AUSTRALIAN UNIX USERS GROUP NEWSLETTER

CONTENTS

Little Fishies

All issues of AUUGN so far produced are available on micro-fiche. As far as backissues of volumes one and two, they are now only available on micro-fiche, except for five issues of volume 2 number 6. In general, as bond paper copies run out, little fish will be the only way to obtain AUUGNs.

Should you have a preference for receiving your AUUGNs in this reduced form, just drop me a line, and future issues will be supplied on fiche only. Micro-fiche issues will be mailed about a week after the normal mailing depending on production time.

In this issue

As you can see from the size of this issue, there is a lot in it. As promised there is a summary of the last US meeting, thanks to Ian Johnstone, Andrew Hume and Piers Lauder who served as unpaid scribes. Several interesting papers are also included.

I have held back a Troll user manual (see speaker 38 in the US summary), which is really a collection of manual entries for the Troll system. Also held back is the 43 page, double reduced and double sided, list of registrants for the San Francisco meeting. Quite a lot of people really. I plan to publish it in an issue with some spare space, but you may obtain a copy from me before that.

Those of you interested in obtaining a MODULA compiler will be pleased to see that it is available locally. To keep your brains in trim there is a little puzzle included. This is not new but as you can see it is to be used in our first year computing course and it stirred up so much interest among staff that I have included it here.

Newsletter Exchanges

I have entered into an exchange agreement with the US newsletter editor, yes it still exists (see letter this issue), and it would appear that Waldo Wedel is really taking the job seriously. I have the last four issues he has produced and hope to include items from them in the next issue.

The Next Meeting

The next AUUG meeting will be held at the Australian Graduate School of Management at the University of New South Wales on March 16, 1981.

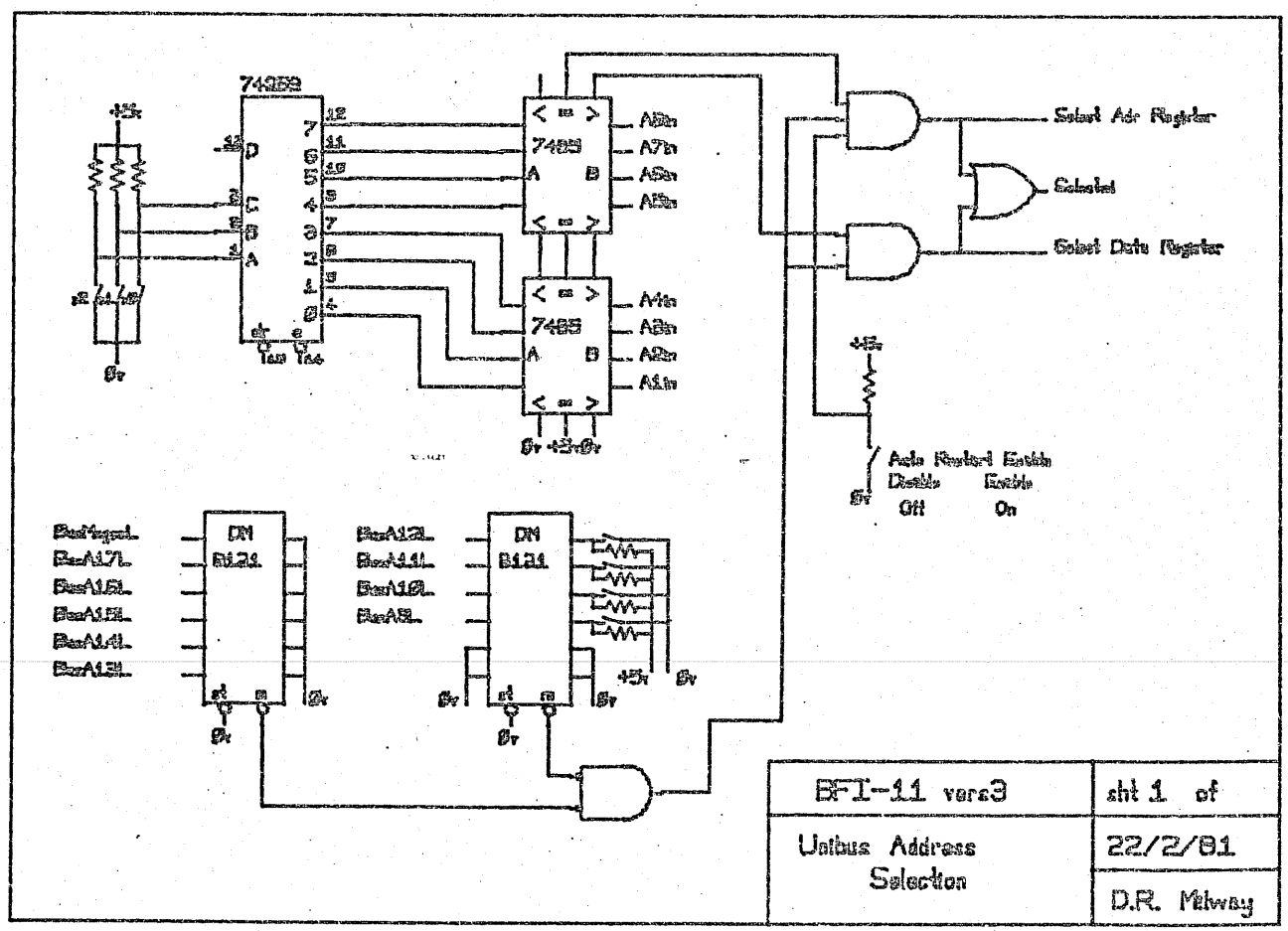
This meeting is being timed to coincide with a visit home by Ian Johnstone, presently working at Bell Laboratories. Ian will give a brief presentation and answer queries from the floor on what is happening with UNIX at Bell and elsewhere in the U.S.A.

Details are being mailed separately to all subscribers, but any enquiries should be directed to Lindsay Harris, AGSM, UNSW, PO Box 1, Kensington 2033, AUSTRALIA.

Art, is art, is art, is....

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

(02) 662-3781



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

SCHOOL OF ELECTRICAL ENGINEERING

PUZZLE OF THE WEEK

I plan to give this problem to my Computing 1 students, but I thought it might be interesting to see how long it takes the 'experts' to solve it.

First Prize - the satisfaction of finding the solution.

****

The are two integers each between 1 and 100 (non-inclusive). P knows their product; S knows their sum. Obviously, if they told each other the sum and product, they could figure out what the integers were. Instead, they have the following conversation:

P : "I don't know what the numbers are"

Q : "I knew you didn't and neither do I"

P : "Oh! Now I know"

Q : "Oh! So do I"

What are the numbers?

P.S. You don't need a computer for this one.

Jeffrey Tobias

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

SCHOOL OF ELECTRICAL ENGINEERING

MODULA-2 UNDER UNIX

We are pleased to announce the availability of Professor Niklaus Wirth's new language, Modula-2, operating under the UNIX timesharing system, on PDP-11 computers. The following pages give details of the various Modula-2 compilers that are to be made available.

Compiler M2UNIX11 is an adaption of Professor Wirth's original RT-11 based compiler, and will execute under both Level 6 and Level 7 UNIX on PDP11 computer systems. The compiler produces PDP-11 code directly, with no assembly or interpretive phase. The linker enables the combination of separately compiled modules with rigorous type checking being employed. The entire system is being distributed in both object and source form.

A project is currently underway to implement the Modula-2 system under VAX/UNIX. No expected time scale can as yet be given for this project, and a further announcement will be made when it becomes available.

The following sheets are self-explanatory. To receive the UNIX version of Modula-2, simply complete a copy of the licence agreement, and send it, with Aust.$150, to

Jeffrey Tobias,
Department of Computer Science,
Univeristy of New South Wales,
P.O. Box 1,
Kensington, N.S.W. 2033,
AUSTRALIA.

Institut für Informatik
ETH Zürich
CH-8092 Zürich

January 1981

Release of Modula-2 compilers

The design of the programming language Modula-2, started in 1977, was followed by implementation efforts of various compilers. A version designed for the PDP-11 computer and its RT-11 operating system was released in the summer of 1980. We are now happy to announce the releases of the following compilers on April 1, 1981. They will be distributed under a licensing agreement with the purpose of protecting the language from arbitrary changes and extensions.

1. Compiler M2RT11, operating on the PDP-11 (LSI-11) under RT-11, generating PDP-11 code. The compiler is written in Modula-2 itself.

2. Compiler M2UNIX11, as in 1., but operating under UNIX.

3. Compiler SMILER-11, generating code for the PDP-11. This compiler is written in Pascal and operates on the CDC-Cyber computer. It can be compiled using Release 3 of the Pascal 6000 compiler. Adaptation to other Pascal compilers should be relatively easy.

4. Compiler SMILER-6809, generating code for the M6809 microprocessor, written in Pascal (see also 3.).

5. Compiler SMILER-68000, generating code for the M68000 microprocessor, written in Pascal (see also 3., release date: June 1.1981).

6. Compiler M2M, written in Modula-2, generating M-Code. M-Code is defined by an interpreter described in Modula-2. This compiler is primarily used to transport Modula-2 onto other computers by either
   1. coding the M-Code interpreter in machine language of the target computer, yielding an interpretive implementation, or
   2. replacing the compiler's pass 4 by a program directly generating code for the target machine.

   Note: a translation of this compiler into Pascal is planned, but no release date can be given at this time.

Compilers 1 and 6 were designed at the Institut für Informatik of ETH, compilers 3-5 were translated into Pascal at the Computation Center of ETH, and compiler 2 is an adaptation of 1. made at the Australian Atomic Energy Commission. Compilers 3-5 will be distributed in source form only.

The fee for each compiler is Sfr. 350 (A$ 150 for M2UNIX11). Our intention in distributing compilers for Modula-2 is to provide a modern tool for programming and thereby to advance the state of software engineering. The above fee must therefore not be regarded as a price for the compiler, but rather as a handling charge and coverage of documentation, tape, package and postage.

If you wish to receive a compiler, please fill out and sign the enclosed license agreement and return it to

Institut für Informatik
ETH-Zentrum
CH-8092 Zürich

(for compiler 1, 3-6) or to

Department of Computer Science
attn. Dr. J. Tobias
University of New South Wales
P.O. Box 1
Kensington, N.S.W. 2033
Australia

(for compiler 2).

Institut für Informatik
ETH Zürich
CH-8092 Zürich

January, 1981

License agreement

concerning use and distribution of the program subsequently called the compiler for the programming language Modula-2 between

.....

.....

subsequently called the licensee, and the Swiss Federal Institute of Technology in Zürich (Eidgenössische Technische Hochschule), subsequently called ETH, the licensor.

1. The license

1.1 ETH grants the licensee the right to use the compiler for its employees, customers, or students. He has no right to sell or give the compiler to third parties.

1.2 The licensee obtains the right to use the compiler for any of its requirements (education, production, demonstration) at no cost, except for the initial handling charge.

1.3 The licensee agrees not to charge any "software cost" for the use of the compiler to its paying customers.

1.4 The licensee does not obtain the right to sell the compiler, nor its modified versions, for commercial profit, without further agreement with ETH.

1.5 ETH has no obligation to "maintain" the compiler, i.e. to adapt it to new operating systems, nor to extend the language Modula-2. However, it will inform the licensee about discovered errors and their correction.

2. Documentation

2.1 ETH will make available its documentation at production cost, and retain the copyright for them.

2.2 ETH grants the licensee the right to produce its own manuals, handbooks, or other documentation under the provision that ETH'S authorship of the language and its compilers is explicitly mentioned. In particular, it must be mentioned that Modula-2 and the compiler were designed and developed at the Institut für Informatik, ETH Zürich.

3. Rights and authorship

3.1 The licensee has the right to adapt the compiler to its own computing environment. He agrees, however, not to alter the compiler in such a way that the language accepted by the compiler is changed. The reference document is the report "Modula-2" by N. Wirth of ETH.

3.2 Extensions of the language are not considered as changes. However, the licensee agrees to report planned extensions to ETH; he is discouraged from making substantial extensions in the interest of program portability.

3.3 If the licensee derives another compiler from the one subject to this license, he agrees to report its availability and, if requested, to make it available to the licensor under the same conditions as stated in this license.

Location:

Date:

The Licensee:

.....

The Licensor:

.....

UNIX on the LSI-11/23

Dave Horsfall
Computing Services Unit
University of NSW

This article describes the author's experiences in implementing UNIX on a LSI-11/23. Readers may remember my previous article discussing UNIX implementation on a PDP-11/34 (published in the Vol I No 6 issue of AUUGN).

The implementation in question is that belonging to the School of Food Technology consisting of a Pertec CD20 system, which is actually an LSI-11/23 in disguise mounted inside a 20 Mb disk assembly. Two buttons stick out of the cabinet: the line time clock (LTC) which is turned on and off (more about this later) and the RUN button. Everything else is handled through ODT which is actually implemented in the micro-code. The LSI-11/23 is basically an LSI-11 that grew up. It has all the nice features that UNIX wants, such as memory management, optional FPU, 4 hardware priority levels etc.

The biggest problem once again was the lack of a switch register. Not only is there no physical device, it does not even respond to the bus address. The bootstrap therefore had to be cobbled up to always prompt for the path name. This is actually an assembly option. In addition, a pseudo switch register is maintained at memory location 50 (octal) initially set to 173030 for the benefit of /etc/init; the symbol SWR addresses this instead; and a program swr written to read/set this word. All this is included by defining the symbol NO SWR at compile time.

When booting therefore, one responds to the ODT prompt with say 773030G and replying to the `?' prompt with unix. This gives you a single user system. To come up multi-user, change the pseudo switch register to say 1 and log off. No modification to /etc/init is therefore necessary.

The line time clock (LTC) normally runs (and interrupts !) all the time and must be disabled by the push button. It is recommended that the internal jumper is used instead to convert this to a clock at address 777546. Without this, the LTC must be disabled for the bootstrap to work and enabled again for UNIX.

This installation also has an interesting array of terminals. Aside from the usual DLV11-J, they have 2 DLV11-F's (single line programmable speed) and a DLV11-E (like the DL11-E but also programmable). To support these, the KL driver (kl.c) was modified to set up the baud rates in the hardware, and also to recognize the concept of single-open terminals for their LA120 and Sanders. The original driver was sadly lacking in these respects. It also had to be modified for the XON/XOFF facility I recently added to the (level 6) TTY driver. This facility allows the LA120 and Sanders to operate at 2400 baud, and is extremely useful for humans, too.

Any attempt to change the baud rate on the DVL11-J (indeed any terminal interface not supporting programmable speeds) is quietly ignored by the driver. It actually enforces the hard-wired speed (specified on a per-line basis at compile time) so that a gtty following a stty to change the speeds will discover that it didn't work. Of course, the stty command does this for the user. Incidentally, the DLV11-E driver does not support dial-in lines properly yet, but I'm working on it. By the time this appears in print it should be running.

