and order all back issues, without yet having a UNIX licence? If so, we would be most grateful if you could set the necessary wheels moving; I understand that the subscription is \$12 per annum, and the back issues \$10 the lot. If this proves possible, the rest of my questions may well be answered by the newsletters; please don't waste your time answering them individually if this is the case.

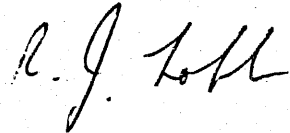
I understand that version 6 UNIX has been superseded by version 7, and that the new version requires a PDP 11/44 or upwards. We were intending, however, to order only a small LSI 11/23 System, reputed to be software-compatible with the PDP 11/34, so presumably we would be forced to stick with version 6. I would therefore be interested to know if Algol-68 (available for \$250 from an English University, I'm told) and Fortran-77 could be made to run under version 6.

Finally, I understand that MINIUNIX is operational in the Chemical Engineering Department at your university; perhaps someone there would be able to answer my remaining questions. We have several PDP 11/20s and a PDP 11/10 around the university and some of the owning departments might be interested in MINIUNIX. I understand that it will support three or four users (presumably on a 28k word system) but I am interested to know how much memory the resident operating system requires, i.e. how much is available for each user. Also, I would like to know what problem-oriented languages are available; perhaps you have an overview note that

you could pass on? On the assumption that MINIUNIX is in fact available for distribution, what medium/media do you distribute it on? Obviously we would need to obtain a licence from Bell Labs first; would a Version 6 licence suffice or is a special licence needed?

Well, that's the end of my questions; I hope they do not waste too much of your time. Many thanks for any help you can give.

Yours sincerely,

A handwritten signature in cursive script, appearing to read 'R. J. Lobb'.

Richard Lobb

THE UNIVERSITY OF NEW SOUTH WALES

P.O. BOX 1 • KENSINGTON • N.S.W. • AUSTRALIA • 2033

TELEPHONE 663 0351

EXTN3781



PLEASE QUOTE

February 13, 1980

Dear Richard,

I understand your situation fully. Trying to get money is always a long term, frustrating and sometimes disappointing thing. Your information is completely correct about costs etc, but I am afraid I cannot send you our newsletter unless you have a UNIX licence. You are also correct that most of your questions would be answered by the past newsletters, but no licence no back issues.

Western Electric now markets UNIX in a wide variety of guises including MINIUNIX, UNIX (V6), PWB UNIX, V.7 (level 7 UNIX) and 32V (VAX UNIX). The 11/23 system is relatively new, and I know of no systems locally running UNIX. If you wish to break the first ground then good luck. Personally I would go for a slightly larger machine if the money could be found. Our department teaches about 1500 students per year on a PDP11/70 and a VAX (on the way), not to mention the other smaller research and support machines about. An LS11/23 seems a little small for a Computer Science department, but then maybe I am just used to having plenty of machine power.

As you can see from the list of UNIXes available, level 6 is still alive and well so running that on a smaller PDP would be a reasonable proposition. There are some very good versions of level 6 around, notably the one we run here on anything from 11/34s to 11/70s. I don't know if Fortran-77 could run on a small machine. It may run into the address space limitation. I have not heard of the ALGOL-68 from the UK, but the chairman of the UK users group may be able to help you. His address is:

Alan Mason
Dept of Electrical and Electronic Engineering
Heriot-Watt University
31-35 Grassmarket
EDINBURGH. EH1 2HT.
United Kingdom.

I have heard of an extended ALGOL-68, called ALGOL-68S available from:

Algol 68 Distribution Manager
Dept. of Computer Science
Winnipeg
Manitoba R3T 2N2
CANADA

I have enclosed a letter from Chris Rowles at Chem Eng Dept at Sydney Uni, not here, in answer to the rest of your letter. I dont know how far you are from him, but we do have one paid-up newsletter reader in New Zealand. His address is:

John Hine
Dept of Information Science
Victoria University
Private Mailbag
Wellington
NEW ZEALAND

I think the best thing you can do is get a MINIUNIX licence now, then get the newsletter from us, while doing your best to get a nice big machine for the new Computer Science Department.

I hope I have been of help.

Yours sincerely,

Peter Ivanov

Dept. Computer Science,
University of N.S.W.
P.O. Box 1,
Kensington,
N.S.W. 2033,
Australia.

TELEPHONE
345 1844

TELEGRAMS
UNIMELB PARKVILLE



University of Melbourne

DEPARTMENT OF COMPUTER SCIENCE

Parkville, Victoria 3052

December 5th, 1979

Peter Ivanov,
Department of Computer Science,
University of New South Wales,
Kensington,
New South Wales.

Dear Peter,

Finally, my subscription to AUUGN, and not before time too I guess.

I am also including a copy of a reply I sent to a letter from Ross Gayler from Psychology at Queensland Uni, and his letter so the reply makes sense. These may be of interest to other UNIX users.

No relevant progress on telytype drivers yet, I still haven't heard from Prof Kelly to discover anything about what Qld Uni is doing. I am in the process of writing a terminal handler for our VAX (which has arrived, but isn't accepted yet, and we appear to be stuck with VMS till that happens) as I couldn't stand more than about a week of the standard terminal handler. This version will be machine independant (to the outside world anyway) and more or less fully upward compatible (a few progs like 'stty' and 'getty' will have to change.) There will be nothing that will either require or preclude the use of a micro to do most mundane terminal handling.

Yours sincerely,

Robert Elz.



University of Queensland

ST. LUCIA, BRISBANE, AUSTRALIA, 4067

ADDRESS REPLY TO REGISTRAR
BUT IN REPLY PLEASE QUOTE.

FOR ENQUIRIES REGARDING
THIS LETTER TELEPHONE

Ross Gayler,
Psychology Department,
University of Queensland,
St. Lucia, Qld. 4067.
November 5th, 1979.

Robert Elz,
Department of Computer Science,
University of Melbourne,
Parkville, VIC 3052

Dear Robert,

I see from AUUGN Vol. 1 No. 5 that you have been working your way through the Toronto conference tape. Could you please send a list of the DEC devices for which there are drivers. I am specifically interested in RLO1, RKD7 and RMD2 drivers.

I am also interested in producing simulated phototypesetter output on a Versatec printer/plotter. The 'vcat' program for this application is mentioned in the Toronto conference notes (speaker 20). Do you have 'vcat'? Could you also please let me know the most convenient way for you to send software to me.

Yours faithfully,

Ross Gayler

TELEPHONE
345 1844

TELEGRAMS
UNIMELB PARKVILLE



University of Melbourne

DEPARTMENT OF COMPUTER SCIENCE

Parkville, Victoria 3052
December 5th, 1979

Ross Gayler,
Psychology Department,
University of Queensland,
St. Lucia, Qld.

Dear Ross,

Sorry for the delay, but there is quite a lot of Toronto conference tape, and there are device drivers spread all over it.

I am including a list of most of them, the left hand column is a guide to where they came from originally. There seem to be several RK07 drivers, but the only RM02 driver appears to be a sort of general driver for lots of discs from UNSW (it uses #ifdefs to decide which sort of disc is really being used.) There are no RL drivers at all, but there is a note from Minnesota that says that a working RL-11 driver is available from

Scott Bertlinson
Rosemount Inc.,
9333 Penn Ave. So.,
Bloomington, MN. 55431
USA.