SOFTWARE TOOLS USERS GROUP

A G E N D A

January 20, 1981
San Francisco, CA

Debbie Scherrer, Lawrence Berkeley Laboratory
Chairman

9:00 AM INTRODUCTION

* The Next Generation of Software Tools
  Anthony I. Wasserman, UC San Francisco and UC Berkeley

SESSION 1 - The Virtual Operating System Approach
Dennis Hall, Lawrence Berkeley Laboratory, Chair

* Virtual Operating Systems
  Dennis Hall, Lawrence Berkeley Laboratory

* Virtual Aethers
  Joseph Sventek, Lawrence Berkeley Laboratory

* Portability Layers of Graphics Software
  Mike O'Dell, Lawrence Berkeley Laboratory

* IEEE Micro Operating Systems Interface Standards
  Sam Kirk, TRW/VIDAR

10:30 BREAK

11:00 SESSION 2 - NEW AND ENHANCED UTILITIES
David Martin, Hughes Aircraft, Chair

* Toward a More Interactive Shell
  David Martin, Hughes Aircraft

* A Virtual Terminal Handler
  Allen Akin, Georgia Institute of Technology

* Product Development and the Software Tools
  Edward G. Happ, Interactive Data Corporation

12:30 LUNCH

1:30 SESSION 3 - IMPLEMENTATION ISSUES
Joe Sventek, Lawrence Berkeley Laboratory, Chair

Issues and Experiences in the Implementation of the Software Tools Primitive Functions

A series of short presentations on the implementations of the Software Tools Primitive Functions on various operating systems will be given. Particular emphasis will be given to those assumptions of the primitive functions which impeded the implementation efforts. Systems to be discussed are:

* LTSS, CTSS on CDC7600, CRAY-1
  Margaret Hug, Los Alamos Scientific Laboratory
* MAX4 on Modcomp IV
  Bob Upshaw, Lawrence Berkeley Laboratory
* CP/M on Z80
  Philip Scherrer, Unicorn Systems
* MPX on SEL
  Walt Donovan, NASA/Ames Research Center
* RSX-11M on PDP11
  Joe Sventek, Lawrence Berkeley Laboratory
* VMS on VAX-11
  Joe Sventek, Lawrence Berkeley Laboratory

Both general and specific questions will be fielded from the floor after the completion of the presentations.

3:00 BREAK

3:30 SESSION 4 - FUTURE DIRECTIONS - Panel Discussion

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

SESSION 5 - SPECIAL INTEREST GROUP MEETINGS

Ratfor, Network, Primitives, and Text Formatting & Editing SIGS
Plus meetings of people interested in implementing the tools on various machines and operating systems, including DEC, IBM, CDC, DataGeneral, Hewlett-Packard, Modcomp, Xerox, Honeywell, Univac, Micros, and others

Reports will be given at the end of the session on the activities of the various groups

USENIX Winter 1981 Conference

Jack Tar Hotel, San Francisco

Wednesday, January 21st, 1981

through

Friday, January 23rd, 1981

Program Agenda

Conference Chairman:

Thomas Ferrin
University of California, San Francisco

Technical Program:

Michael O'Dell
Lawrence Berkeley Laboratory

William Joy
University of California, Berkeley

Conference Coordinator:

Joan Apodaca
University of California, San Francisco

Wednesday, January 21, 1981			
0830-1000			On site registration in Lobby
0930-1000			*Coffee*
	Introduction		**Thomas Ferrin, UCSF**
1000-1015	T. Ferrin	UCSF	Opening remarks and announcements
1015-1115	D. Ritchie	BTL	Evolution of the UNIX Timesharing System
1115-1130	L. Isley	Western	What's happening at BTL/Western Electric
1130-1145	W. Munson	DEC	What's happening at DEC
1145-1200	R. Fabry	UCB	DARPA VAX/UNIX support effort
			Lunch
1300-1330	L. Katz	USENIX	Business Meeting (Part 1)
	System Portability		**Chair: Sam Leffer, Sytek, Inc.**
1330-1350	A. Nemeth	BBN	C/70 Micro-machine Hardware Overview
1350-1410	C. Howe	BBN	C/70 Macrocode Architecture
1410-1430	A. Nemeth	BBN	Porting UNIX to the C/70
1430-1500	T. Heines	CSU	Porting UNIX to the Series/1
			Coffee Break
	System Portability, cont.		**Chair: William Joy, UCB**
1530-1545	K. Wilson	Cornell	C on the FPS-164
1545-1555	D. Strict	U. of Pittsburg	A Truly Portable I/O Library
	Network Topics		**Chair: William Joy, UCB (continuing)**
1555-1615	P. Hardie	U. of Sask.	IBM Front End to Hasp Multileaving
1615-1630	K. Harrenstein	SRI	DEAFnet
1630-1700	L. Katz	USENIX	Business Meeting (Part 2)
			Break
1800-2000			**Reception in El-Dorado Room**
			Dinner

All Regular Sessions are in the International Room

Want to join USENIX?

Benefits include receiving monthly copies of *;login:*, the UNIX newsletter, software distribution tapes (institutional members only) and announcements of upcoming USENIX conferences. Membership application forms are available on the registration tables or you can write to:

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

† *UNIX is a trademark of Bell Laboratories*

Thursday, January 22, 1981			
	Kernel Topics		**Chair: Michael O'Dell, LBL**
0900-0915	W. Jolitz	USGS	USGS/UCB V7 System
0915-0930	K. Harrenstein	SRI	Compact and Simple Kernel Overlays
0930-0945	M. Tilson	HCR	V7 Conversion Tools
0945-1000	P. Staubach	U. of Oklahoma	Multi-controller disk driver
1000-1015	J. Reeds	UCB	Floating point save problems
1015-1030	A. Romberger	UCB	Real time I/O using LPA-11
			Coffee Break
1100-1115	S. Leffler	Sytek, Inc.	Terminal Linking Line Discipline
1115-1130	M. Tilson	HCR	Low-Cost Terminal MUX
1130-1145	R. Broersma	NOSC	Multi-Host Terminal Front End
1145-1200	W. Joy	UCB	A Crash-resistant UNIX File System
			Lunch
	VAX		**Chair: Michael O'Brien, RAND**
1300-1320	D. Kashtan	SRI	Eunice: UNIX Emulation on VMS
1320-1340	W. Joy	UCB	Vax/UNIX Enhancements and Directions
1340-1400	R. Pike	BTL	11/750: Comet Haley or Kohoutek?
1400-1420	A. Stettner	DEC	VAX News From DEC
1420-1500	above + R. Kridle	Panel	Vax Roundtable
			Coffee Break
	Networking		**Chair: Bruce Borden, 3Com Corp.**
1530-1550	R. Gurwitz	BBN	VAX/UNIX ARPA-net Support Project
1550-1610	J. Mullen	Mitre	An IP/TCP Network Front End
1610-1630	S. Tepper	RAND	The RAND Network Front End
1630-1700	G. Popek	UCLA	Locus: The UCLA Distributed System
			Dinner
	Special Sessions		
1900-2200	Telegraph Hill Room		Vendor Hardware/Software Exposition
1900-2200	Twin Peaks Room		Interactive Systems Corporation
1900-2200	Marina&Seacliff Rooms		VAX Special Interest Group
1900-2200	Presidio&Pacific Hghts Rooms		Open Birds of a Feather (Small Groups)

The Bug Bucket

A bucket with a large bug attached is located by the registration tables. Everyone that has discovered a bug in any UNIX software is encourged to fill out a "bug report form" (available at the registration tables) detailing as much information as possible about the problem and put the completed form in the bug bucket. The results will be sorted, tabulated, and eventually published in the *;login:* newsletter.

Friday, January 23, 1981			
	Word Processing		**Chair: Eric Allman, UCB**
0900-0915	G. Aikens	Owl Associates	Device Independent Screen Editor
0915-0930	K. Auerbach	INTERACTIVE	Network Independent Message System
0930-1000	M. Kampe	INTERACTIVE	A New Text Formatting Package
1000-1030	M. Horton	UCB	Terminal Independent CRT Sofware
			Coffee Break
	Data Base Systems		**Chair: Roland Johnson, LBL**
1100-1120	A. Wasserman	UCSF	TROLL: A Compact Relational System
1120-1140	M. Meyer	UCB	A Database Application Design System
1140-1200	E. Allman	UCB	INGRES: Status and Directions
			Lunch
	Applications		**Chair: Peter Kreps, LBL**
1300-1320	R. Webster	USACE	Environmental Technical Info. System
1320-1340	S. Pozgaj	HCR	Unix as a Large Application Base
1340-1420	S. Bourne	BTL	The "Draw" Circuit Design System
1420-1435	R. Henry	UCB	Compiler Error Analysis for Fast Debugging
1435-1450	P. Kessler	UCB	A Pascal Compiler for the VAX
1450-1500	J. Thompson	U. of Oklahoma	Tcheck: a file system tree checker
			Coffee Break
	Applications, cont		**Chair: Michael Wahrman, RAND**
1530-1550	J. Joyce	ITS	UNIX Aides for English Courses
1550-1700			*Open Time & Overflow*

Conference proceedings and attendees list.

There are no current plans to publish and distribute official proceedings from this conference; however, we do plan to publish a set of notes taken during the conference in a future issue of *;login:*. If you want to be assured of getting a copy, make sure your USENIX membership is up to date. Also on Friday, a list will be distributed with the names and addresses of all conference registrants.

What follows is an attempt to summarise the proceedings of the Software Tools and USENIX meetings held at the Jack Tar Hotel, San Francisco, Tuesday 20th January through Friday 23rd January 1981.

The summary was compiled "from afar" using notes taken by the unpaid scribes whose names appear below (thanks fellas)

Ian Johnstone, Bell Laboratories, U.S.A.
Andrew Hume, Bell Laboratories, U.S.A. and
Piers Lauder, University of Sydney, Australia.

The summary does not purport to be true and correct in all details, merely the best that I can do given the quality of note taking (very good in parts but showing the strain in others) and the recollections of the only attendee to whom I have immediate access. Special thanks to Piers for his help.

Peter Ivanov
University of N.S.W.
Australia
16/2/81.

Software Tools User Group Meeting
Tuesday, January 20, 1981
Introduction
Chair: Debbie Scherrer, LBL

Speaker 1: 9:00am Anthony Wasserman, UCSF

The next generation of software tools

Have two goals in software. To improve the quality of software and the methods of generating it. Software tools reduce the rate of cost increase in software production.

Sees tools as means not ends. All tools have limits and as Hoare said 'A hammer is useful because it is not a screw driver'.

The next generation of software tools will be based on: generation of a 'knowledge base' of programmers use of tools, capturing information on program structure, design and construction in real time and utilizing sophisticated human interfaces.

It is important to support the entire software cycle of source, documentation, control files, test data and use.

Software Tools User Group Meeting
Tuesday, January 20, 1981
The Virtual Operating System Approach
Chair: Dennis Hall, LBL

Speaker 2: 9:40am Dennis Hall, LBL

Virtual operating systems

Presented areas of interest and correspondents notes. A SNOBOL to FORTRAN translator is available from D. Hansen, Uni of Texas.

Speaker 3: 10:10am Joseph Sventek, LBL

Virtual Aethers

Described a means for communication between processes. Essentially software implementation of ethernet. Primitives allow create and access of an aether, access and deaccess of aether, deposit and receive messages.

Speaker 4: 10:30am Mike O'Dell, LBL

Portability layers of graphics software

How to integrate graphics and software tools for business graphics, data presentation and analysis and high performance graphics.

The top layer of graphics interface consists of image composition or generation language. Middle layer would consist of utility routines with independence from lowest layer which could be an implementation of the SIGRAPH CORE.

Speaker 5: 10:50am Sam Kirk, TRW/VIDAR

IEEE micro operating systems interface standards

Discussed operating system interface standards and the IEEE committee on same. Portability across hardware especially micro-processors.

------- Coffee Break -------

Software Tools User Group Meeting
Tuesday, January 20, 1981
New And Enhanced Utilities
Chair: David Martin, Highes Aircraft

Speaker 6: 11:30am David Martin, Hughes Aircraft

Toward a more interactive shell

David spoke of a better shell with 'csh' features. Includes error correcting capabilities, feedback mechanisms such as in place file name expansions and some terminal handling capabilities. Under VMS.

Speaker 7: 12:00 Allen Akin, Georgia Inst. Tech.

A virtual terminal handler

Developed on a PRIME features full screen editing for all terminals. Similar to TERMCAP facility available in 4th Berkeley UNIX distribution.

Software Tools User Group Meeting
Tuesday, January 20, 1981
Implementation Issues
Chair: Joe Sventek, LBL

Speaker 8: 1:30pm Margaret Hug, Los Alamos Scientific Lab.

LTSS, CTSS on CDC7600, CRAY-1

Los Alamos users are very sophisticated but hate changes in their environment. There exist single programs of two or three hundred thousand lines which thus have lifetimes of 20-30 years. Problems encountered with the CDC7600 include the character set, word boundaries, line lengths, eol and eof conventions and other ghastly problems to do with the operating system.

Speaker 9: 2:00pm Bob Upshaw, LBL

MAX4 on Modcomp IV

A good quote

"I program all night, manage all day, and sleep during meetings!"

Here I quote direct from the notes: Spawn must return status, primitive to search paths necessary, non-standard FORTRANs a problem. Gave lots of statistics on implementation costs.

Speaker 10: Philip Scherrer, Unicorn Systems

CP/M on the Z80

Basic operating system for Z80 micros. Problems encountered; FIV compiler needs order dependent declarations, restricted index expressions, linker requires order dependant input modules.

Speaker 11: Walt Donovan, NASA/Ames Research Centre

MPX on SEL

Described a 32 bit fast CPU for aerospace research. Came with no software. Software tools implemented in 2 man months because UNIX was unavailable. Problems encountered include unsigned characters, fixed file sizes, line truncation at 72 characters and a FORTRAN compiler that could not handle lower case 'w'.

Speaker 12: Joe Sventek, LBL

RSX-11m on a PDP11 and VMS on a VAX

Separated directory manipulations from file manipulations because on all systems but UNIX they are different objects. Directories must be read sequentially. Support of 'raw' terminal I/O. Problems were record oriented file structure on disk, many file types, system calls with many variations, processes can not be created cheaply and difficult to support inter terminal I/O.

USENIX Meeting
Wednesday, January 21, 1981
Introduction
Chair: Thomas Ferrin, UCSF

Speaker 1: 10:00am T. Ferrin, UCSF

Opening Remarks and Announcements

Tom opened the meeting with organizational matters but mentioned that this was the largest meeting so far with over 800 attendees. A list of attendees is included in this summary.

Speaker 2: 10:15am D. Ritchie, Bell Labs

Evolution of the UNIX Timesharing System

Dennis spoke of the early days of UNIX, but unfortunately our correspondents could only summarize what he said by 'see Sydney talk', referring to the talks Dennis gave in Australia in 1979. Sorry all those who did not hear him then.

Speaker 3: 11:15am L. Isley, Western Electric

What's happening at BTL/Western Electric

Some figures on numbers of licenses and installations were mentioned although on the whole these numbers seem to be very much on the low side.

Type	Licensees	Installations
Commercial	176	287
Government	96	197
Educational	608	1524
Administrative	17	51
Totals	897	2059

What about new releases from Western? There is a new Western Electric organization to handle software licensing and this body may be more responsive to requests. Approval/disapproval may be quicker at least. There are presently about eight systems under consideration for distribution, though what they are is open to conjecture. Decisions should be made within two months and there is a simplified licensing scheme to come.

A mild threat was apparent in the announcement that Bell intends to start auditing licenses, i.e. a bunch of Bell heavies may descend on a site to have a look around.....

Speaker 4: 11:30am Bill Munson, DEC

What's happening at DEC

Bill gave an amusing talk describing the DEC UNIX group. Dec now has a tape which contains UNIX drivers for various bits of DEC hardware that Bell distributions did not support. DEC also will provide interested parties with a tape of DEC produced UNIX software for $1500 ($150 educational). Some DEC promotional literature now mentions UNIX!

Best quote was:

"BLISS is ignorance!"

Speaker 5: 11:45am Robert Fabry, UCB

DARPA VAX/UNIX support effort

The project is being undertaken on a VAX11/780 by Bob Fabry, Bill Joy, Laura Tong and ten graduate students. They have been given about $700,000 over 18 months.

Major efforts are towards fast access to big files, coupling files into address space, interprocess communications, networking (ARPA, ethernet) and performance enhancements.

Other efforts are; Fortran (speed execution), UUCP, automatic kernel configuration, directory structure, screen management, SDB, line disciplines, flexibility of kernel buffering and software control over networks.

For distribution:

Laura Tong
Computer System Research Group
Computer Science
UCB 97720

(415) 662 7780

------ LUNCH ------

Speaker 6: 1:00pm Lou Katz, USENIX

Business Meeting

The next US meeting will be 23-26th June at the "Joe C. Thompson Conference Center", University of Texas, Austin, Texas. Papers will be called for in February.

The January 1982 meeting will be in Santa Monica hosted by Interactive Systems Corporation. Future conferences will be held on the west coast in winter and the east in summer. The USENIX conference scheduler is

John Donnelly
P.O. Box 3000
NCAR
Boulder Colorado

Another useful address is the newsletter editor:

Newsletter Editor
Computation Center
University of Texas
Austin
TX 78712

[login@utexas-11]

dialin on (512) 474 5511, login name 'login', password USENIX

USENIX Meeting
Wednesday, January 21, 1981
System Portability
Chair: Sam Leffer, Sytek, Inc.

Speaker 7: 1:30pm Alan Nemeth, BBN

C/70 Micro-machine hardware overview

Goals of the project include support of existing applications, provision of a base for stand-alone systems, and a source of cheap computing power. Flexibility and simplicity of architecture, programming and construction were also important.

The result is claimed to be about as fast as a PDP11/70 for only $50,000. Other uses include emulation of obsolete equipment, a stand-alone 'C' machine, or a terminal controller. Fifteen to twenty of them exist now and it is planned to have about 75 operational by June.

Speaker 8: 1:50pm Carl Howe, BBN

C/70 Macrocode Architecture

Goals include an address space greater than 16 bits, efficient addressing of C data types, more register variables, fast and efficient subroutine linkage, direct access to machine from C and compact encoding of instructions.

The instruction set performs 10, 20 an 40 bit arithmetic operations, has an instruction length of from 1 to 4 words (20bit) with 19 addressing modes (including array element and structure element). Forty instructions use all 19 modes and 43 use others (stack frame manipulation).

The registers (all 1024 of them) are used as a cache and all macro instructions are restartable after an exception. C/70 code uses 4-5% less space than PDP11/70 code and some instruction sequences are notably faster, for example a function call, including register saves, takes only 6.7uS compared to 17.4uS on the PDP11/70.

Speaker 9: 2:10pm Alan Nemeth, BBN

Porting UNIX to the C/70

C/70 UNIX is straight level 7 with the addition of a network and screen editor. Kernel differences consist of memory management (1Kb clicks), context switching, traps and system calls, and coping with 10 bit bytes. Out of 36 kernel modules about 12 required substantial changes.

Some future directions will be to implement critical routines in micro-code, expand local network support and multi-processor UNIX.

Speaker 10: 2:30pm G. Popek, UCLA

LOCUS: The UCLA distributed system

Goals of this project included application code compatibility, network transparency, performance, maintenance of local autonomy, extensibility and reliability.

All files on the system belong to the one file system and each file is referred to by a global unique file system number/inode number pair. A physical mount table translates the global FS number to major/minor devices. A global mount table contains information about the whole network.

Performance of local actions are within 5% of normal level 7 speeds while a remote read runs at about 9mS/block elapsed time. Some directories and files are still local such as "/tmp". For more information contact:

B. Walker
UCLA Computer Science Department
3804 Boelter Hall
405 Hilgard Ave.
L.A. 90024

------ Coffee Break ------

USENIX Meeting
Wednesday, January 21, 1981
System Portability, Continued...
Chair: William Joy, UCB

Speaker 11: 3:30pm Kenneth Wilson, Cornell

C on the FPS AP-190L

Ken described the solution to those stubborn array problems, crystallographers please note. The array processor discussed, connected to an IBM370-168, has 96K 38bit words, cost about $130,000 and is about eight times faster than a 370-168 for the job it does. Only trouble is that it is programmed in FORTRAN or a micro-code assembler.

The Cornell people want to get C going on it and possibly interface it to a VAX.

Speaker 12: 3:45pm Daniel Strict, U. of Pittsburg

A truly portable I/O library

The new portable I/O library described has been totally rewritten and is not covered by a Bell license. It supports things like (and here I have only copied the notes) r/w via a single I/O stream, concurrent independent sequential access and file block caching in user file space.

Speaker 13: 4:00pm Peter Hardie, U. of Sask

IBM front end to HASP multileaving

Described a PDP11/70 linked to an IBM via a DQ11. Used PDP as a front end for the IBM. Tried to give his users an IBM-like environment as most real work was carried out on the IBM.

Speaker 14: 4:15pm Kenneth Harrenstein, SRI

DEAFnet

The SRI DEAFnet project is working towards a computer message system on a nation wide network for the deaf community. UNIX was chosen for use on the gateways because PDP-11 plus total software control made a good combination.

Changes to the software were made to cope with the naive user population. Deaf people's knowledge of English is less than normal, since it is really a second language. Many deaf people already have ttys for communication (5bit type) so special tty driver support was needed for the primitive character set and slow speeds. The BAUDOT 3-row keyboard presented special problems.

DEAFnet has 300 users in Washington DC and a long waiting list. Peter hopes in the future everyone will use electronic mail and so the deaf will benefit.

Speaker 15: 4:45pm David Tilbrook, BNSR

ANGUS prototyping system

ANGUS (ANother God-damned UNIX System) is UNIX in combination with TIPS and MASCOT. TIPS is a data-base package written by David and MASCOT was developed by the British armed forces and uses modular primitives for expressing software structure for multiprocessor systems. David also described MENU, an interactive enquiry system which interfaces users to systems such as data bases. MENU interprets TIPS format entries and uses simple, editable scripts.

USENIX Meeting
Thursday, January 22, 1981
Kernel Topics
Chair: Michael O'Dell, LBL

Speaker 16: 9:10am Kenneth Harrenstein, SRI

Compact and simple kernel overlays

Ken described another version of kernel overlays for squeezing large driver systems UNIX onto small PDP-11s. Provided overlay scheme similar to the UNSW big UNIX scheme, but much easier to use. Clean specification of what gets overlayed. Only text may be overlayed, not data.

Speaker 17: 9:20am Mike Tilson, HCR

V7 conversion tools

Mike described the work he did to fit V7 onto small PDP11s. Basically another kernel overlaying scheme with a simple specification but once again no data overlaying.

Speaker 18: 9:34am Bill Jolitz, USGS

USGS/UCB V7 system

Development directions include

- Improved filesystem
- Kernel optimization; more buffers, bigger clist, hashed inode table, hashed buffer table

- New features; load averages, text overlays for user processes, tenex style tty driver, line disciplines (multiple ttys)
- Administrative features; dynamic file quotas, special '/tmp' file protection, expanded accounting, performance monitoring, privileged accounts, limit number of processes by process group
- Recent features; sturdy file system, job control, ECC for bigger transfers, kernel overlays.

Speaker 19: 9:45am Peter Staubach, U. of Oklahoma

Multi-controller disc driver

Faced with a PDP11/70 and a user population of 2500 (40/50 people typically logged in) was having space problems so wrote a multi controller disc driver to combine various drivers into one table driven driver

Speaker 20: 10:00am Jim Reeds, UCB

Floating point save problems

Described the floating point problem below, to do with the asynchronous nature of the floating point processor:

```
while (1)
{
        flag=0;
        x=x/0.;
        if(flag == 0) printf("arrggg\n");
}

fpcatch()
{
        signal(SIGFPE, fpcatch);
        flag++;
}
```

The FPE in kernel mode is misscheduled. As FP error regs are read only they should be saved in the user structure so that multiple signals may be handled one at a time. The fix to 'psig' in the system involves forcing it to handle multiple signals one at a time.

Speaker 21: 10:15am Asa Romberger, UCB

Real time I/O using LPA-11

Discussed the LPA-11 peripheral controller as a solution to problems with real time I/O.

------ Coffee Break ------

Speaker 22: 11:00am Bill Shannon, Sytek, Inc.

Terminal linking line discipline

Described a method of connecting terminal ports, at the driver level, similar to that used locally. Possible to connect or disconnect one or more terminal ports. For local readers the method does not use a quit char, just disconnect.

Speaker 23: 11:15am Mike Tilson, HCR

Low cost terminal MUX

As we all know, high speed tty output is expensive on CPU time and DMA MUXes are usually expensive in dollars. The goals were to off-load some tty function from the kernel, flexibility and low cost. The result was a 6800 driven terminal concentrator connected by a single serial line to the CPU. The control protocol is byte oriented with flow control, error corrections, data compression but only 7bit data. In practice four 9600 baud terminals may be supported by one 9600 baud serial line, and the users don't notice. Still to go is the move of tty.c to the 6800, interconnection of machines and a "standard terminal" specification.

Speaker 24: 11:30am Ronald Broersma, NOSC

Multi-host terminal front end

In an environment with many possible hosts, many terminals and very many possible interconnections, terminals are connected to the various hosts via a 16 bit data bus. Each terminal has a Z80 with 8Kbytes and each host has a master interface capable of connection by RS232 or parallel DMA.

The result is fewer cables, less complication, less software overheads for the hosts (char translation, padding, flow control etc) and flexibility.

Speaker 25: 11:45am William Joy, UCB

A crash-resistant UNIX file system

Bill described measures in hand at UCB to improve the UNIX file system robustness. Concerned with the lack of error checking in the kernel and problems caused by delayed I/O. Aims to improve reliability, recoverability and performance.

Reliability can be enhanced by writing indexing data (inodes and indirect blocks) for files after the data, and removing the indexing before the data. Recoverability can be improved by an automatic repair program (mentioned 'fsck') and automatic reboot. Remaining problems are caused by the kernel attempting to continue after hard disc errors, spreading the cancer.

------ LUNCH ------

USENIX Meeting
Thursday, January 22, 1981
VAX
Chair: Michael O'Brien, RAND

Speaker 26: 1:00pm David Kashtan, SRI

Eunice: UNIX emulation on VMS

Eunice to achieve efficient emulation of UNIX system calls and minimization of source level modifications to UNIX programs. The emulation so far should make it possible to run UNIX programs without recompilation or relinking. The emulation of the file structure maps VMS logical files to UNIX path names. All signals bar SIGPIPE are available and programs can fork, although process creation is expensive so 'empty' processes are kept around to use.

Speaker 27: 11:45am William Joy, UCB

VAX/UNIX enhancements and directions

Bill spoke of five directions:

1. Virtual memory; want sharing and performance. Possible solution is a segment based approach, better access controls and paging per segment rather than per process.

2. The file system; looking for higher performance through locality and better algorithms. Explained the notion of many cylinder groups, reflecting the directory structure and containing groupings of inodes and data, rather than one global division of inodes and data per file system. Algorithms to handle cylinder groups needed.

3. IPC mechanisms; user level protocols, transport protocols.

4. Networks; make various technologies available; ethernet, arpanet, csnet, chaosnet and provide a coherent interface.

5. Introduce line disciplines to MPX (via ioctl calls) with better buffering.

Bill foreshadowed a Berkeley distribution update in March which will include VAX750 support, F77 fixed up, INGRES, bug fixes, multi MBA stuff and shell scripts executable by the system. Future distributions will not support PDP11s.

Speaker 28: 1:45pm Rob Pike,BTL

11/750: Comet Haley or Kohoutek?

Things in general about the 750; no massbus or FPA yet, UNIBUS much nicer to use, TU58 instead of floppy discs for the diagnostic device, available with RK07s (DMA takes too much bus time and the servo system is unstable, available

later with RM80), tape unit is a TS11 (the most difficult device to mount a tape on ever). In general 60% of a 780 for 40% of the cost.

The design uses gate arrays but has some problems; e.g. the probe instruction and micro code loops if the last instruction in a page causes a write protect violation (eg after a fork). The floating point unit is inaccurate and the 'movtuc' instruction does not work if the destination and source overlap. The power control is a bit flaky, signals ok before it is.

The UNIX system for the 750 is identical to the 780 system to the user. All changes, about eleven, were made in the system. The 750 also supports DH11s. On the whole 'a good personal computer'. Some times for a shell loop are:

	11/70	11/780	11/750	11/34	11/23
user	1.9	1.4	2.8	2.7	4.6
sys	16.3	2.9	6.7	25.0	38.0

Speaker 29: 2:00pm Armando Stettner, DEC

VAX news from DEC

For openers, DEC says that the DH11 is not supported on the 750.

The DEC UNIX group goal is to bring up new devices on a UNIX system to make sure they run OK. Should you have problems, mail them to:

Armando Stettner
Bill Munson
Continental Bvlde
MK 1-1/D29
Merrimac NH 03054

The panel discussion which followed this talk brought a suggestion that UNIX users organize to force DEC to offer better packages for UNIX (eg no RK07s). Users thought that the VAX was generally very reliable. Any problems contact the above people.

------ Coffee Break ------

USENIX Meeting
Thursday, January 22, 1981
Networking
Chair: Bruce Borden, 3Com Corp.

Speaker 30: 3:10pm Robert Gurwitz, BBN

VAX/UNIX ARPA-net support project

Some modifications to UNIX kernel have been introduced for efficient support of ARPA net TCP and IP. Otherwise user level code uses LARGE address space. Separate access to protocol levels is provided using standard I/O calls. Non-blocking I/O has been introduced. Network buffers are allocated using separate memory map.

Distribution is in early March and will be included in the Berkeley distributions.

Speaker 31: 3:30pm John Mullen, Mitre

An IP/TCP network front end

This system, which "makes network access similar to disc access" uses a Zilog Z8000 based network front end to isolate hosts from the "protocol explosion". Cost of the interface is about $9,000 for 200Kbps data rates.

Speaker 32: 4:00pm Mike London, RAND

The RAND network front end

"NCP is the way of the past consistent with the
desire to solve yesterdays problems tomorrow."

Given that NCP on the VAX works now, design goals were to offload NCP from the host to the NFE (Network Front End) and allow multiple hosts to share one ARPA host protocol. As a side effect you get a free star net between hosts sharing and NFE.