'Vcat' is on the tape, though no-one here has looked at it. I think I may have heard somewhere that UNSW use a Versatec in the way you are planning, it might be worth contacting them.

The only way that I can send you anything is on 9 track 800 bpi mag tape (odd parity). If this is OK, let me know what you want & I will send a tape up by courier. I can write 'dtp', 'tp' & 'tar' format, or ANSI standard, though our 'tp' format does not correspond to the 'tp' that Ian Johnstone distributes (it is more like the original Bell/Chicago versions).

If you cannot read 800 bpi tapes, then I suggest you contact AGSM, all of this is available from them. Similarly, if you want the whole Toronto conference distribution, (all 53 MB of it) you would also do better to contact AGSM.

As an aside, I might mention that, although there is a vast amount of software on the conference tape, not everything that is described in the conference notes is on it, in fact rather more is missing than is actually there (many speakers must have described things they didn't want to distribute, or simply didn't have with them). There is, however, a great deal on the tape that isn't described anywhere.

Yours sincerely,

A handwritten signature in cursive script, appearing to read 'K. R. Elz'.

Robert Elz.

```

UNSW      Ampex DM 9100
case      ar11 analog digital interface ( untested )
UNSW_unused Burroughs TD800 Poll Select Communications Protocol ( via dl11 )
UNSW      CDC UT200 emulator ( via dp11 )
UNSW_unused CDC UT200 emulator ( via du11 )
purdue_other Comtal display
UNSW      cr11 ( ascii & binary, buffered card reader )
case      dc11 for Tally printer
purdue_other dc11
tor       dc11
aforce    dh11 with dm11-bb
case      dh11 with dm11-bb
purdue    dh11 with dm11-bb
tor       dh11 with dm11-bb
case      Diablo model 40 on SI 3047 controller
case      dj11
UNSW      dj11
UNSW_unused dj11
case      dl11 for most printers
case      dl11 for Tally printer
tor       dl11
purdue_other dm11
tor       dma xfer device to 11/10 driving RAP
purdue_other dmc11
purdue_other dn11 ACU
tor       dn11 ACU
purdue_other dp11
tor       dp11
UNSW_unused dr11k
tor       dv disc
case      dz11 ( > 8 lines, uses RTS/CTS but no modem control )
tor       dz11 ( with [limited] IBM 2741 support )
rit       dz11
UNSW      dz11
purdue    fake tty highspeed input
aforce    GP dr11c for C/A/T
purdue_other GP dr11c for C/A/T
tor       GP dr11c for C/A/T
tor       Graphic Wonder Keyboard
tor       Graphic Wonder
tor       gt-40
purdue    Houston Electrostatic printer on dh11 with dm11-bb
tor       kl/dl11
UNSW      kl/dl11
purdue    kl11/dl11
case      LDS-2 graphics
purdue    lp11
tor       lp11
UNSW      lp11
UNSW      multiplexed tty
tor       music machine driver
purdue_other pc11
tor       pc11
case      pseudo tty
purdue    pseudo tty
UNSW      rhp04/rjp04/rm02/rm03

```

UNSW	rhp04/rjp04
minnesota	rk discs (overlapped seek)
case	rk05
UNSW	rk06/07 (tested single drive rk06)
rit	rk07 on rk611 (1 drive only)
UNSW	rk11/rk03/rk05
okla	rm03
purdue	rm03
purdue	rp04 on SI 9400 controller
aforce	rp04/rp06
purdue	rp04
tor	rp04
purdue	rs03/04
tor	rs03/04
case	rx01 floppy (RT-11 compatible)
case	rx02 floppy
minnesota	rx11 floppy (for slow DEC drives)
minnesota	rx11 floppy
rit	rx11
tor	SI disc
tor	Summagraphics Tablet
purdue	system tables
purdue	tc11 (dectape)
tor	tc11 dectape
tor	tju16
UNSW	tm11
tor	Trivial Colour Video
aforce	tu16
purdue	tu16
UNSW	tu16
purdue_other	Versatec driver (dma only)
UNSW_unused	Versatec on dl11
purdue_other	Versatec plotter
purdue_other	vt01 via dr11c to 11/20
tor	vt01 via dr11c to 11/20
tor	Xylogics Phoenix Z11 disc (CDC storage, 1 drive, untested)
tor	Zeta plotter interface

University of Essex

Department of Electrical
Engineering Science
Wivenhoe Park
Colchester CO4 3SQ

Tel: Colchester 44144 (STD Code 020 6)
Telegraphic address: University Colchester
Telex: 98440 (UNILIB COLCHSTR)

DBA/MS

7th January, 1979

Peter Ivanov,
Editor AUUGN,
Dept. of Computer Science,
Electrical Engineering,
University of New South Wales,
P.O. Box 1,
Kensington 2033
AUSTRALIA

Dear Peter,

AUUGN

With respect to your request that I pay a subscription to the AUUGN, I was assuming that as editor of the UK newsletter I would get one free, in return for sending you copies of ours, also free. If this isn't the case then I can of course raise our Australian subscription to \$12 per year! Do let me know which way we should do it.

I haven't much material at hand, but in the interest of continuity will produce the next issue in about a month.

Happy New Year.

Yours sincerely,



D.B. Anderson
Secretary, UK Unix SIG

UKUUG

UNITED KINGDOM UNIX USER GROUP

COMMITTEE

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

R.A. Mason,
Department of Electrical and
Electronic Engineering,
Heriot-Watt University,
31-35 Grassmarket,
EDINBURGH. EH1 2HT

Tel: 031 225 8432 Ext: 155

8th January, 1980.

Dr. P. Ivanov,
Department of Computer Science,
P.O. Box 1,
Kensington 2033,
AUSTRALIA.

Dear Peter,

I have recently been in contact with Mel Ferentz of the U.S. UNIX User Group and have arranged a reciprocal agreement whereby we will exchange newsletters and software, leaving each group to arrange its own internal distribution. I am very keen to establish a similar agreement with the AUUG and would be glad to hear your thoughts on the matter. If you are in agreement I would like to also suggest that we could exchange past newsletters so that we each have a comprehensive set on file.

The U.K. group represents 48 sites and though these sites are with few exceptions small, numbers are building sufficiently fast that we are hoping to step up newsletter production (at present bi-annual). By increasing the frequency of the newsletter to make it quarterly it is hoped that it will become a more active communication media.

Software development in the U.K. has dropped markedly in the past year while everyone awaited the V7 distribution. We now have it and with it the promise of more to come as our local sites look into networking, V7 on non-separate I/D machines, special purpose UNIX machines and a host of other possibilities.

I look forward to hearing from you in the near future.

Yours sincerely,



R.A. Mason

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.



PLEASE QUOTE

January 21, 1980

To: Alan Mason, Heriot-Watt University
Bruce Anderson, University of Essex

Dear Bruce and Alan,

I came into the job of AUJGN editor stone cold. I don't know who Ianj was corresponding with overseas or how he got his copies of the UKUJGN. Hence the massed reminder notices sent to every reader who had not subscribed. Ian still has what ever you sent him, so I am very interested in getting a few back issues.

As you know subscriptions to AUJGN cost \$12 (Aust.), and this covers costs quite well for production and mailing the six yearly issues. But Ian under-costed the production of sets of back issues (\$10), and I am presently only comfortably in front of the creditors.

You have both written requesting exchange of newsletters. I am more than happy to exchange with one of you, but I cannot afford both. So, I am sending Bruce a complete set of issues up to this date and crediting him with a subscription free of charge, since as editor of your newsletter he seems the logical one to get them. I assume I will receive a complete set of your issues, and future ones, in due course. I plan to write to Mel Ferentz (do you know his present address?) to arrange a similar continuing swap.

Should you, Alan, want your own set of back issues and a subscription I am afraid you will have to send me \$22 (Aust.) to get them. It may well be more cost effective to photo copy the set I am

sending Bruce.

Our next user group meeting is in 9 days time, the day before the Australian Computer Science Conference, in Canberra. We have more than 40 payed-up AUUGN readers and should get a few more at the meeting.

We also have V7, for both IIs and VAXs, but have decided to proceed slowly in the conversion to make a good job of transporting our V6 improvements. Sydney Uni has their VAX up and running and ours arrives in April-May so DEC says. From the number of VAXes being installed there should be a real flood of information on VAX V7 in the near future. Ian, who is now at Bell, is embarking on a system performance analysis of UNIX on the machines at his disposal (VAX,70,45 etc). His "reports from America" should make interesting reading in future AUUGNs.

Bruce, say hello to Phil McCrea for me and say we all look forward to his return.

Yours sincerely,

Peter Ivanov

Dept. Computer Science,
University of N.S.W.
P.O. Box 1,
Kensington,
N.S.W. 2033,
Australia.

UKUUG

UNITED KINGDOM UNIX USER GROUP

COMMITTEE

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

Alan Mason,
Department of Electrical &
Electronic Engineering,
Heriot-Watt University,
31-35 Grassmarket,
EDINBURGH. EH1 2HT

1st February, 1980.

Peter Ivanov,
Department of Computer Science,
University of N.S.W.,
P.O. Box 1,
Kensington,
N.S.W. 2033,
Australia.

Dear Peter,

Thank you for your recent letter.

I am glad you have agreed to the exchange of newsletters etc., and you were correct in assuming that these should go directly to Bruce, as Newsletter Editor. He will make copies of these and send them to our Committee Members, who in turn can dissect them so that only the pertinent content will appear in our own local newsletter. Your name has been put on our mail-list for future issues and Bruce will see to shipping you past issues.

As stated in my previous letter a similar agreement has been made between ourselves and USENIX and should you wish to make contact the current address is:

Melvin Ferentz,
USENIX Association,
Box 8,
The Rockefeller University,
1230 York Avenue,
New York, New York 10021.

The number of non-UK sites covered by our group is climbing (8 Netherlands, 4 German, 2 French, 1 Finland) and it looks as though we will soon be considering a more general European grouping. This and the growing numbers of 'UNIX-Like' systems for non DEC hardware may give us some trouble organisationally, but I think it is a safe bet to say that the reciprocal agreements will survive throughout all this local perturbation.

cont/...

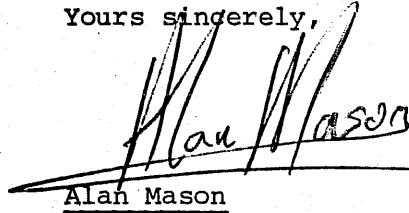
Peter Ivanov

1st February, 1980.

Like yourselves, we are in the process of converting to V7 and are following the cautious 'lets get it right' path. Our first VAX UNIX site is coming up shortly (with two more following in the near future) but to date development work has been done on 11/45. Development has been along two paths. The first involves porting certain non Bell V6 utilities and in some cases merging them with V7 facilities. The second is in the provision of V7 to run on non-separate i/d machines since over 90% of our sites have 11/34's and 11/40's. We have already a V7 version running on an 11/60 and hope to have a distributable 'all 11' package before the summer.

Finally, it has been rumoured (source unconfirmed) that the first world-wide UUG meeting will be held in the Australias - have you any information on this?

Yours sincerely,

A handwritten signature in black ink, appearing to read "Alan Mason", written over a horizontal line.

Alan Mason

University of Essex

Department of Electrical
Engineering Science
Wivenhoe Park
Colchester CO4 3SQ

Tel: Colchester 44144 (STD Code 020 6)
Telegraphic address: University Colchester
Telex: 98440 (UNILIB COLCHSTR)

DBA/MS

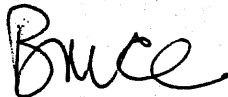
5th February, 1980

Peter Ivanov,
Dept. of Computer Science,
University of N.S.W.,
P.O. Box 1,
Kensington,
N.S.W. 2033,
Australia

Dear Peter,

Thanks for your letter of 21st January (which I can't put in our newsletter because airmail paper with typing on both sides won't Xerox!). Yes, the exchange should be with me. Our latest issue (6) should be on its way to DECUS before this is posted. Unfortunately I can't send you the back issues, as they've run out. John Lions should have them. By the way, I have AUUGN Vol. 1 numbers 1-6 already.

Yours sincerely,



D.B. Anderson.



Human Computing Resources Corporation 10 St. Mary Street, Toronto, Ontario, Canada M4Y 1P9
416-922-1937

17 January 1980

Ian Johnstone
University of New South Wales
Australian Graduate School of Management
P.O. Box 1
Kensington 2033
AUSTRALIA

Dear Mr. Johnstone:

The following information may be of interest to the readers of the UNIX newsletter.

Human Computing Resources Corporation intends to provide an increasing number of software products for the UNIX community. Currently we provide:

1. RT/EMT -- a system which allows RT-11 programs to be run under the UNIX operating system. RT/EMT will execute unchanged RT-11 binary programs.
2. HCR/BASIC -- an enhancement to standard UNIX BASIC. HCR/BASIC conforms to ANSI X3.60-1978. It also has a number of additional features, such as complete string handling, sequential and random access files, and access to UNIX commands under program control.

We intend to make other product announcements in the future. Anyone wishing information about price and availability should write or call me.

Yours very truly,

Michael D. Tilson
Software Products Manager



20 Dec 79

Peter Ivanov
Editor AUUGN
Computer Science
Electrical Engineering
University of New South Wales
P.O. Box 1
KENSINGTON 2033
AUSTRALIA

Dear Dr. Ivanov:

The Australian Newsletter is naturally interesting to us, as your Canadian counterparts. In fact, we seem to get more out of it than from the other newsletters so we don't want to miss an issue. However, we must subscribe as DCIEM (the institute where I work), rather than as the Canadian DECUS UNIX SIG. This means red tape, and a long delay. We have initiated a request for a two-year subscription, but you probably won't get it for six months or more. Could you see your way clear to sending us issues as if you had received the order?

Unfortunately, we can't reciprocate, since so far we haven't put together a newsletter. Perhaps if we can con someone into being an editor, we might do it.

Yours sincerely,

M.M. Taylor
for Chief

THE UNIVERSITY OF NEW SOUTH WALES

P.O. BOX 1 • KENSINGTON • N.S.W. • AUSTRALIA • 2033

TELEPHONE 663 0351

EXTN.

3781

PLEASE QUOTE



February 13, 1980

Dear Martin,

I read your recent letter with some dismay, since although I have pleaded with all readers not to send order forms and other such junk, I have been snowed under by the quantity of 'paper warfare' sent by various administrative departments all over the world.