The implementation is in C for a PDP11/34 NFE talking to a VAX. Interface by UNIX I/O calls, but everything interesting is done by 'ioctl' calls. DMC11s don't work on VAXs.

Speaker 33: 4:10pm P. J. Jalics, CSU

Porting UNIX to the Series/1

Why port UNIX to the IBM Series/1? To verify portability and for that best of all possible reasons, to make use of a useless machine. The machine characteristics are 16bit, byte addressable 256Kbytes, segmentation (sep I and D), reasonable devices, price about the same as a PDP11/34 and available with

two operating systems, both disasters.

Preparatory actions were; Get John Lions book, get xref of entire kernel, get lucky with finding a C compiler almost written for the Series/1, throw away IBM software, buy a PDP11 and install UNIX on it, devise S1 bootstrap and file system transfer.

Some two years and much experience later the system works. About thirty percent of code changed, with about 66% of the whole effort going into the C compiler.

USENIX Meeting
Friday, January 23, 1981
Word Processing
Chair: Eric Allman, UCB

Speaker 34: 9:00am Gary Aikens, Owl Associates

Device independent screen editor

The system described is similar to the TERMCAP system described later.

Speaker 35: 9:15am Karl Auerbach, INTERACTIVE

Network independent message system

A new electronic mail system provides network operation using a variety of telecommunications media. It may be configured for personal preferences, supplemented by local software and is easily ported. Features include asynchronous delivery, aliases for recipients, automatic redistributions, delivery confirmation, urgency, mailing lists etc.

Speaker 36: 9:30am Mark Kampe, INTERACTIVE

A new text formatting package

Motivation for development of a new formatting package include device independence, a broader spectrum of formatting needs and wishing to escape from fundamental nroff/troff. Obvious troff problems are performance, maintainability and extensibility, complexity of major macro packages and the difficulty of mastering the input language.

Described an intermediate language between the text formatter and output device. Developed a new formatter FF with simple input, built in 'mm' features and very fast. The formatting system supports a variety of output devices.

Speaker 37: 10:15am Mark Horton, UCB

Terminal independent CRT software

Mark described the TERMCAP facility, now able to understand more than 150 terminals. Information stored in the data-base includes size of screen, capabilities that exist and how to drive them, cursor addressing, padding (time delays), login initialization and much more. The 'termlib' library interface to TERMCAP data base has been extensively improved.

The CURSES screen optimization library is a high level interface optimally handling cursor movement and screen updates, multiple windows and stand-out mode. The system is available on the fourth Berkeley distribution.

------ Coffee Break ------

USENIX Meeting
Friday, January 23, 1981
Data Base Systems
Chair: Roland Johnson, LBL

Speaker 38: 11:00 Tony Wasserman, UCSF

TROLL: A compact relational system

TROLL is an interface language to the PLAIN data base handler, provides a language for testing and tuning the data base handler and provides a "poor mans" query language. PLAIN extends a PASCAL like language to data management providing a modular, compact vehicle for research into DBMS implementations.

There is or will be a paper on the system in Soft. P. and E. Distribution of version 3 will be in March, and you should write to Tony to get on the mailing list.

Speaker 39: 11:30am Marc Meyer, UCB

A database application design system

The APPLIED package provides simple and consistent menus, forms and implicit (designer specified) protection of users, reducing the task of coding applications for naive users. Tools used include MCOMM a menu oriented command interpreter, DBROWSE for data transformation by data base editor, INGRES for data storage and FRMT for data formatting (uses nroff, tbl etc).

Speaker 40: 11:40 Eric Allman, UCB

INGRES: Status and directions

Ingres 6.3 runs on large PDP11s with floating point under V7 UNIX. It was released in the second Berkeley distribution or may be obtained directly from Eric Allman for $150.

Ingres 7 runs on a VAX under the fourth Berkeley distribution. Is two-and-a-half times faster than the PDP11/70 version and will be released in March. Both versions of Ingres are in the public domain.

Documentation includes "intro" $5, ref. manual, tutorial, design etc $30. Address to contact:

Tandy Warnow (docco and distribution)
Eric Allman (technical)

Project INGRES
Electronics Research Lab.
Cory Hall
University of California 94720

(415) 642 2344

------ LUNCH ------

USENIX Meeting
Friday, January 23, 1981
Applications
Chair: Peter Kreps, LBL

Speaker 41: 1:00pm Calvin Corbin, USACE

Environmental technical information system

Calvin described a set of tools to access economic/environmental effects of large scale decisions (disasters!?!).

Speaker 42: 1:30pm S. Bourne, BTL

The DRAW circuit design system

The DRAW system, originally developed by Sandy Frazer and Joe Condon, allows interactive design of circuits using global concepts such as bussing and macro expansion. Static analysis program PLH analyses the resulting circuit. PLACE lays out the circuit elements while WIRE produces input to semi or automatic wire rap machines.

Released for use at MIT, other educational institutions may follow.

Speaker 43: 1:55pm Mike Tilson, HCR

UNIX as a large application base

Hardware: 3 PDP11/55, 8 DL11, 2 RP06, 2 RP05 and many VT100s. The CALRS system (a customer telephone fault reporting system) supports 60 users, a data base of more than 200Mbytes and must be fast reliable and maintainable.

Under RSX the package took 6 years to develop and the annual software maintenance cost $1M. Under UNIX the code size decreased by a factor of three, speed increased by a factor of three, the system took about 1.5 years to develop and takes about half a person to maintain.

Speaker 44: 2:20pm Robert Henry, UCB

Compiler error analysis for fast debugging

A method for condensing diagnostics into something manageable. Solution used is to insert error diagnostics for some types of error into the source as comments. Can handle most languages like this.

Speaker 45: 2:35pm Peter Kessler, UCB

A Pascal compiler for the VAX

This compiler will be released in the fourth Berkeley distribution. It compiles exactly the same language as 'pi'. Based on F77 methodology with one extra pass. Uses F77 pass one and C optimizer. Runtime library written in C. Is compatible with ADB, SDB and PROF. Interfaces to other languages (C, F77, LISP). Extensions to the language include non-scalar functions, record comparisons, reading/writing enumerators and separate compilation of modules.

Speaker 46: 2:50pm John Thompson, U. of Oklahoma

TCHECK: A file system tree checker

John described a new file system checker using K-formula grammars. Uses a push-down automaton machine to parse the file system and generate exception conditions. In theory this method could be very useful in file system error correction. Write to John at

Computer Science Dept.
Uni of Oklahoma
Norman, OK 73019

(405) 325 4721

------ Coffee Break ------

USENIX Meeting
Friday, January 23, 1981
Applications, continued.....
Chair: Michael Wahrman, RAND

Unfortunately our scribes had to leave the meeting at this time. Three further speakers (see below) were heard followed by another USENIX business meeting.

Speaker 47: 3:30pm J. Joyce, ITS

UNIX aides for English Courses

Speaker 48: 3:30pm Lyn Watl, USAF

Nroff

Speaker 49: 4:05pm John Bass

Porting UNIX to small machines

Speaker 50: 4:20pm

USENIX Business

THE WOLLONGONG GROUP

960 San Antonio Road △ Palo Alto, CA 94303 △ (415) 493-3698

Standard UNIX* Version 7 (with SCCS) Now Available for
PERKIN-ELMER 32-bit Computers (11/45 to VAX Class Systems)
and
Wollongong's SERIES 1 Computer (Powerful, Low-Cost System)

PERKIN-ELMER/UNIX

The Wollongong Group's exclusive **EDITION VII*** and **EDITION VII WORKBENCH** for PERKIN-ELMER Computers provides availability, performance, reliability, and price advantages for anyone with a UNIX system requirement. **EDITION VII WORKBENCH** combines the key feature of PROGRAMMERS WORKBENCH (Source Code Control System) with UNIX Version 7 for the most powerful 32-bit UNIX offering available.

SERIES 1/UNIX

Our SERIES 1 computer is a powerful low cost system that supports up to eight users with **EDITION VII** or **EDITION VII WORKBENCH.** Full system prices begin under $30,000. With the SERIES 1 you don't have to sacrifice performance and capability to meet budget constraints.

UNIX TRAINING AND SUPPORT

The Wollongong Group provides a full line of training and support for UNIX. Standard training classes are offered at our training facilities in Palo Alto. Custom-tailored and on-sight training classes are also available.

For your convenience, we have attached training registration and information request forms. Do not hesitate to call us now to arrange a demonstration, request technical and pricing information, or to ask us any questions you may have regarding our company and products.

*UNIX is a trademark of Bell Laboratories.
EDITION VII is a trademark of The Wollongong Group.

THE WOLLONGONG GROUP

960 San Antonio Road △ Palo Alto, CA 94303 △ (415) 493-3698

December 9, 1980

Dear UNIX User:

A three day INTRODUCTION TO UNIX* course will be held February 10th thru 12th, 1981 at The Wollongong Group's Palo Alto training facility.

This three day program features on-line lab sessions to compliment classroom presentations. Participants will learn to use the standard facilities of UNIX, seventh edition. In addition, you will gain the background necessary to read, understand, and use UNIX utilities to be covered in other courses. We have designed the course to span all major sections of UNIX from logging-on to the operation of UNIX in a software development environment.

Additional courses in the Wollongong series are:

- UNIX for Advanced Users. A continuation of the introductory course. The advanced course will provide an in-depth knowledge of the UNIX operating system, including
- "C" Programming Language. As a result of this course, participants will be able to write, compile and execute "C" programs and take advantage of unique strengths of the "C" programming language.
- Text Formatting Under UNIX. This course focuses on advanced processing capabilities of the UNIX operating system, and covers utilization of the editor and NROFF, in a word processing and text processing environment.

- Software Development Using UNIX and SCCS. Extensive coverage will be given to the Source Code Control System, plus all other UNIX facilities which contribute to its use in facilitating software development.
- UNIX Operating for System Operations Personnel. Participants will learn to operate a UNIX installation, including system organization, backup, file system repair and reconfiguration procedures.

Tuition for the Introductory UNIX course is $400 per person for the three day program and includes computer time and all required course materials.

To register, contact Barbara Snarr at (415) 493-3698.

In addition to the UNIX training courses, The Wollongong Group offers 16 and 32 bit UNIX software and hardware systems. Please use the enclosed information request card or call us for any additional information you may require.

Sincerely,

Paul V. Cubbage

Paul V. Cubbage
Vice President

PVC:brs

*UNIX is trademark of Bell Laboratories.

Pete, Can you believe this? Now perhaps you can see why they aren't active in the Aust. User Group Jan

MISTRESS

A Relational Database Management System
For small to medium applications

MISTRESS

A NEW RELATIONAL DATABASE MANAGEMENT SYSTEM
DESIGNED BY THE CREATORS OF MRS

MISTRESS

READILY CUSTOMIZED, FULLY MAINTAINED,
EXTENSIVELY SUPPORTED

MISTRESS

WRITTEN AND IMPLEMENTED USING
FULL SOFTWARE ENGINEERING PRINCIPLES –
MODULAR DESIGN, TOP-DOWN PROGRAMMING,
TRANSPORTABLE CODE

MISTRESS

DOWNWARD-COMPATIBLE WITH THE MRS
QUERY LANGUAGE

GENERAL FEATURES:

- a superset of the English-like Query Language used in MRS
- a host-language interface via "C" callable procedures
- restrictions on the numbers of tuples and attributes removed
- an ongoing program of enhancement and development

- more data types:
  - several varieties of character strings
  - 3 kinds of integers: 8, 16, and 32-bit
  - 2 kinds of floating point: 32 and 64-bit
  - 3 formats of date data type
  - dollar data type for commercial users

APPLICATIONS PACKAGES:

A number of applications packages are currently under development. These include:

- report generator
- screen oriented forms interface
- transaction processing system

Other packages are being designed for future release. Custom applications and other services will be undertaken on request at negotiable rates.

MISTRESS is designed to run on most existing PDP-11 and LSI-11/23 configurations under the Unix™ operating system (V7, PWB or V6). In addition, it is targeted for other configurations and operating system environments.

MISTRESS is sold under both standard and educational licences. A standard licence includes binaries, documentation, and one year's support. Ongoing support is available for a yearly fee. An educational licence, at a special price for non-profit academic institutions, includes only binaries and documentation, with no support.

Interested in acquiring a MISTRESS? Write to:

RHODNIUS INC.,
Product Information,
P.O. Box 1, Station D,
Scarborough, Ontario
CANADA M1R 4Y7

For prompt service, telephone

John Kornatowski
(416) 922-1743

Unix is a registered trademark of Western Electric

New Product Release

January 13, 1980

NEW PRODUCT RELEASE

The Santa Cruz Operation announces an agreement with Tymshare to enter the UNIX* world with a major product release. This new product will be based on Tymshare's Dynasty System, a low cost DEC 11/23* package complete with floppy and Winchester disk storage. The product includes the DYNIX* Operating System, which is derived from UNIX seventh edition, and licensed under the authority of Western Electric.

Dynasty hardware is fully supported with on-site customer service provided by Tymshare Computer Maintenance.

A DYNIX support center has been established by the Santa Cruz Operation. Included with each system is a software warranty providing telephone consulting support for user problems. User and system administration training is also available.

The Santa Cruz Operation, a technical and management consulting firm, offers technical services in product and software development. A complete range of system support services for UNIX installations is available.

* UNIX is a registered trademark of Bell Laboratories
DEC 11/23 is a registered trademark of Digital Equipment Corp.
DYNIX is a registered trademark of The Santa Cruz Operation Inc.

500 chestnut street, santa cruz, california, 95060 (408) 425-7222

Amdahl ANNOUNCES

Universal Timesharing System

Release 1.0

Large Systems Facilities and Support for the Timesharing Community

Licensed under the UNIX* Timesharing Operating System (*Unix is a trademark of Bell Laboratories.)

September 18, 1980

Amdahl announces the availability of the Amdahl Universal Timesharing System (UTS), a new timesharing system for the Amdahl 470 customer. UTS is a modified form of the UNIX Timesharing Operating System Version 7.0 that has been widely used on a number of minicomputers for several years. UTS addresses the longstanding need of the large computer user for a powerful, user-oriented timesharing system that provides advanced functions combined with simplicity, while offering a growth path for the large timesharing user who has outgrown the minicomputer environment.

Amdahl UTS operates as a virtual machine in the user's VM/370 environment. It has been designed and implemented to allow the timesharing user to concentrate on solving the current problem or completing the current task without having to be concerned about the idiosyncracies of the timesharing tools. Amdahl UTS allows the user to focus on the objective rather than on the timesharing system.

Highlights

Amdahl UTS conforms to UNIX Timesharing Operating System Version 7.0 externals and adds support of full-screen sessions on IBM 3270-compatible terminals. As a result, it has a logically consistent user interface and simplicity of use, with many advanced features not found in the more traditional timesharing systems. The user of Amdahl UTS will find the following features especially helpful:

- Asynchronous processing that allows the user to execute multiple, parallel tasks from a single terminal, without keyboard lockup. The result is better use of time spent at the terminal.
- A single, simple I/O interface that allows the same program to communicate with a device, a user at a terminal, or another program. This means greater simplicity and flexibility to the developer as well as the user of the UTS facilities.
- An easily used, powerful text processor to aid the user in document preparation and memorandum formatting.
- Hierarchical file structure that provides flexibility in manipulating data, data structures, and devices.
- Source maintenance system that provides control and reporting of maintenance and version levels of source material such as programs, data, documents, etc.
- Simple yet comprehensive data editing facilities that provide the user with convenient, easy-to-use data manipulation, allowing the user to concentrate on the data itself rather than the structures and requirements imposed by the editor.
- Full-screen processing capabilities for the editor as well as other data manipulation functions for IBM 3270-compatible devices.