Naturally I shall send issues, knowing that money will come eventually, but have you thought of what I consider a rather obvious solution.

Why don't you personally pay for the subscription, send the money to me now, and when I send a receipt back claim it under 'petty cash' or some similar method of recouping small sums of money. Surely the milk bill for the local staff room must be of about the same order of magnitude.

Failing the success of this approach I have enclosed an invoice which may allow you to trim some of the red tape.

Finally, although you can't send me your newsletters, perhaps you could jot down the odd note about what's happening over there in the world of UNIX for inclusion in our newsletter.

Merry Christmas (a bit late) and a happy new decade.

Yours sincerely,

Peter Ivanov

Dept. Computer Science,
University of N.S.W.
P.O. Box 1,
Kensington,
N.S.W. 2033,
Australia.

TELEPHONE
345 1844

TELEGRAMS
UNIMELB PARKVILLE



University of Melbourne

DEPARTMENT OF COMPUTER SCIENCE

Parkville, Victoria 3052

January 17th, 1980

Peter Ivanov,
Editor AUUGN,
Department of Computer Science,
Electrical Engineering,
University of N.S.W.
P.O. Box 1
KENSINGTON 2033

Dear Peter,

Could you include the enclosed letter (from Clive Nicholas) in the newsletter, along with a plea for anyone who can offer some assistance to please contact him directly ?

I have received very little response concerning the International Unix Users Meeting which is tentatively scheduled for Melbourne, around the same time as the World Computer Congress (IFIP'80). In addition, there has been considerable doubt as to whether the World Chess Championships would indeed be held in Melbourne (there seems to be a lack of idle 370/168's for Belle's competitors !). In the end the event has been rescheduled in Germany, earlier in the year. Consequently I seriously doubt whether Ken Thompson will be in Melbourne in October and perhaps we should revert to a normal users meeting in the September vacation.

Shortly, (31st January) I will be leaving Melbourne (where V32 Unix is running quite smoothly on the VAX) to go to Monash University where they have 5 (or more!) VAX's all running VMS. There is no prize for guessing which University which doesn't currently have a Unix licence will be acquiring one post haste- at which time I will renew my subscription to AUUGN and gladly rejoin the Users Meetings.

Regards,

Ken

Ken J. McDonell



Edinburgh Regional Computing Centre

James Clerk Maxwell Building The King's Buildings Mayfield Road Edinburgh EH9 3JZ

Ref CHN/1333 Date 20 December 1979 Telephone 031-667 1081 Ext2635

Dr Ken J McDonell
Department of Computer Science
University of Melbourne
Parkville
Victoria 3052
Australia

Dear Ken

We are users of Unix on the departmental PDP11s in the University and hope to extend the service by attaching a phototypesetter. We are trying to contact as many sites as possible who have experience in operating such a device as part of a system. I have been led to believe that there may be sites in Australia and if you have any knowledge of such sites, I would be most grateful for details of contacts that might prove fruitful. Thanking you for your help.

Happy Christmas.

Yours sincerely

A handwritten signature in cursive script that reads 'C H Nicholas'.

C H Nicholas
Manager of User Support Unit

5/2/80.

Peter,
As discussed today, we are interested in finding a color that may be used with UNIX and TEKWIN-EMER (INTERDATA).

Also please send a receipt with the newsletters

Thanks.

Jeff Ferguson.

Nepean CAE DP4
PO Box 10
Kingwood 2750.
AT&T.

16.January.1980

Dear Ian:

I doubt that you'd remember me, but we met briefly at the Toronto UNIX users' meeting. I was quite fascinated by your presentation and approached you later to ask if we could initiate some dialogue about the problems inherent with running large UNIX systems. You gave me your address and it has sat unused since then. At a recent local UNIX users' meeting I was told to get in touch with you for information on how to get the Australian UNIX newsletter. This jogged my memory and, after looking up my notes from the Toronto meeting, I recalled who you were and why I wanted to talk to you. There! That's my introduction, for what it's worth.

I am the resident systems programmer here at Northern Telecom, and I'm struggling to make the job more than just hacking. System hacking is fun, but only if one is learning a lot in the process, as happens, for example, in universities. What this means is that I try to put useful extensions onto UNIX (we are using v6) and add good software tools, both those which I perceive a need for and those deemed to be necessary by others. Currently, we are marketing a rather large automated telephone-repair service bureau system. It runs under RSX11-M and my group has been working for about a year to replace it by a UNIX-based system. I won't go into all the details, but, if you're familiar with RSX11-M and MACRO-11, you'll appreciate that we have a large problem with reliability, maintainability, flexibility, etc. My group has proposed UNIX as the answer to all these problems.

The system we are producing is called CALRS (Central Automated Loop Repair System). It is entirely table-driven, and thus has a very large data space. The code is shared by all processes running it, and in fact, that code will be [practically] the sole process running in the production (as opposed to development) environment. This has necessitated adding a shared data facility to UNIX, among other modifications. (For example, a PCL-11 driver was also added for networking capability and various system calls have been added to provide CALRS-necessitated functionality.)

To date, our development has been proceeding extremely well. We are ahead of schedule, below cost, and all manner of other good things that have made a very good name for UNIX within Northern. We are now on the verge of making this system available to the customer, of which there will be many worldwide. However, before doing so, we must take a careful look at performance. This is actually what I wished to talk to you about.

I have modified the system profiler and added a report generator to interpret it. This will be a big help in the weeks ahead. However, we have other aspects to consider. CALRS/UNIX is essentially one pro-

cess (on a PDP11/55), 64 terminals (users), and about 3 KP06 drives' worth of files shared among all the users. This is where questions of performance arise. How do we modify our DH11/tty drivers to get better response for the users? How do we modify the UNIX file system to get better throughput on accesses? What are the tradeoffs involved in various system changes we are contemplating? Should we go to V7 UNIX? Etc....

What I am hoping, Ian, is that I can plug into some meaningful information exchange. I would hate to struggle in the darkness alone, and I don't like re-inventing the wheel. I have heard that you blokes in Australia have done a lot of work along similar (i.e. performance) lines. Therefore, I would like to get information on how to receive the Australian newsletter. Would you please let me know? I am also interested in back issues. Is there any policy for making them available? I would also like to know whether you would be interested in some form of continued contact between ourselves. I do not want to "sponge" information from you. If there is any benefit you could derive from me, I would like the opportunity to establish a closer contact. If not, I shall not be offended if you indicate an unwillingness for further discourse. I imagine that you must have quite a large net of correspondence worldwide already.

In any case, I do look forward to a reply from you. Perhaps we shall see each other at the Boulder conference. I intend to be there. Hope to see you.

Yours Truly,

Stephen Pozgaj
Software Engineer
Department 1751
Northern Telecom Canada Ltd.
P.O. Box 3000
Brampton, Ontario, Canada
L6V 2M6
(416) 451-9150, X-5996

Dear Stephen,

As you may know by now, Ian is working at Bell Labs in New Jersey ~~for~~ during 1980. I will forward a copy of your letter to him. Enclosed is a copy of AUUGN and how to pay for it.