System command authority facilities to provide levels of system command authorization based on the needs of the various users, ranging from data entry to full system functions and facilities. File access control, established by the file owner at the file level.
RJE (or SEND) facility supported by RSCS or VNET.

eatures

System Requirements. UTS runs a virtual machine under ndahl's currently supported ease of VM/370. No modations to VM are required to erate Amdahl UTS.

Sales Facts

Packaging and Installation. Customers ordering Amdahl UTS receive the product on magnetic tape. When ordering, customers should specify either a 1600 or 6250 BPI 9-track tape.

Customers may license one copy of the *UTS Problem Determination Guide* (L1011.0) and one copy of the *UTS Programmer's Guide* (L1012.0). Also available is the *UTS Installation and Configuration Guide* (M1085.0) and the *UTS General Information Manual* (M1081.0). Availability of documentation is planned to coincide with date of first customer shipment.

Support and Service. UTS Release 1.0 is distributed only with central support. Support for UTS will be provided by Amdahl's Central Programming Services as specified in the UTS License Agreement.

Price. Amdahl UTS (4PV0-F1) is licensed to holders of a current Amdahl Maintenance Agreement under a UTS License Agreement. The monthly license fee will be $3,000 per processor, with a 6-month minimum installation period. Price is subject to change without notice.

First Customer Shipment. First customer shipment for UTS Release 1.0 is planned for second quarter, 1981.

Availability and Delivery. The customer must have and maintain a current license from Western Electric for UNIX Timesharing Operating System Version 7.0 for the processor designated to operate Amdahl UTS. Acknowledgement from Western Electric that the customer has a current license for the UNIX Timesharing Operating System Version 7.0 is a condition precedent to the granting of a license for Amdahl UTS. Shipment of UTS will occur within 15 days from receipt by Amdahl of a signed UTS License Agreement.

For more information about Amdahl UTS and other Amdahl products, contact your local Amdahl marketing representative.

amdahl®

Amdahl Corporation
1250 East Arques Avenue
P.O. Box 470
Sunnyvale, California 94086
408/746-6000

XENIX OPERATING SYSTEM

A Portable UNIX Version 7

Now Available for PDP-11s

Human Computing Resources Corporation announces the availability in Canada of XENIX™, Microsoft's adaptation of Bell Laboratories' Version 7 UNIX™ operating system. For the last decade the UNIX system has set the standard for power, flexibility, and useability among time-sharing systems. UNIX versions already exist on a wide variety of minis and mainframes. Within the next year, XENIX will be available on all major 16 bit microcomputers. *Today,* XENIX provides a complete UNIX Version 7, useable on the PDP 11/23, 11/34, 11/40, 11/44, 11/45, 11/55, 11/60, and 11/70, for prices starting at $2995.

Features of XENIX

- An interactive, multi-user, multi-tasking operating system with a flexible user interface.
- A tree-structured file system. Files may be shared among various users or protected in several ways. Subtrees representing specialized file systems may easily be added or removed.
- Compatible file, device, and interprocess input/output. This means that programs can easily communicate with different devices which can be chosen at run-time. Programs can just as easily communicate with the file system or with each other. Users need never again suffer the indignities of obscure and error-prone job control languages.
- The ability to initiate asynchronous processes, which can then communicate with each other.
- A flexible command language. This allows foreground and background program execution, the chaining together of various software tools, and the construction of background batch streams.
- System accounting and security access protection.
- Over 100 utility programs.

Applications

- Program development. Programmers throughout the world have acclaimed the power and congeniality of the UNIX environment.
- Word processing and document preparation. The UNIX system includes a text editor, typesetting software for correspondence quality printers and for a phototypesetter, formatting packages for common document styles, programs that compose tables and mathematical equations, programs that assist in checking spelling and constructing bibliographies, and numerous other utilities.
- Data processing applications, both local and distributed.
- Scientific, engineering, and educational applications.

Systems Software Included with XENIX

- C and FORTRAN 77 compilers that produce compatible object code.
- Top-down and bottom-up tools for constructing compilers and interpreters.
- Assembler, linkage editor, loader.
- Powerful symbolic debugger.
- Device drivers for dozens of common input-output devices.

NOV 80

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

AU UGN 45 416-922-1937

Other Software Available with XENIX

For an extra charge, the following are available now or soon:

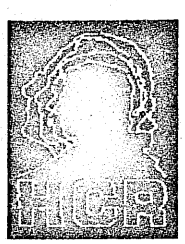
- RT/EMT™, a software package which allows RT-11* software development to occur under UNIX, available now.
- PASCAL, BASIC, and SP/k (a PL/I subset), available now.
- A full line of C compilers and cross-compilers.
- COBOL, and several data base management systems.
- The UNET™ network communications package from 3COM Corporation, Menlo Park, California.

Portability of Programs and Programmers

- XENIX is an enhancement of the original Bell Laboratories UNIX Version 7, modified to port it to new machines. Unlike new implementations, it has undergone development, evolution, and testing at several thousand sites for over a decade.
- Versions of UNIX now exist for the DEC PDP 11/23, 11/34, 11/40, 11/44, 11/45, 11/55, 11/60, and 11/70, for the DEC VAX 11/780, the Perkin-Elmer (Interdata) 8/32, the Honeywell Level 6, the Amdahl 470 (where it is called UTS), and others.
- Microsoft is developing versions of XENIX for the Zilog Z8000, for the Intel 8086, and for the Motorola M68000. These will be available in 1981.
- Because of the widespread availablity of UNIX and XENIX systems on minis, mainframes, and the popular 16-bit microprocessors, software can be protected against hardware-obsolescence, and software efforts can be shared instead of duplicated. Not only will software be portable, but so will be programmers! They will be able to move from system to system with the same software and the same skills.

Support, Availability, and Price

- Microsoft has set the standard for price, performance, and support in software for 8-bit microprocessors. Now XENIX, a true UNIX Version 7, promises to be the standard operating system for the computers of the eighties.
- Microsoft and Human Computing Resources Corporation will provide customizations and modifications for OEMs and other purchasers.
- One year maintenance and bug fixing is included.
- A regular newsletter will keep customers abreast of new developments.
- Initial customer training is included. Further customer training is available, at your site if desired.
- A reasonable amount of telephone assistance is also included.
- XENIX is available *now* on the PDP-11s listed above.
- A perpetual license to use XENIX on a single CPU, plus one year's support as described above, costs an amount depending upon the maximum number of users who may be logged in to the system at any one time. For a 1 user system, the price is $2995. For 2 users, $3395; 3 users, $3695; 4 users, $3995; and, for each user over 4, add $250. Thus an 8 user system is $4995, and a 16 user system, $6995.
- For further information, or to arrange a demonstration of UNIX, call Human Computing Resources Corporation.



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

RT/EMT

RT-11* under UNIX*

Human Computing Resources Corporation announces the availability of **RT/EMT***, a software package which allows RT-11 software development to occur under the UNIX timesharing system. The package consists of two major software items, an RT-11 emulator, and an RT-11 command interpreter. The emulator directly executes *unchanged* RT-11 binary programs. The command interpreter supports most of the commands of the RT-11 keyboard monitor. The use of the command interpreter is optional. The RT-11 emulator may be invoked directly at the UNIX command level.

Features

- RT-11 binary files may be executed without any changes. The features of the RT-11 V3B SJ monitor are supported.
- System overhead is comparable to RT-11.
- Runs in user mode. UNIX operating system changes not required.
- RT-11 file systems are simulated by UNIX directories containing ordinary UNIX files. Thus UNIX utilities may be applied. (A separate program may be used to copy files to and from actual RT-11 media.)
- Each RT-11 user has his/her own simulated file system. Under UNIX timesharing, more than one user may develop RT-11 software simultaneously. Unlike some other RT-11 "timesharing" systems, no one user can bring down the entire system.
- Full source included.

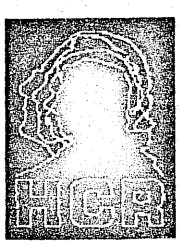
Applications

- Users of RT-11 applications programs can make use of UNIX timesharing without learning a new system.
- Useful in place of additional RT-11 development systems. If your programmers require another RT-11 development machine, you should consider RT/EMT instead.
- Large RT-11 software development projects can be centralized in one machine. UNIX facilitates cooperative projects and sharing of code. Backup copies of valuable RT-11 files are made as part of the normal UNIX file backup procedures.
- UNIX users can access RT-11 software packages. (For example, MACRO, LINK, FORTRAN, and many software products.)

Limitations

- RT/EMT works only on PDP-11 versions of UNIX. (It will not run under VAX or Interdata versions of UNIX. It will not run under Mini-UNIX.)
- HCR supplies only the RT-11 emulator and RT-11 command interpreter. Any RT-11 software (e.g. MACRO, LINK, PIP) requires a license from DEC. (Note: many UNIX sites already have such a license, since DEC often includes software as part of its hardware packages.)
- Emulated programs must not access device registers, device trap vectors, or undocumented monitor offsets.
- Certain other minor differences are documented in the RT/EMT documentation.

JAN 80



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

HCR/BASIC

A true BASIC for UNIX*

Human Computing Resources Corporation announces the availability of **HCR/BASIC***, a full function BASIC designed for the UNIX environment. HCR/BASIC conforms to the ANSI Standard X3.60-1978, and includes a number of powerful extensions.

Features

- Compatible with the majority of BASICs available today. Includes the following statements and commands:

append	*at eof*	*bye*	*chain*	*chdir*
clear	*close*	*data*	*def*	*del*
draw	*dump*	*edit*	*else*	*end*
endif	*erase*	*exec*	*field*	*file*
fnend	*for*	*get*	*gosub*	*goto*
help	*if*	*input*	*let*	*linput*
list	*load*	*new*	*next*	*on*
open	*option*	*print*	*put*	*randomize*
read	*rem*	*restore*	*return*	*run*
save	*stop*			

- There is full support for strings and string variables, using functions such as: *mid$, left$, right$, space$,* and *str$.*
- Includes Tektronix line drawing graphics commands.
- File I/O allows the user to *input* and *print* up to six files at once. Random access to structured files is also supported.
- Block structured *if-then-else* is supported. The writing of readable programs is also facilitated by the ability to include blank lines and trailing comments.
- The new command *exec* allows the HCR/BASIC user to execute standard UNIX utilities under BASIC program control.
- Full source included.

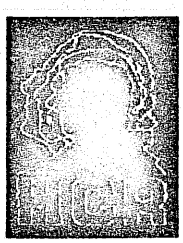
Applications

- BASIC is a very popular language, especially for microcomputers. UNIX sites interested in teaching programming will find HCR/BASIC very useful.
- Casual users of UNIX, especially those buying timesharing for business, may only know BASIC.
- A large number of application programs are available in BASIC. HCR/BASIC will run many of them unchanged.

Limitations

- HCR/BASIC works only on PDP-11 versions of UNIX.
- HCR/BASIC is an enhancement of UNIX BASIC, therefore, you must be licensed for UNIX source. (Licenses for binary sites by special arrangement.)

JAN 80



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

TROFF Spoken Here . . .

Technical Type & Composition is a commercial typesetting house with *Unix* facilities. We would be glad to typeset your project from a *tp tape* that has *troff -t code* or from a straight text or data file.

Type Styles Available: *ttc* has over 250 different type fonts. Styles range from the mundane *Times, Geneva and Century* to the exotic: *Trump Mediaeval, Avant Garde, Eurostyle, Latin Script, Garamond and Souvenir.*

We consider typesetting an art as well as a computerized science.
We therefore include the following features in all work which we troff:

Ligatures: We automatically insert ligatures unless you request otherwise. These special characters add a subtle, professional touch to any publication.

Kerning: There are over 200 character combinations that require kerning to make the visual spacing even and precise-looking. We automatically kern those combinations.

Archive: Your jobs will be kept for a period of one year to save you the expense of re-keying projects that you wish to reset at a later date.

RC repro paper is used on all jobs to give permanent, high contrast type on a translucent, bright white paper.

troff -t files sent to us on a 9 track, 800 bpi mag tape can be processed overnight. When kerning or exotic fonts are desired we must troff the file ourselves.

Prices

Phototypesetting Services	*Prices From*
troff -t > mag.tape_file | **ttc** | **cat mag.tape_file > /dev/cat &**	$2.00 per page
troff_mag.tape_file | **ttc** | **troff [*]**	2.50 per page
nroff_mag.tape_file | **ttc** | **troff [*]**	3.00 per page
non.*roff_mag.tape_file | **ttc** | **troff [*]**	4.00 per page

Advantages: Typesetting reduces weight, space and number of pages of any document. Typesetting increases the readability of your documents. Troffing from your mag tape avoids the delays and errors introduced when material must be re-keyed.

Oh Yes . . . We will also work from floppy disk & from non-*Unix* computer systems and their non-*Unix* tapes to produce directories, class schedules, books, catalogues, newsletters, documentation or *.

Technical Type & Composition

6443 Lardon Rd
Salem, OR 97303

(503) 371-8655

Whitesmiths, Ltd.
POB 1132 Ansonia Sta.
New York, N.Y. 10023
Cable: SOFTWARE NEWYORK
212 799-1200

Whitesmiths, Ltd. is a 2½ year old company specializing in the production of quality computer software. We currently employ over a dozen people in New York City and in Concord, Massachusetts. Our licensees number approximately one thousand companies.

Off the shelf products include C compilers for four target instruction sets -- the complete PDP-11 family, the Intel 8080 and Zilog Z/80, the VAX family, and the Motorola 68000. Our compilers run on all major operating systems supported on these machines: UNIX V6, UNIX V7, RSX-11M, RT-11, IAS, RSTS/E, CP/M, CDOS, UNIX V32, VMS, and VERSAdos. A full line of cross compilers for the Intel 8080 is available. With each compiler comes an extensive library of portable functions.

Now in prerelease to selected customers are a cross compiler plus assembler/loader for the MC68000, a Pascal to C translator for full ISO Pascal, and a UNIX equivalent operating system called Idris currently operating on the LSI-11. Available under Idris is an extensive collection of utilities for word processing, program development, and applications support.

Future plans call for a memory managed PDP-11 Idris, more utilities, Idris ported to other machines supported by our C compilers, and new C compilers for machines such as the Intel 8086 and the IBM 370.

For more information write or call for our Fall 1980 Software Catalogue.

Systems: 1780 Broadway, Suite 601, New York N.Y. 10019. (212) 245-5841
Research: POB 544 Concord, Ma. 01742. (617) 369-8499
Corporate: 19 North Sixth Avenue, Highland Park, N.J. 08940.

COHERENT™

A Preliminary Description of the System

The Mark Williams Company announces **COHERENT,**™ a state of the art, third generation operating system. **COHERENT** is a totally independent development of The Mark Williams Company. **COHERENT** contains a number of software innovations not available elsewhere, while maintaining compatibility with UNIX*. From this starting point, it goes on to include further substantial software innovation. The primary goal of **COHERENT** is to provide a friendly environment for program development and the subsequent application of programming tools to a wide diversity of tasks. The intent is to privide the user with a wide range of software building blocks from which he can select programs and utilities to solve his problems in the most straight-forward manner.

COHERENT and all of its associated software are written totally in the high-level programming language **C.** In fact, one of the premises of the system is that essentially all programming will be done in one of a number of portable, high-level languages—namely, **C,** Pascal, or Fortran 77. The portable **C** compiler conforms with the standard described in *The C Programming Language* (by Kernighan and Ritchie, Prentice-Hall, 1978). In addition, all of the features introduced into the language since this standard are also included in the compiler. Using **C** as the primary implementation language yields a high degree of maintainability, portability, and ease of modification with no noticeable performance penalty. Two of the basic tenets observed throughout the design are that good performance is more a result of well-designed algorithms than of coding tricks, and that the reliability gained by implementation in a higher-level language far outweighs any small loss in execution speed.

1. Summary of Features

COHERENT provides **C** language source compatibility with programs written to run under Seventh Edition UNIX, enabling the large base of software written to run under UNIX (from numerous sources) to be available to the **COHERENT** user. The major features of **COHERENT** include:

- multiuser and multi-tasking facilities,
- running processes in foreground and background,
- compatible mechanisms for file, device, and interprocess i/o facilities,
- the shell command interpreter—modifiable for particular applications,
- distributed file system with tree-structured, hierarchical design,
- unified file structure and input-output system,
- pipes and multiplexed channels for interprocess communication,
- asynchronous software interrupts,
- generalized segmentation (shared data, writable instruction spaces),
- ability to lock processes in memory for real-time applications,
- fast swapping with swap storage cache,
- minimal interrupt lockout time for real-time applications,
- reliable power failure recovery facilities,
- fast disc accesses through disc buffer cache,
- loadable device drivers,
- large non-mapped buffer cache,
- process timing, profiling and debugging trace features.

*UNIX is a trademark of Bell Labs

In addition to the standard commands for manipulating processes, files, and the like, in its initial release **COHERENT** will include the following major software components:

- **SHELL** the command interpreter
- **STDIO** portable, standard i/o library plus run-time support routines
- **AS** assembler for the host machine
- **CROSS** a number of cross-assemblers for other machines with compatible object format with **"AS"** above
- **LD** linkage editor for **"AS"** or **"CROSS"**, above
- **AR** file archiver and object library maintainer
- **DB** symbolic debugger for **C, Pascal, Fortran,** and assembler both run-time and post-mortem debugging
- **ED** context-oriented text editor with regular expression patterns
- **SED** stream editor, fashioned after **"ED"**
- **GREP** pattern matching filter
- **AWK** pattern scanning and processing language
- **LEX** lexical analyzer generator
- **YACC** parser generator language
- **NROFF** Nroff-compatible text formatter; output devices range from terminals to typesetters
- **LEARN** computer-aided instruction about computers
- **DC** desk calculator
- **LOGIN** control access to the system
- **QUOTA** package of accounting programs to control filespace and processor use
- **LPR** spooler for line printers
- **MAIL** electronic personal message system

Of course, **COHERENT** will have an ever-expanding number of programming and language tools and basic commands in future releases.

2. Command Interpreter

The *shell*, or command interpreter, of **COHERENT** provides the basic user interface to the system. It is the normal way of controlling the filesystem, processes and their interactions. The following is a brief summary of the major features:

commands with arguments,
pipelines,
redirection of command input, output and error,
background processing,
shell variables and expression manipulation,
filename expansion **(GLOB)**.

The essential contribution of such a programmable shell is that many tasks can be accomplished by a shell file (program) rather than by a program written in a compiled language, such as **C.** It has been found that such "shell programs" are debugged considerably faster, both because the programmer is working with higher-level basic constructs, and because they are easier to maintain than programs producing similar results but written in a compiled language. Very often, non-programmers can implement substantial applications in the shell that they would never have been able to achieve by some other means.

3. Language Support

The realm of language support is one of the major strengths of **COHERENT.** The following language processors will be supported initially:

- **C** a portable compiler for the language **C,** including stricter type enforcement in the manner of **LINT.**
- **FORTRAN** portable compiler supporting the full ANS Fortran 77 standard.
- **PASCAL** portable implementation of the complete ISO standard Pascal, with strict checking at compile time. A true compiler with state of the art error recovery and interactive i/o.
- **XYBASIC™** a state of the art Basic with the interactive capabilities of an interpreter and the performance advantages of a compiler.

The unified design philosophy underlying the implementation of these languages has contributed significantly to the ease of their portability. In particular, the existence of a generalized code generator is such that with a minimal effort (i.e. about one man-month) all of the above language processors can be made to run on a new machine. Another result of the unified approach to language processors is that the code generation effort is done once and if it is well-written, all of the associated compilers benefit. The net result is that the compilers running under **COHERENT** produce extremely tight code very closely rivalling that produced by an *(experienced)* assembler programmer. Finally, the unified coder and conformable calling sequences permit the intermixture of these languages in a single program.

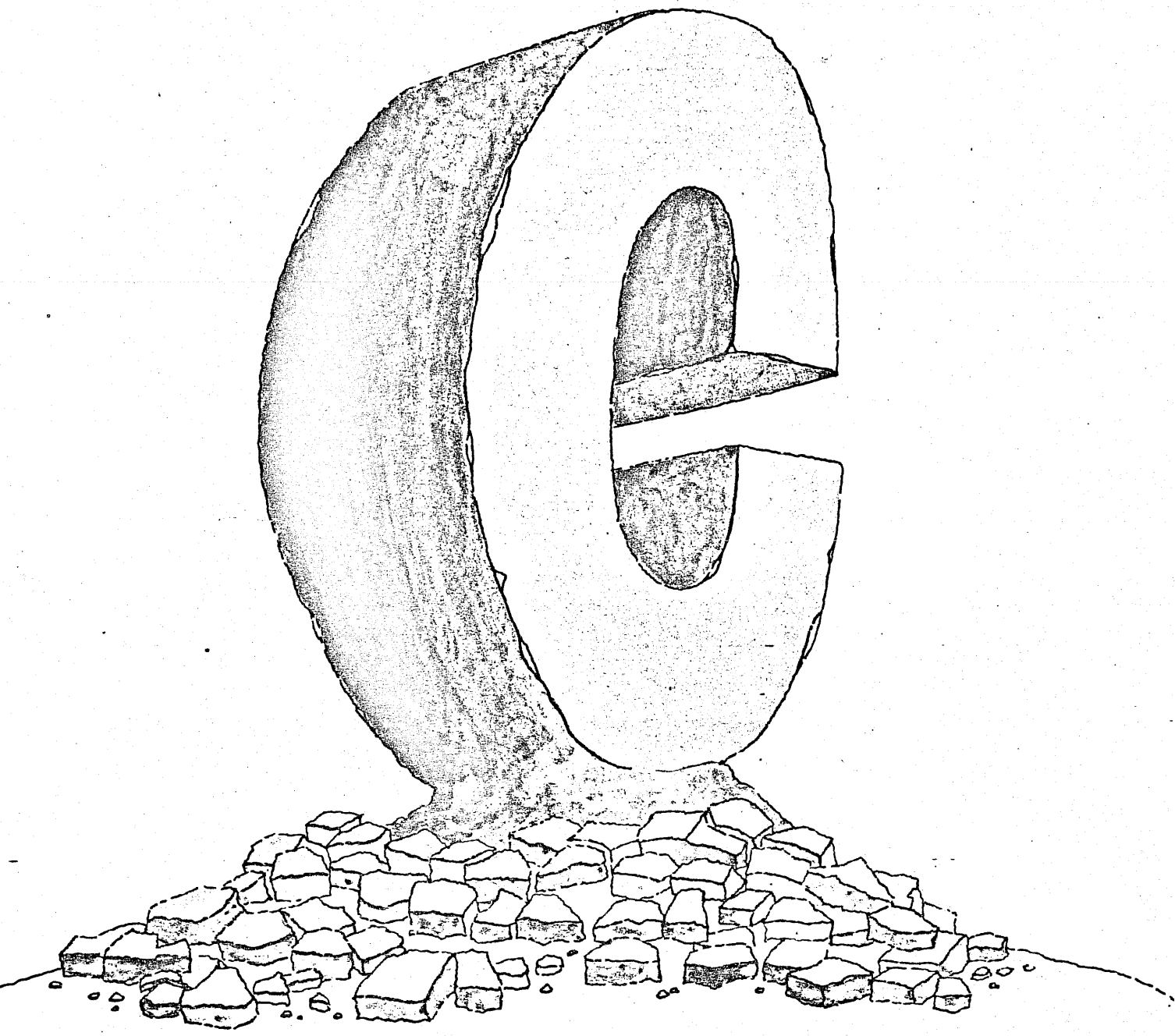
4. The Operating System

In part because of the language portability discussed above, and in part because of a substantial effort in achieving a greater degree of machine-independence in the design and implementation of the **COHERENT** operating system, only a small effort need be invested to port the whole system to a new machine. Because of this, an investment in **COHERENT** software is not tied to a single processor; but rather, applications can move with the entire system to a new processor in the order of 2 man-months effort by qualified programmers.

The initial version of **COHERENT** is available for the Digital Equipment Corporation PDP-11 computers with memory-mapping, such as the PDP 11/34. Machines which will be supported in the coming months are the Intel 8086, Zilog Z8000, and Motorola 68000. Machines for which ports are being considered are the DEC VAX 11/780 and the IBM 370, among others.

Because **COHERENT** has been developed independently, the pricing is exceptionally attractive. Further, it comes with complete end-user support from the developer enabling the system to be extremely viable in industrial and commercial applications.

Address inquiries to:
Mark Williams Company,
1430 W. Wrightwood Ave.,
Chicago, IL 60614
Tel: 312-472-6659



THE C MACHINE

THE FIRST COMPUTER OPTIMIZED TO EXECUTE THE C LANGUAGE

BBN Computer

Introducing C/70, BBN Computer's new UNIX®-based software development system. The first computer designed around C as its machine language.

The C Machine executes C programs fast. With 20-bit addressing, it gives you up to sixteen times the single-program space available in 16-bit computers. And it comes with a microcode that's user-programmable for efficiency as well as custom applications.

Now there's a software development system that requires no lower level language than C. For operating systems, general utilities and network communications. For real time systems.

BBN Computer's C/70, the first system optimized for the C language.

A FULLY SUPPORTED UNIX OPERATING SYSTEM

UNIX is an advanced interactive operating system of unusual simplicity and generality. Its high intelligibility makes it simple to comprehend, yet powerful enough to do things users want.

The C Machine is the first computer designed around UNIX. Specifically, an enhanced version of the Western Electric UNIX, Version 7, has been written totally in C. It's fully supported by BBN Computer and features network capability.

The UNIX kernel of the C Machine is remarkably well space-encoded. Because of the C Machine's unmatched instruction efficiency, its UNIX kernel is 30 percent smaller than kernels in other computers. The same unmatched instruction efficiency allows the C/70 UNIX kernel to be coded totally in C. This means easy maintenance and high reliability. By contrast, machines not optimized for UNIX (like the PDP-11/70)* require over 1K of assembly language code in their UNIX kernel to perform satisfactorily.

Other C Machine features include a fast function call to encourage modularization of programs. A compiler that matches the definitions in *The C Language*, the reference book by Kernighan and Ritchie. And support of sharable text code.

Also included are electronic mail, network access, and a crossnet debugger.

Electronic mail runs on the mailbox principle with cross-network routines. Specifications include multiple addressing, address lists, forwarding, and selective filing.

The network control software connects UNIX to any network that uses BBN C/30 or Pluribus packet-switch processors. This includes ARPANET. Equipped with the interactive terminal software TELNET, UNIX interfaces a remote user as a virtual terminal. This means you can access remote net hosts as though you were a local terminal on that host. In turn, remote users look like local terminals to UNIX. Also included is a file transfer program (FTP) that lets you efficiently transfer bulk data files to/from remote and local hosts and lets remote users transfer files to/from UNIX.

With the crossnet debugger, a C program resident on the C Machine can be used to debug another C Machine. Down to the UNIX kernel level. In ARPANET or any other network that uses BBN C/30 or PLURIBUS packet-switch processors.

The C Machine's UNIX package offers extensive word processing software and more: interactive context editor, ED, text alignment and page formatter, NROFF, and a spelling correction editor with a 40,000 word dictionary, SPELL. Also available are tools that systematize compiler construction, YACC and LEX, and a report generator, AWK. Plus there are system management utilities—log-in accounting, file system consistency check, archive and library maintenance, system and process display, and on-line users' manual.

MEMORY ONLY OPERATING SYSTEM (CMOS)

A unique C Machine feature is its memory only operating system (CMOS). CMOS is designed for real-time use. It handles scheduling, process-switching and interTask communications. Write your applications code in C, link in the CMOS library, and run your real-time applications.

UNIX is a registered trademark of Western Electric. PDP-11/70 is a registered trademark of Digital Equipment Corporation.

THE C MACHINE PROCESSOR

The C Machine is a fast, powerful, microprogrammable 20-bit computer with 32-bit microwords. Processor components include 8K x 32-bit micromemory, a 1K x 12-bit dispatch memory, a 1K x 20-bit fast register scratchpad, and an arithmetic-logical unit (ALU), all connected by source and destination buses.