Andrew Hume

;login:

THE UNIX NEWSLETTER

VOLUME 5 NUMBER 1

JANUARY 1980

Page 1

HAPPY NEW YEAR

The effective functioning of the USENIX ASSOCIATION begins with the new year. Enclosed in this mailing are the Articles of Association and By-Laws. The Board of Directors, meeting in Toronto, adopted these and directed that the membership fees for calendar year 1980 shall be as set out below. Applications for membership are enclosed and may be duplicated if you need more copies. The Board decided that a single Institutional Membership is sufficient for all installations on a single campus or plant site. Each distribution tape will be sent to each Institutional Member. An Institutional Member may reproduce all materials we send, including the tape and this newsletter, for use only on that campus or plant site.

BACK ISSUES

We regret not having been able to answer the many inquiries about back issues. We owe all those who paid for July 1978 through June 1979 the 1979 issues and these will be sent as soon as possible. They will include, in addition to correspondence that was never published, the minutes of the Santa Monica Meeting and of several previous meetings. Rather than be in a perpetual state of catch-up, it was decided to omit issues dated July - December 1979.

MEETINGS

As you all should know by now, the next meeting will be held in Boulder, Colorado on January 30 through February 2. The Software Tools Users Group will meet in Boulder on January 29.

The committee organizing future meetings has accepted the invitation of the University of Delaware to hold the June 1980 meeting there. Details will be announced at Boulder and will appear in our February issue.

INSTALLATION SURVEY

Included with the membership form is an installation survey form, based upon that which accompanied the Boulder meeting registration form. If you have sent in the latter, you need not complete this form. We will be mounting a data base at Rockefeller University with the survey information in it so that we can expeditiously answer questions of the form, "Who has a ...".

FOURTH SOFTWARE DISTRIBUTION

Submissions for the Fourth Software Distribution may be brought to the Boulder meeting or mailed to arrive in New York before February 15, 1980. On that date, we will start packaging the distribution with a target date

Institutional Members with Educational Licenses - \$100.00
All other Institutional Members - \$300.00
Individual or Public Members - \$12.00

If Individual or Public Member requires an invoice or receipt, there will be an added charge of \$3.00 for bookkeeping.

for first mailings of April Fool's Day. Those who have made submissions to this and previous software distributions will be receiving release forms which must be signed and returned before we distribute the tape.

STAFF NEWS

We are happy to report that the two part-time employees of the association authorized by the Board of Directors are aboard. As a consequence we hope to be able to process your letters and phone calls with greater promptness than in the past. Susan Roth and Lorrin Wong have joined Armand Gazes and Mel Ferentz, who continue to serve as volunteers. While we continue to prefer written inquiries, we will try to handle telephone calls. Please avoid asking for a specific individual when you call, if possible. If you tell the person answering the nature of your call, it will be handled by the appropriate available person. The number is 212-360-1182. If we are inundated with telephone calls, we may have to restrict this service to Institutional Members but, assuming reasonableness on both sides, we would hope that wouldn't be necessary.

EXCHANGE ARRANGEMENTS

The Usenix Association, while accepting members world-wide, has no desire to compete with similar organizations in other countries. In fact, we are most anxious to enter into cooperative arrangements with Unix users' group abroad. We have spoken with the groups in Great Britain and Australia and look forward to formal agreements with these and other groups that would give each group the right to reproduce for its membership tapes and newsletters of the others. Canada is a special case in that the

people we spoke with in Toronto seemed less interested in a separate organization than in the problems of moving tapes across the border. If that is the case, we would be happy to contract with some Canadian licensee to reproduce and mail the distribution tapes within Canada.

NOTICES

This document may contain information covered by one or more licenses, copyrights, and non-disclosure agreements. Permission to copy without fee all or part of this material is granted to Institutional Members of the Usenix group provided that copies are made for internal use at the member campus or plant site. To copy otherwise, or to republish, requires specific permission.

Editorial material, payments, software submission, subscription requests, and address changes should be addressed to:

Usenix Association
Box 8
The Rockefeller University
1230 York Avenue
New York, New York 10021

**ARTICLES OF ASSOCIATION
OF
USENIX ASSOCIATION**

The undersigned, for the purpose of forming a voluntary unincorporated association under the laws of the State of New York, hereby agree:

FIRST: The name of the association is USENIX ASSOCIATION.

SECOND: The purposes for which the association is formed are:

A. To provide appropriate means and opportunities for the exchange of information and ideas among authorized representatives of holders of licenses and sublicenses for software obtained from the Western Electric Company including, but not restricted to, UNIX, PWB/UNIX, Phototypesetter, Mini-UNIX, and other operating systems written in the C programming language, all subject to the terms of the several licenses and sublicenses of such holders.

B. To foster the free exchange and communication of information relating to those aspects of the systems and programs described in A above which are publicly available and not subject to disclosure restrictions, and to any other programming language, system of programming or operating systems of interest to the members of the association.

C. The association shall have the power to purchase, hire, rent, own and hold all necessary real and personal property, together with all other property, conveniences or other things requisite in the judgement of the board of directors of the association for the implementation of the purposes of the association.

THIRD: The association is organized and operated not for pecuniary profit. Except as may otherwise be permitted by the Internal Revenue Code as now in effect or hereafter amended to organizations exempt from tax under Section 501(a) thereof (as successor sections thereto), and the corresponding laws of the State of New York, no substantial part of the activities of the association shall be carrying on propaganda, or otherwise attempting to influence legislation, and no part of the activities of the association shall be participating in, or intervening in (including the publishing or distributing of statements), any political campaign on behalf of any candidate for public office.

FOURTH: The office of the association within the State of New York is to be located in the City of New York, County of New York.

FIFTH: The sole management of the affairs of the association shall be entrusted to not less than five persons, each having one vote, who shall constitute a board of directors. The names and addresses of the initial directors of the association are:

NAME	Address
Melvin Ferentz	The Rockefeller University 1230 York Avenue New York, NY 10028
Lou Katz	Columbia University College of Physicians & Surgeons 630 West 168th Street New York, NY 10032
Mars Gralia	Applied Physics Laboratory The Johns Hopkins University Laurel, Maryland 20810

Lewis A. Law

Harvard University Science Center
One Oxford Street
Cambridge, Massachusetts 02138

Peter Weiner

Interactive Systems Corp.
1526 Cloverfield Blvd.
Santa Monica, CA 90404

SIXTH: No part of the income of the association shall inure to the benefit of the benefit of any member, director or officer of the association, or any private individual (except that reasonable compensation may be paid for services rendered to or for the association affecting one or more of its purposes), and no member, director, or officer of the association or any private individual shall be entitled to share in the distribution of any of the assets on dissolution of the association.

SEVENTH: In the event of dissolution, all of the remaining assets and property of the association shall after necessary expenses thereof be applied to accomplish the purposes for which the association is organized by distributing such property and assets for the furtherance of the work of institutions with similar purposes and objects. In the event of voluntary dissolution, such institutions shall be selected in the discretion of the directors.

EIGHTH: These articles of association may be amended at any annual, regular or special meeting of the members provided that notice of the proposed amendment is given with the notice of any such meeting.

IN WITNESS WHEREOF the undersigned, natural persons over the age of eighteen years, subscribe these articles of association this 20th day of June, 1979.