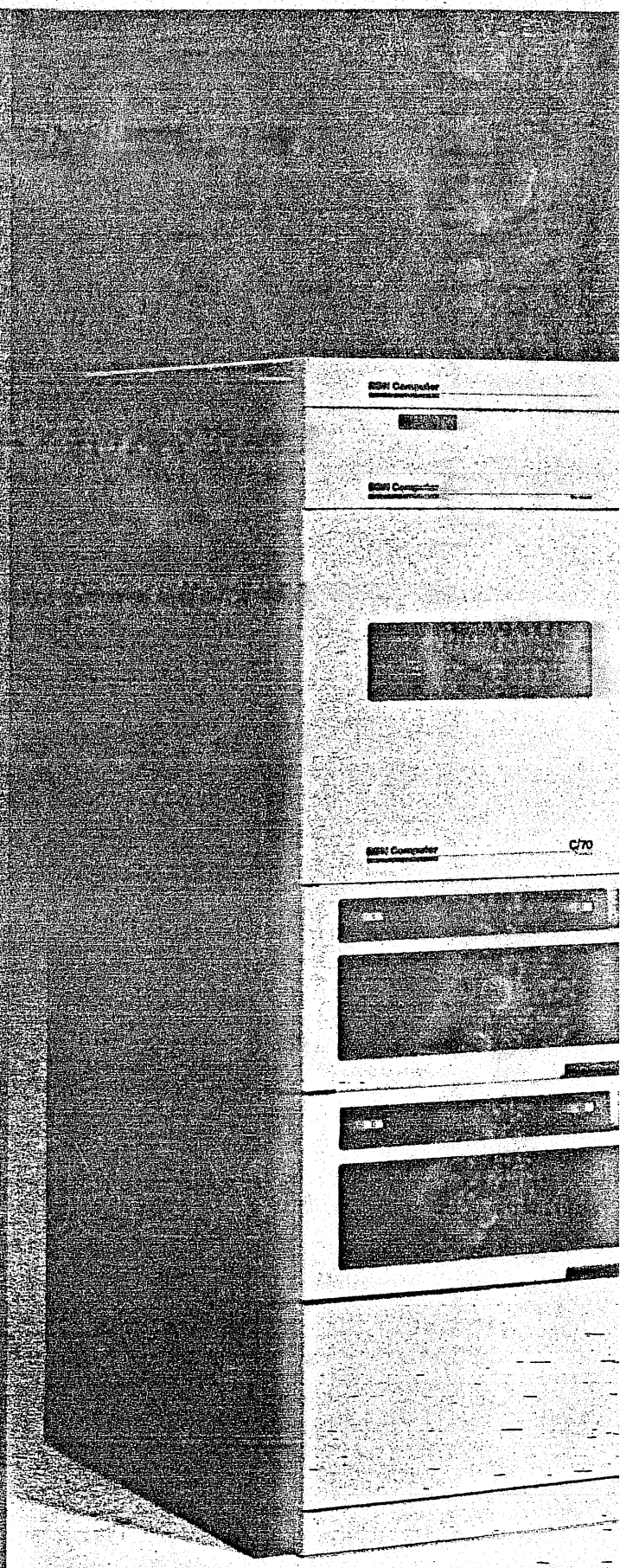
An instruction mapper and memory memory management unit specialize this architecture for efficient execution of the C programming language and UNIX. The instruction mapper takes the C instruction set and does some elementary bit swapping, rather than force the microcode to do shifting. It then sends an output to dispatch memory that quickly generates the right branch and allows microcode routines to be relocated.

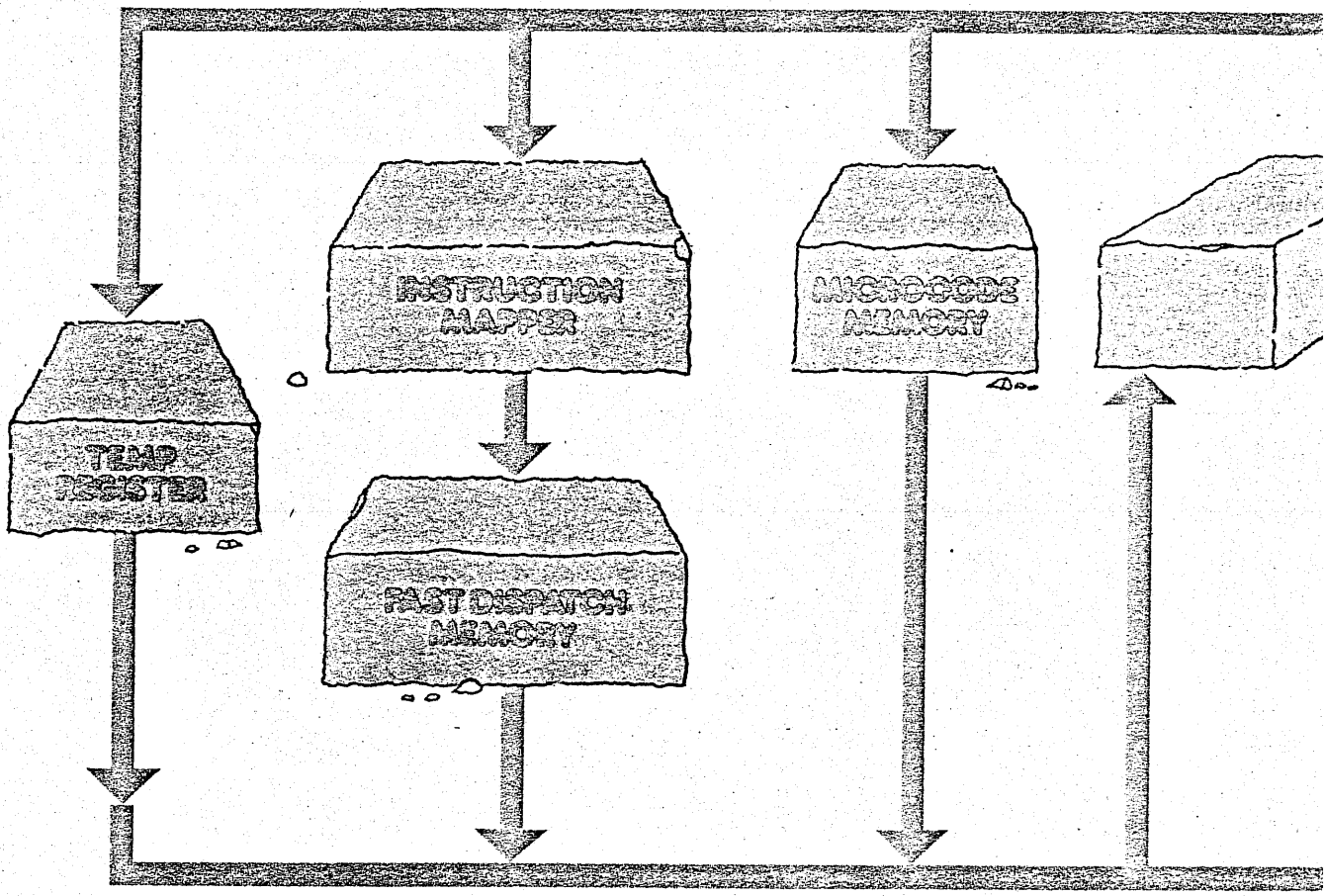
The C Machine boasts high reliability. A self-check microcode and error detection and correction memory—with six additional memory bits per word—detect all two-bit and correct all single-bit errors.

MICROCODE INSTRUCTION

To insure maximum system performance, BBN Computer has implemented in microcode the most frequent constructs found in a large sampling of actual C programs. Approximately 5K of the system's 8K x 32-bit micromemory is occupied by C Machine support. The remainder is available for system programming. Microcoded instructions perform constant handling, referencing of automatic variables, fetching of C structure elements, function calling, and data type conversion.

The C Machine's instruction set is optimized for automatic variables. A single word instruction can reference up to 31 bytes of local automatic variables, one-, two-, or four-bytes long. This saves space since the number of automatic variables declared in an individual subroutine tends to be small—two or three on average. Further, while most machines have different instructions for short and long variables, the C Machine uses the same instruction with a reserved value to fetch long variables from the next word.





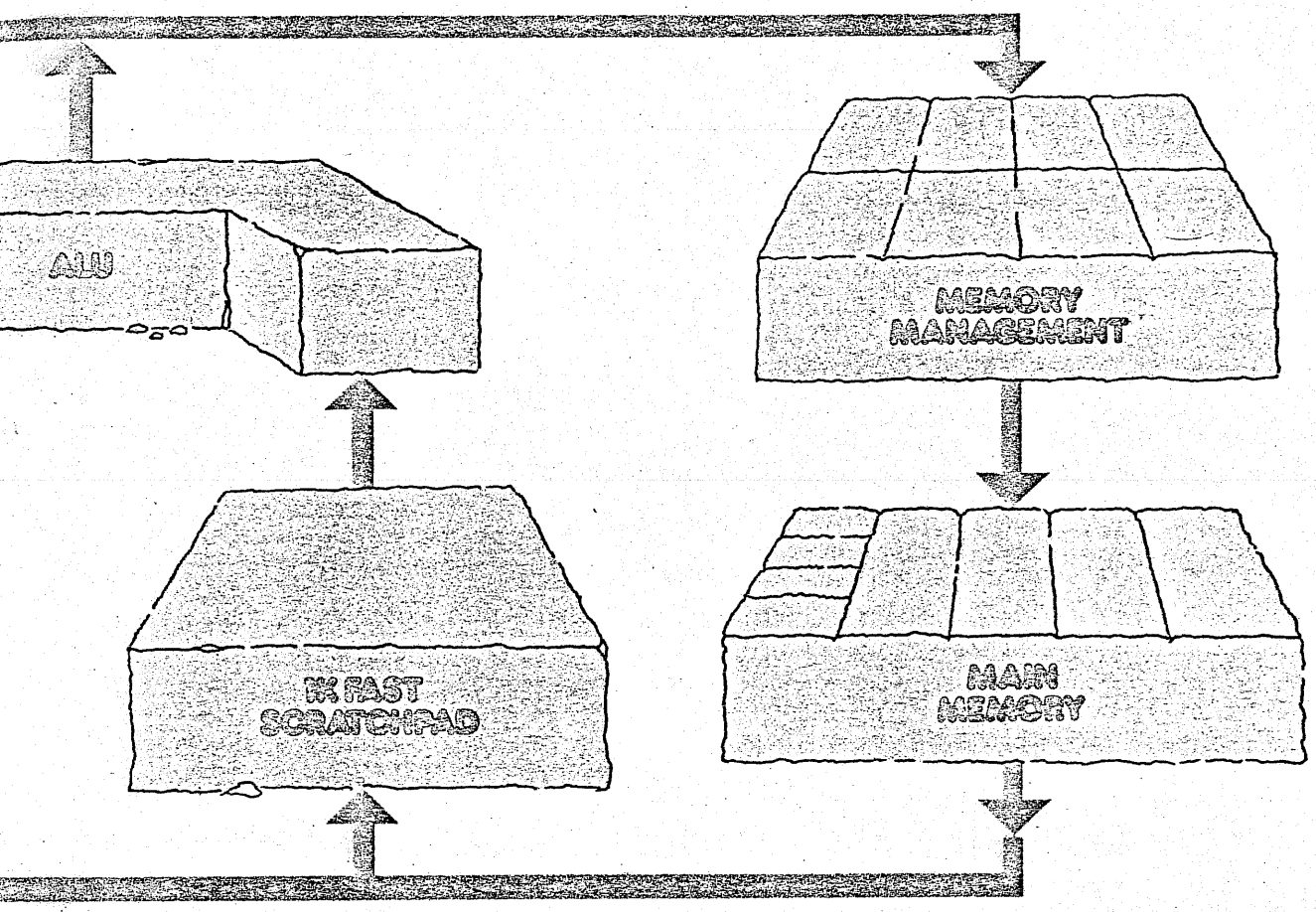
Constants are represented in a similar way and most are generated in one word. Such schemes significantly reduce the C Machine's required memory, thus minimizing program space and execution time. And addressing modes are optimized for fetching those C variables. The C Machine has 17 addressing modes, including: constant, resister, resister indirect, immediate, static, static indirect, automatic local and automatic local indirect. These hardware capabilities directly support addressing in the C programming language.

In most computers, special machine functions such as I/O and memory mapping are accessed through assembly language. The C Machine provides these capabilities through C-level subroutine calls to microcode routines. Consistent call linkages mean that you use the same syntax for microcall routines and any C routines you write. Hence, a frequently performed function can be moved to microcode for faster execution without modifying the application programs which call it.

SCRATCHPAD

Perhaps the C Machine's most exciting feature is its large, high-speed scratchpad—1K x 20 bits that allows you to perform subroutine calls fast. With 1,000 hardware registers, context saving can be done by simply designating a new set. This procedure dramatically reduces save and restore time, down to 8 microseconds.

Operating system variables, most of them I/O related, occupy 384 words of the C Machine's scratchpad. The rest of the scratchpad holds C register variables. There are 128 words for system registers and 512 words for user registers. Both are supported by the C language concept of a register variable that lets you explicitly declare a program variable be stored in a fast register.



MAIN MEMORY

The C Machine can access up to two megabytes of physical memory in 540 ns. The 20-bit word length means that single process space is up to sixteen times larger than in many 16-bit computers. Since most C programs are now written for these 16-bit computers, there is little danger that you will run out of memory when transforming them to the C Machine.

Main memory is organized to make management simple. Process space is divided into three regions: text containing executable and possibly sharable code, data containing static variables, and the stack containing automatic variables.

Text is loaded at the beginning of a process space followed by data. This separation allows multiple users to share a single copy of text, each having a separate data area.

The stack, which handles subroutine variables, begins at the top of the process space and grows toward the data area. This allows excellent usage of memory, since space is dynamically assigned when a subroutine begins and then reclaimed when it is finished.

MEMORY MANAGEMENT

The C Machine's memory management unit provides a one megabyte segmented virtual memory per process. It has access control, address translation, fault handling logic, and fast context switching.

Translation of logical byte addresses to physical word addresses, fault detection, and access verification take only one machine cycle. Access flags provide read only, read/write, execute only, and full access privileges. These rights are supplemented by text and data flags to insure that text and data segments are protected from improper reference.

Segment bounds are also monitored. Status bits indicate which segments have been modified, which referenced, and which are not currently in physical memory. These hardware indicators significantly speed swapping decisions in the C Machine.

The memory management unit has space to handle eight individual process spaces. This the amount of context switching necessary to run multiple processes is cut down significantly because there are often less than eight C Machine

users in the run queue, then eight process spaces can be set up and left alone while the C Machine switches between users by changing only one pointer. Other computers (with their limited number of process maps) are forced to do time-consuming swaps much more often.

INPUT/OUTPUT

To minimize I/O interface hardware and the cost of peripheral controllers, many traditional hardware functions are implemented in microcode. I/O controllers perform only very basic tasks such as buffering, handshaking, electrical level conversion, and serial-parallel conversion. Microcode performs all other functions such as transfer of data to memory (DMA), status reporting, word assembly and disassembly, parsing, and checksumming.

Using microcode for most I/O logic allows a single design of high speed communications port to serve asynchronous, synchronous, and binary synchronous devices. And changing a single IC allows support for SDLC/HDLC.

With the C Machine's full duplex communications I/O board supports six binary synchronous and four ARPANET host ports. A terminal card supports 32 asynchronous ports up to 19.2 Kbaud with modem control, and one HDLC or one ARPANET host port. A standard ARPANET 1822 interface is used for network I/O.

The C Machine Processor also includes two serial asynchronous ports. One serial port runs the dynamic debugger (DDT) from the operator's console. The other interfaces a microcassette for automatically reloading firmware.

The C Machine's disk control supports up to two industry-standard Storage Module Disk drives (SMD) of any size and includes a 256-word FIFO to increase system throughput. Compared to 80% with other controllers, a full 89% of the C Machine's disk (640 data bits per 720 bits) is available to the user. On a 160 Mbyte drive, this means an additional eight to ten Mbytes of data can be stored.

BBN Computer was able to achieve this efficiency by tailoring the microcode for disk I/O. The microcode provides many of the more complex disk control functions such as error correction and conversion of logical sector addresses to physical head/track/sector addresses. This significantly reduces disk controller hardware.

For other types of serial devices such as CRT's, the C Machine uses a unique I/O scheme. Microcode controls free buffers associated with a device's queue before interrupting the processor. Thus C level involvement in I/O is much smaller than other computers which must ask an I/O device explicitly to transfer data to memory.

PHYSICAL SPECIFICATIONS:

The C Machine is mountable in a standard 19 inch rack cabinet and occupies 19 inches of vertical space. The basic chassis supports eight cards: three CPU and memory cards, two I/O cards, and three slots for user options.