BY-LAWS OF USENIX ASSOCIATION

TABLE OF CONTENTS

Article I. Members

Sec. 1. Qualifications and Election 1

Sec. 2. Delegates 1

Sec. 3. Fees, Dues and Assessments 1

Sec. 4. Termination 1

Article II. Meetings of Members

Sec. 1. Meetings 1

Sec. 2. Notice of Meetings 2

Sec. 3. Notice of Meetings 2

Sec. 4. Quorum at Meetings 2

Sec. 5. Nominations 2

Sec. 6. Voting 2

Article III. Board of Directors

Sec. 1. Number of Directors 3

Sec. 2. Election and Term of Directors 3

Sec. 3. Newly Created Directorships and Vacancies 3

Sec. 4. Resignations 3

Sec. 5. Removal of Directors 3

Sec. 6. Meetings 3

Sec. 7. Quorum and Voting 4

Sec. 8. Committees of the Board 4

Sec. 9. Written Consent of Directors in Lieu of a Meeting 4

Sec. 10. Compensation of Directors 4

Sec. 11. Loans to Directors or Officers 4

Article IV. Officers, Agents and Employees

Sec. 1. General Provisions 4

Sec. 2. Powers and Duties of the Chairman of the Board 4

Sec. 3. Powers and Duties of the President 5

Sec. 4. Powers and Duties of Vice Presidents 5

Sec. 5. Powers and Duties of the Secretary 5

Sec. 6. Powers and Duties of the Treasurer 5

Sec. 7. Powers and Duties of Assistant Secretaries 5

Sec. 8. Powers and Duties of Assistant Treasurers 5

Article V. Indemnification 6

Article VI. Checks, Notes, Drafts, etc. 6

Article VII. Amendments 6

Article VIII. Fiscal Year 6

BY-LAWS
OF
USENIX ASSOCIATION

ARTICLE I
Members

Section 1. Qualifications and Election.

Any person, including a corporate person, who has a bona fide interest in the purposes of the Association and who satisfies the following criteria may become a member of the Association by the vote of the Board of Directors. Any holder of a license or sublicense from the Western Electric Company for software is eligible for admission as an Institutional Member; holders of such licenses or sublicenses are eligible for an Institutional Membership for each central processing unit maintained by such holder although there is no requirement that more than one Institutional Membership be held by any such holder. Any Bell System installation is eligible for any number of Non-Voting Institutional Memberships. Any individual who is employed by or affiliated with an Institutional Member and who is bound by the non-disclosure provisions of the license or sublicense held by the Institutional Member or any employee of a Non-Voting Institutional Member, is eligible to become an Individual Member, any such prospective Individual Member must furnish to the Association evidence as to qualification for such status and must notify the Association promptly if and when such status changes. Any other individual is eligible to become a Public Member of the Association. Only an Institutional Member is entitled to vote at meetings of the members of the Association.

Section 2. Delegates.

Any corporate person who is a member shall designate, by written statement addressed to the Secretary of the Association, a delegate to represent such member at all meetings and to act on behalf of such member on all matters. Communications sent to or received from delegates shall be deemed sent to or received from their respective members. Each corporate person who is a member may at any time remove any person serving as its delegate and upon the removal, resignation or death of its delegate designate a new delegate by a written statement addressed to the Secretary of the Association.

Section 3. Fees, Dues and Assessments.

Initiation fees, dues and assessments may be levied upon the members of the Board of Directors.

Section 4. Termination.

Members may be suspended or expelled by the Board of Directors for refusal or failure to comply with the By-Laws or for other good and sufficient cause, including failure to pay any fees, dues and assessments levied by the Board of Directors. Any member may withdraw from the Association after fulfilling all obligations to it by giving written notice to the Secretary of the Association, which notice shall be presented to the Board of Directors by the Secretary at the first meeting after its receipt.

ARTICLE II

Meetings of Members

Section 1. Meetings.

Meetings of the members may be called by the Board of Directors or by the President, and shall be called by the President or Vice President or the Secretary at the written demand of a majority of the Board of Directors or at the written demand of the members entitled to cast at least ten per cent (10%) of the total number of votes entitled to be cast at such meeting. Any such call or demand shall state the purpose or purposes of the proposed meeting. On failure of any officer above specified to call such special meeting when duly demanded, any signer of such demand may call such special meeting and give the notice thereof. Special meetings shall be held at such hour on such day and at such place within or without the State of New York as may be specified in the notice thereof. At any special meeting only such business may be transacted which is related to the purpose or purposes set forth in the notice thereof, but any special meeting may be called and held in conjunction with an annual meeting of the members.

Section 2 Record Date for Meetings and Other Purposes.

For the purpose of determining the members entitled to notice of or to vote at any meeting of members or any adjournment thereof, or to express consent to or dissent from any proposal without meeting, or for the purpose of determining members entitled to receive any distribution or the allotment of any rights, or for the purpose of any other action, the Board of Directors may fix, in advance, a date as the record date for any such determination of members. Such date shall not be more than fifty nor less than ten days before the date of such meeting, nor more than fifty days prior to any other action. If no record date is so fixed by the Board of Directors (a) the record date for the determination of members entitled to notice of or to vote at a meeting of members shall be at the close of business on the day next preceding the day on which notice is given, or, if no notice is given, the day on which the meeting is held, and (b) the record date for determining members for any other purpose shall be at the close of business on the day on which the resolution of the Board of Directors relating thereto is adopted.

A determination of members of record entitled to notice of or to vote at any meeting of members, made in accordance with this Section, shall apply to any adjournment thereof, unless the Board of Directors fixes a new record date under this Section for the adjourned meeting.

Section 3 Notice of Meetings.

Whenever members are required or permitted to take any action at a meeting, written notice shall be given stating the place, date and hour of the meeting and indicating that it is being issued by or at the direction of the person or persons calling meeting. Notice of a meeting shall also state the purpose or purposes for which the meeting is called. A copy of the notice of any meeting shall be given personally or by first class mail, not less than ten nor more than fifty days before the date of the meeting, to each member entitled to vote at such meeting. If mailed, such notice shall be given by depositing it in the United States mail, with postage thereon prepaid, directed to the member at such member's address as it happens appears on the record of members, or, if such member shall have filed with the Secretary of the Association a written request that notices be mailed to some other address, then directed to such member at such other address.

When a meeting is adjourned to another time or place, it shall not be necessary to give any notice of the adjourned meeting, if the time and place to which the meeting is adjourned are announced at the meeting at which the adjournment is taken, and at the adjourned meeting any business may be transacted that might have been transacted on the original date of the meeting. However, if, after the adjournment, the Board of Directors fixes a new record date for the adjourned meeting, a notice of the adjourned meeting shall be given to each member of record on the new record date entitled to notice under this Section.

Section 4 Quorum at Meetings.

Except as otherwise provided by law, the members entitled to cast a majority of the total number of votes entitled to be cast thereat (whether in person or by ballot) shall constitute a quorum at any meeting of members for the transaction of any business, but the members present may adjourn any meeting to another time or place despite the absence of a quorum.

Section 5 Nominations.

Whenever the directors shall fix a meeting of members for the purpose of electing directors and officers, the directors shall cause notice of the date of such meeting to be given to all Individual Members at least three months in advance of said meeting together with notification of their opportunity to make nominations and the procedure therefor. All Individual Members shall be eligible to make nominations of candidates for election as directors or as President or Vice President of the Association. All nominations must be submitted in writing to the Secretary at least two months prior to a meeting of members which is being held for the purpose of electing new directors and officers. All nominations must bear the signatures of at least five Individual Members.

Section 6 Voting.

Whenever the President, the Vice President or the directors are to be elected by the members, they shall be elected by a plurality of the votes by mail ballot by the members entitled to vote in the election. Whenever any action, other than the election of the President, the Vice President or the directors, is to be taken by vote of the members, it shall, except as otherwise required by law, be authorized by a majority of the votes cast at a meeting of members by the members entitled to vote thereon.

The Secretary shall compile a ballot within two weeks following the close of nominations which includes a brief summary of the qualifications of each candidate. The Secretary will mail said ballot to all Institutional Members. Each ballot must bear a due date not less than two weeks after the date of entry of the ballots into

the mails. The ballots will be counted within one week following the due date. No ballots will be counted after that time regardless of postmark. The newly elected directors and officers will be informed immediately of the results of the election and the date their term begins.

Except as otherwise provided by law, every Institutional Member of record of the Association entitled to vote on any matter at any meeting of members shall be entitled to one vote. Upon the demand of any member, the vote at any election of the President, the Vice President or the directors, or the vote upon any question before a meeting, shall be by ballot, but otherwise the method of voting shall be discretionary with the person presiding at the meeting.

ARTICLE III

Board of Directors

Section 1. Number of Directors

The number of directors constituting the entire Board shall be not less than five, each of whom must be an Individual Member or Public Member. Subject to such limitation, the entire Board shall consist of five directors until changed as hereinafter provided. The number of directors may be changed at any time and from time to time at any meeting of the members by vote of the members entitled to vote for the election of directors, or at any meeting of the Board by the vote of a majority of the entire Board, except that no decrease shall shorten the term of any incumbent director. Unless and until changed in accordance with this Section the number of directors constituting the entire Board shall continue in effect and no further action shall be required to fix such number at any meeting of the members for the election of directors.

Section 2. Election and Term of Directors.

An election meeting shall be held in each even numbered year. At each election meeting of members, directors shall be elected to hold office until the next election meeting. The term of office of each director shall be from the time of election and qualification until a successor shall have been elected and shall have qualified.

Section 3. Newly Created Directorships and Vacancies.

Newly created directorships resulting from an increase in the number of directors and vacancies occurring in the Board for any reason, including the removal of directors by the members without cause, may be filled either by vote of the members at a meeting of the members or by vote of a majority of the directors then in office, although less than a quorum exists.

Section 4. Resignations.

Any director may resign from office at any time by delivering a resignation in writing to the Association, and the acceptance of such resignation, unless required by the terms thereof, shall not be necessary to make such resignation effective.

Section 5. Removal of Directors.

Any or all of the directors may be removed for cause or without cause, by vote of the members. Any director may be removed for cause by action of the Board provided there is a quorum of not less than a majority present at the meeting of directors at which such action is taken.

Section 6. Meetings.

Meetings of the Board may be held at any place within or without the State of New York as the Board from time to time may fix or as shall be specified in the respective notice or waivers of notice thereof. Any one or more members of the Board of Directors or of any committee thereof may participate in a meeting of such Board or committee by means of a conference telephone or similar communications equipment allowing all persons participating in the meeting to hear each other at the same time. Participation by such means shall constitute presence in person at a meeting. A meeting of the Board for the appointment of officers shall be held on the day on which a meeting of the members shall have been held, at the same place and as soon after the holding of such meeting of members as is practicable, and no notice thereof need be given. The Board may fix times and places for regular meetings of the Board and no notice of such meetings need be given. Special meetings of the Board shall be held whenever called by the President or by at least one-fifth of the directors for the time being in office. Notice of each such meeting shall be given by the Secretary or by a person calling the meeting to each director by mailing the same not later than the second day before the meeting,

or personally or by telegraph, cabling or telephoning the same not later than the day before the meeting. Notice of a meeting need not be given to any director who submits a signed waiver of notice whether before or after the meeting, or who attends the meeting without protesting, prior thereto or at its commencement, the lack of notice.

Section 7. Quorum and Voting.

Unless a greater proportion is required by law, a quorum for the transaction of any business, in the event the number of the entire Board is fifteen members or less, shall consist one-third of the entire number of members, and, in the event the number of the entire Board is more than fifteen members, shall consist of five members plus one additional member for every ten members for fraction thereof in excess of fifteen. Except as otherwise provided by law or by these By-Laws, the vote of a majority of the directors present at such time, shall be the act of the Board, but a majority of directors present, whether or not a quorum is present, may adjourn any meeting to another time and place. No notice of any such adjournment need be given.

Section 8. Committees of the Board.

The Board by resolution adopted by a majority of the entire Board, may designate from among its members an Executive Committee and other committees, each consisting of three or more directors, and each of which to the extent provided in the resolution, shall have all the authority of the Board, to the full extent permitted by law. Minutes of the proceedings of any such committee shall be kept and shall be submitted to the Board at its next meeting. The Board may designate one or more directors as alternate members of any such committee, who may replace any absent member or members at any meeting of such committee. Each committee shall serve at the pleasure of the Board.

Section 9. Written Consent of Directors in Lieu of a Meeting.

Any action required or permitted to be taken by the Board or any committee thereof may be taken without a meeting if all members of the Board or such committee consent in writing to the adoption of a resolution authorizing such action. Each resolution so adopted and the written consents thereto by members of the Board or such committee shall be filed with the minutes of the proceedings of the Board or such committee.

Section 10. Compensation of Directors.

Directors may receive compensation for services rendered to the Association in their capacities as directors or otherwise in such reasonable amounts as may be fixed from time to time by the affirmative vote of a majority of the entire Board.

Section 11. Loans to Directors or Officers.

A loan shall not be made by the Association to any director or officer.

ARTICLE IV

Officers, Agents and Employees

Section 1. General Provisions.

The officers of the Association shall be a President, a Secretary and a Treasurer, and may include a Chairman of the Board, one or more Vice Presidents, one or more Assistant Secretaries and one or more Assistant Treasurers. Each officer must be an Individual Member and the President and the Vice President must be members of the Board of Directors. The President (if there is no Vice President) and the Vice President shall be elected by the members. The other officers shall be appointed by the Board of Directors at the first meeting of the Board after the meeting of the members. The Board may appoint other officers, agents and employees, who shall have such authority and perform such duties as may be prescribed by the Board. All officers shall hold office until the meeting of the Board following the next meeting of the members after their appointment and until their successors shall have been appointed and shall have qualified. Any two or more offices may be held by the same person, except the offices of President and Secretary. Any officer, agent or employee of the Association may be removed by the Board with or without cause. Such removal without cause shall be without prejudice to such person's contract rights, if any, but the appointment of any person as an officer, agent or employee of the Association shall not of itself create contract rights. Officers may receive

compensation in such reasonable amounts as may be fixed by a majority vote of the entire Board. The compensation of other agents and employees appointed by the Board, but this power may be delegated to any officer, agent or employee as to persons under their direction or control. The Board may require officers, agents or employees to give security for the faithful performance of their duties.

Section 2. Powers and Duties of the Chairman of the Board.

The Chairman of the Board (if there be one) shall preside at all meetings of the members and of the Board at which the Chairman is present and shall perform such other duties as the Board may designate. In the absence or inability to act of the President, the Chairman of the Board shall perform the duties and exercise the powers of the President. The performance of any such duty by the Chairman of the Board shall be conclusive evidence of the Chairman's power to act.

Section 3. Powers and Duties of the President.

The President shall be the chief executive officer of the Association. In the absence of the Chairman of the Board (or if there be none), the President shall preside at all meetings of the members and of the Board at which the President is present. The President shall have general charge of the business and affairs of the Association and may employ and discharge employees and agents of the Association, except such as shall be appointed by the Board, and may delegate

The President may vote the shares or other securities of any type or kind which may at any time be owned by the Association, may execute any shareholders' or other consents in respect thereof and may delegate such powers by executing proxies, or otherwise, on behalf of the Association. The Board, by resolution from time to time, may confer like powers upon any other person or persons.

Section 4. Powers and Duties of the Vice President.

The Vice President shall have such powers and perform such duties as the Board of Directors and/or the President may prescribe. The Vice President shall succeed to the office of President at the next meeting of the members. In the absence or inability to act of the President (and of the Chairman of the Board, if there be one), unless the Board shall otherwise provide, the Vice President shall perform all the duties and may exercise any of the powers of the President. The performance of any such duty by the Vice President shall be conclusive evidence of such Vice President's power to act.

Section 5. Powers and Duties of the Secretary.

The Secretary shall have charge of the minutes of all proceedings of the members and of the Board of Directors and shall attend to the giving of all notices to members and directors. The Secretary shall have charge of the record of members of the Association and of such other books and papers as the Board may direct and shall have all such powers and duties as generally are incident to the position of Secretary or as may be assigned by the President or the Board.

Section 6. Powers and Duties of the Treasurer.

The Treasurer shall have charge of all funds and securities of the Association, shall endorse the same for deposit or collection when necessary and deposit the same to the credit of the Association in such banks or depositories as the Board of Directors may authorize. The Treasurer may endorse all commercial documents requiring endorsements for or on behalf of the Association and may sign all receipts and vouchers for payments made to the Association and shall have all such powers and duties as generally are incident to the position of Treasurer or as may be assigned by the President or by the Board.

Section 7. Powers and Duties of Assistant Secretaries.

In the absence or inability of the Secretary to act, any Assistant Secretary may perform all the duties and exercise all the powers of the Secretary. The performance of any such duty shall be conclusive evidence of such Assistant Secretary's power to act. An Assistant Secretary shall also perform such other duties as the Secretary or the Board of Directors may assign.

Section 8. Powers and Duties of Assistant Treasurers.

In the absence or inability of the Treasurer to act, an Assistant Treasurer may perform all the duties and exercise all the powers of the Treasurer. The performance of any such duty shall be conclusive evidence of such Assistant Treasurer's power to act. An Assistant Treasurer shall also perform such other duties as the Treasurer or the Board of Directors may assign. Any such Assistant Treasurer may be an individual, a bank or a trust company, as the Board of Directors may determine.

ARTICLE V

Indemnification

Any person made, or threatened to be made, a party to any action or proceeding, whether civil or criminal, by reason of the fact that such person or such person's testator or intestate, is or was a director or officer of the Association or serves or served any other corporation, partnership, joint venture, trust, employee benefit plan or other enterprise in any capacity at the request of the Association, shall be indemnified by the Association, and the Association may advance such person's related expenses, to the full extent permitted by the law. The Association shall have the power to purchase and maintain insurance to indemnify the Association and its directors and officers to the full extent such indemnification is permitted by law.

ARTICLE VI

Checks, Notes, Drafts, etc.

Checks, notes, drafts, acceptances, bills of exchange and other orders or obligations for the payment of money shall be signed by such officer or officers or person or persons as the Board of Directors shall from time to time determine.

ARTICLE VII

Amendments

These By-Laws may be amended or repealed, and new By-Laws may be adopted, (1) by the members at the time entitled to vote in the election of any directors, at any annual meeting of the members or at any special meeting of the members called for that purpose, or (2) by the Board of Directors. Any By-Law adopted by the Board may be amended or repealed by the members entitled to vote thereon as herein provided, but a By-Law adopted by the members may provide that such By-Law shall not be subject to amendment or repeal by the Board. If any By-Law regarding an impending election of directors is adopted, amended or repealed by the Board, there shall be set forth in the notice of the next meeting of members for the election of directors the By-Law so adopted, amended or repealed, together with a concise statement of the changes made.

ARTICLE VIII

Fiscal Year

The fiscal year of the Association shall be the calendar year

APPLICATION FOR MEMBERSHIP

USENIX ASSOCIATION

Institutional Membership

Check One:

- Educational License (\$100)
- Bell System Installation (\$300)
- Other License (\$300)

Licensed for:

- Version 6 Version 7 PWB/UNIX Mini-UNIX UNIX/32

- Check One: Source License Binary Only

Newsletter Mailing Address:

Software Distribution Label Address:

Delegate Address:

Name: _____

Amount Enclosed: \$ _____

Authorized Signature: _____ Date: _____

Return completed form to: Usenix Association, Box 8,
The Rockefeller University, 1230 York Avenue, New York, NY 10021.

(For Individual or Public Membership see other side.)

APPLICATION FOR MEMBERSHIP
USENIX ASSOCIATION
Individual or Public Membership

Individual (\$12)

Institutional Affiliation: _____

Nature of Affiliation: _____

Public (\$12)

Mailing Address (Individual Members must use institution address):

Name: _____

Overseas airmail, add \$5.00

Invoice required, add \$3.00 bookkeeping charge for invoice or receipt

Receipt required

Amount Enclosed: \$ _____

Signature: _____ Date: _____

Return completed form to: Usenix Association, Box 8,

The Rockefeller University, 1230 York Avenue, New York, NY 10021.

(For Institutional Membership see other side.)

SOFTWARE

System Software

Principal Operating System:

UNIX: Version 7 UNIX/32 PWB/UNIX Version 6 Mini

Language Processors:

C: Version 7 Version 6 PWB Phototypesetter
 Own Version _____

Fortran: Bell Princeton CULC 4+ 77 _____

Pascal: Berkeley Amsterdam _____

Basic: Basic Basic + Basic + 2

Other: SNOBOL SPITBOL SNOBOL4 COBOL LISP
 Modula BCPL Ada Algol 68 Simula
 Euclid _____ _____ _____ _____

Network Support

ARPANET DECNET Telenet X.25 IBM SNA _____

Non-Standard Device Drivers

Device: _____ Status: _____

Availability: _____

Other Software

Description: _____

Availability: _____

Return completed form to: Peter Ivanov, Dept of Computer Science,
University of New South Wales,
P.O. Box 1 Kensington 2033.
AUSTRALIA.

AUUG Software Catalogue

Our installation would be prepared to pay \$
towards compilation of the catalogue.

We would like to pay as:

A subscription to the catalogue (eg pay for a tape)

A contribution to the compilation of the catalogue

If your financial system could better handle some other method of payment
please indicate below:

Installation name:

Address:

Please return as soon as possible to:

Peter Ivanov,
Dept. of Computer Science,
University of N.S.W.
P.O. Box 1
Kensington 2033
Australia.