The C Machine draws 500 watts of power, 60 Hz 120V or 50 Hz 220V. Weight is 60 lbs.

PERFORMANCE COMPARISON*

	C/70	11/70
FUNCTION CALL		
INTERPROCESS I/O		
SYSTEM CALL		
PROCESS SWITCH		
ADD		
EXECUTION TIME		

*Actual times and source listings available upon request.

Hints on Configuring a VAX*
(Revised: January 22, 1981)

Bob Kridle and Bill Joy

U. C. Berkeley

ABSTRACT

This document reflects our experiences and opinions in configuring four existing VAXes and two more which are on order. We run UNIX† on all the VAXes and do not concern ourselves here with VMS* drivers and support. UNIX Drivers exist for all the peripherals we recommend.

We do not consider devices which have proven unreliable or whose performance we consider inadequate. In addition, there are a large number of devices with which we have no experience. As a general rule, every new peripheral has required a non-trivial amount of leg work to get up to speed. We suggest using only peripherals which have been previously used successfully on *the type of VAX you are configuring* (780 or 750) or demanding a substantial (50-100 percent) discount for being a guinea pig. Be especially careful of UNIBUS interfaces. Almost every manufacturer of a UNIBUS widget now includes the VAX as a machine on which his device will work. Few of these devices have actually been tested in this situation. Many of them will not work without substantial modification.

Our prime considerations in choosing equipment are:

* Cost
* Maintainability and Maintenance Cost
* Delivery Time
* Redundancy of the system

Disclaimer: This documents reflects our PERSONAL OPINIONS. We are responsible for software and hardware support of VAX systems, and the recommendations we give we have taken ourselves. You may get a lemon, no matter what you buy. All we promise is that this is what we believe. Let us know what you find out. Prices are constantly changing and may be inaccurate.

† UNIX is a trademark of Bell Laboratories.
* VAX and VMS are trademarks of Digital Equipment Corporation.

Overview

We first discuss components, listing the alternatives we have tried and sometimes a few we haven't, and then discuss system packages. We buy a substantial portion of the equipment we use from vendors other than DEC. In all cases, maintenance of such equipment is a consideration and we discuss the way in which we maintain such equipment. The reasons for choosing second vendor equipment are essentially always lower cost equivalent equipment and shorter delivery time.

If you have the money and time required to acquire an all DEC system, there are some advantages. DEC equipment we have here has, in general, proven somewhat more reliable than the equivalent alternate vendor equipment. Time from equipment delivery to running system is also usually shorter. Field service in our area is excellent from DEC and is better than that provided by the alternate peripheral vendors in most areas. For smaller installations this option should be carefully considered. It is nice to call one party for all your problems, if you can afford it. At Berkeley, we are well past the inventory level where self maintenance begins to pay off even on all DEC systems, so this is not a consideration.

We recommend getting field service on at least your CPU for at least the first year. In our area VAX service has been excellent from DEC. It has paid off for us in the cost of parts alone. You can drop the contract after the engineering changes have tapered off and most of the infant failures have occurred. DEC requires a certain amount of its peripheral equipment on the machine to qualify for field service. Some DEC field service offices are not willing provide a DEC maintenance contract without a DEC mass storage peripheral. We have yet to see a reason for this (on a 780), as all the relevant diagnostics run on floppies. If you want to argue with DEC about this, we can provide references for some offices which don't require DEC tape or disks.

All VAXes come with some amount of DEC memory (min. 512K) and VAXes are sold in packages which include disks and sometimes tapes. Unfortunately, the cheapest packages use disks which function poorly (RK07). Currently, we know of no way to reliably run any other DMA on the UNIBUS with these disks. In addition they are small and slow. You can sell them or keep them for service contract reasons.

Memory

Memory is the lowest risk alternate vendor choice. You get enough DEC memory with the system (.5M) to run DEC diagnostics. We buy the remainder of our memory from TRENDATA. (A list of second vendor phone numbers is given at the end of the document.) In the quantities we are buying, we are paying $6,780 per Megabyte. This is the price for government service contract purchasers and is independent of quantity. Universities receive a 15% discount, giving the following breakdown:

CPU	Qty	100%	85%	GSA
11/750	.5M	$6,780	$5,800	$4,990
11/750	1.0M	$12,360	$10,506	$9,980
11/750	1.5M	$18,450	$15,759	$14,970
11/780	.5M	$5,500	$4,675	$3,390
11/780	1.0M	$10,000	$8,500	$6,780
11/780	2.0M	$19,200	$16,320	$13,560

Small quantities (1 to 2 Megabytes) are available from stock; TRENDATA usually

ships in 5 to 7 days. Very large quantities (4 Megabytes and up) take 30 days at the worst. The RAM chips are socketed, and two replacement chips per board are supplied. When chips fail, the system prints out information about the failing chip, and you can pull out the board and replace the chip at your leisure. Since single bit errors are corrected this has never involved any unexpected down time for us. There is a one year return to factory agreement on the boards, included in the purchase price; worst case repair is said to be about 10 days. After one year you send the board and a purchase order not to exceed $225 to the factory for the same service. Typical cost is usually $175. We have returned one board to the plant in about 20 board years.

When purchased from DEC, memory is $16,630 per Megabyte in 1 Mb quantities or $13,560 per Megabyte in 2 Megabyte quantities for 780s. These prices reflect our 15% discount as will all the prices we quote from DEC. Delivery on DEC memory is typically very long, (six months or longer), and maintenance is $220 per Megabyte per month with board replacement through field service. The boards are not socketed.

Other vendors make DEC add-on memory; we are satisfied with TRENDATA and haven't experimented elsewhere. If you are going to have more than 4 Megabytes of memory you will need a CPU expansion cabinet at $3,620 and a second memory controller at $23,100 which comes with a second half-megabyte of DEC memory. We have had one on order for a long time, and haven't got a firm delivery date yet.

Disks

There are three, decreasingly expensive alternatives for disks. The first, and most expensive, is to buy the disk hardware from DEC. The available equipment includes 80M byte removable disks (RM03), 300M byte removable disks (RM05) and 120M byte non-removable disks (RM80). As we mentioned above RK07's don't count. If you get a system with these in it you can sell them or keep them for maintenance reasons only. It is currently possible to get about $15,000 for the two RK07's and controller which come in most packages.

All other DEC disks run on MASSBUS adapters, and are generally other vendors' (e.g., CDC or Memorex) drive hardware with some DEC electronics added. While the new, inexpensive UNIBUS controllers use bit-slice technology and microcode, most of the DEC controllers use a lot of discrete logic. This partially accounts for the high price of the DEC equipment. The following summarizes the cost of the drives available from DEC including installation and after our 15% discount:

Type	Size	First	Mtce	Add-on	Mtce
RM03	80M	$26,800	$170	$17,800	$140
RM05	300M	$38,500	$310	$29,500	$240
RM80	120M	$26,050	$125	$17,300	$95

The second alternative is to get a MASSBUS look-alike. Systems Industries will sell you a modified 9400 controller and storage module CDC 300M drives. This setup is less expensive than the same capacity from DEC, but is currently available only for 11/780's as far as we know. We have not had good experience with SI, although others are apparently satisfied with this equipment. (Note added in proof: see the mail received from George Goble which is attached at the end of this paper. We value George's opinions highly and he has had very good experience running VERY LARGE and heavily loaded systems using Systems Industries equipment. For such taxing applications, having the extra disk channels offered by MASSBUS adaptors or multiple UNIBUS adaptors is a definite advantage.)

The final alternative, and the one which we prefer, is to get a UNIBUS storage module disk controller and some storage module drives. We run most of our disks on the UNIBUS, and have had good luck doing so. All of our VAXes have EMULEX SC21/V controllers with AMPEX 9300-CD 300M drives. The EMULEX controller is a single card in the UNIBUS box and handles up to 4 storage module drives. We get a better price from AMPEX than from CDC, whose 300M drives seem about equivalent, and are perhaps a little more reliable. EMULEX and CDC have an agreement for service through CDC organizations of SC21's with CDC drives. We maintain our Ampex drives in house. We pay $4,600 for the Emulex controller and $10,840 for the AMPEX 9300 including control and read/write cables. We buy DYSAN error free 300M packs for $1,000 each (2 for each drive). Thus the total cost for a 300M drive and its packs is $12,840. Non-discounted CDC prices may add about $5,000. Below we show package prices including installation, cables, etc. Prices include 2 packs.

Another good storage module drive which is running on VAXes (11/750's) at Bell Laboratories with the SC-21V controller is the Fujitsu 160M Winchester drive. This drive costs $5500, quantity one. We have had EXTREMELY good reports on this drive, although we don't have any ourselves. The EMULEX storage module packages we suggest thus are:

Type	Size	First	Add-on	Mtce	Add
FUJITSU	160M	$10,300	$5,500	???	
AMPEX	300M	$17,840	$13,840	???	
CDC	160M	$14,000	$9,500	$225	$150
CDC	300M	$19,500	$12,500	$300	$224

Tapes

We buy EMULEX TC-11/P UNIBUS tape controllers for $3130 with cables, and Kennedy model 9300-3 800/1600 BPI 125 IPS transports for $6060. (This is a discount price; the small volume price is $7,200.) You also need a rack to mount the tape drive. Thus we get a 125 IPS, 800/1600BPI tape drive on the UNIBUS for $9190. EMULEX and CDC have a service agreement for their tape controller with CDC tape drives; the only price I could get was for a 45 ips tension drive similar to the DEC TS/11, for $14,000 installed through the CDC/Emulex agreement; this doesn't appear to be a good deal and we haven't tried the equipment. The Kennedy transport comes with a 15 month factory warranty. Our distributor exchanges/repairs the cards in the controllers based on local diagnostic (off-line) capability in the transport. After the warranty period, card swaps cost about $75. For transport mechanical failures the transport is returned to the factory in Monrovia, California, or we fix it ourselves. Disclaimer: We haven't run the EMULEX controller on an 11/750 yet, but do not expect that this will cause any problems, as it runs well on 11/780's.

Name	Speed	Densities	Cost
KENNEDY	125ips	800/1600	$9,190

Our original VAX system came in a package with a DEC TE16 on its own MBA. The TE16 is reliable, but is rather slow. The DEC TU45 is faster, but fraught with problems as the high maintenance cost reflects. The DEC TU77 is a good transport, but the auto-loading features don't seem to work very well, and it is rather expensive. DEC CSS has a 6250 tape package. It sells for about $45K for the first transport. It includes an autoloading, 1600/6250 125 IPS STC transport. The TS11 is included in packages for the 11/750 except for the RK07 package

system. It does not have a vacuum column, and is thus hard on tapes. It is a problem to load and seems to be rather unreliable.

Name	Speed	Densities	Cost	Mtce	
TS11	45ips	1600	$13,396	$75	(Not recommended)
TE16	45ips	800/1600	$22,000	$147	
TU45	75ips	800/1600	$25,424	$265	(Not recommended)
TU77	125ips	800/1600	$30,453	$235	

(Note added in proof: See the mail from George Goble attached to the end of this document about 6250bpi tape drives. He has recently tested a UNIBUS 6250 bpi drive which works on his 11/780.)

UNIBUS adapter

On machines which are to attach a lot of DMA devices on the UNIBUS, it is not a bad idea to get a second UNIBUS adapter. Our discounted (15%) price for a second UNIBUS adapter and a BA11-K UNIBUS mounting box is $13,800. We recommend this in our current favorite large system package.

Asynchronous terminal interfaces

With a VAX you get 8 lines of DZ-11 which provide some modem control but are NOT DMA! We recommend the ABLE DH-11 emulator, the DMAX/16, which gives 16 lines of DMA output ports for $5000. We have been getting about 30 day delivery on the these. Look for some other choices from ABLE and EMULEX in this area soon. The current alternative is more DZ-11's, which cost $1840 for 8 additional lines, which can be accommodated in the same distribution panel, then $3807 for each additional 16 lines. There have traditionally been delivery problems on DZ-11s.

Both the DZ's and the DH's have input silo's which UNIX can use to reduce interrupt load on input. The DMA output of the DMAX is especially important for graphics applications where high-volume and continuous output occurs.

Modems

We buy VADIC modem racks for the system end. VADIC offers modems which can talk both standard 300 baud and also the two commonly used 1200 baud styles: Bell 212 and VADIC. It is handy to have these "triple" or "3-way" modems in your machine.

Here is a price list of parts; these are quantity prices:

Chassis	$500	(Provides 16 slots)
Power Supplies	$200	(You can use two for redundancy)
300 baud Answer	$200	(Takes 1 slot)
3-way Answer	$750	(Takes 2 slots)
Autodialer	$400	(Takes 2 slots, and need 300 and/or 1200 stuff below)
RS232 to RS466	$400	(Converter for autodialer)
300 Orig/Ans	$390	(Takes 1 slot, for autodialer)
1200 Orig/Ans	$665	(Takes 2 slots, for autodialer)

A rack with 5 3-ways and no auto-dialers would cost about $4,500, and would have 6 free slots for additional modems. A rack containing 5 3-way modems, and a 300 baud and 1200 baud dialer would thus have only one free slot, and would cost about $6,500. This equipment has been plagued with infant failure problems but we know of no other equipment offering equivalent function.

The VADIC home-end originate answer 300/1200 modems which can also talk to Bell 212 modems are $750 when purchased with Voice/Data switches. These plug directly into modular phones. We have had trouble with initial reliability on these (about 1/3 don't work when received, and another 1/6'th fail within a couple of months.) We are investigating other modems (specifically VENTEL) because of the high initial failure rate on the VADIC's, but have no data yet.

Printers

We have been using some PRINTRONICS 300 line per minute dot-matrix printers. They do point-plotting at 60 points per inch. They are not outstandingly cheap ($7,000), but are very ruggedly built.

We have heard of a new Dataproducts B-600 printer, which is a 600LPM band printer. This also sells for $7,500 (with controller), and may be a good buy; Tom Ferrin at UCSF has one he is trying now.

Plotters

Wet paper plotters which are capable of 200 dots/inch are usable both as printers and as output devices for TROFF. We have an old model VARIAN (which requires a lot of attention); newer models are said to be less of a headache. A new VERSATEC 11" model sells for about $8,000. The objections to these guys are that the paper tends to be wet sometimes, stinky, and more expensive than line printer (5 to 10 times more expensive, at $20 per 1000 sheets).

Canon is working with Stanford on a desktop plain-paper copier which uses a laser and should provide good quality output. Early models (with limited availability) cost $12,000 including the interface. Later models may cost less. You may be able to get more information from Luis Trabb-Pardo at Stanford.

Small VAX 11/750 Packages

Our suggested small VAX Package consists of the following from DEC:

11/750 CPU
512K memory
8 lines of DZ-11
LA38 console
VAX/VMS license only

Packaged systems are often cheaper than components and the only option available to some buyers. The cheapest base is a system with RK-07's which we get for $68,425 discounted, and assuming no VMS support. If you can get $15,000 for the RK-07's, the DEC equipment in this system will cost you $53,425. If you can get this as a bare VAX CPU (OEM's can, for instance) you can get this for a discounted price of $39,950, save $13,475, and don't have to hassle with the RK-07s.

To this we add:

.5M TRENDATA memory	$5,000
EMULEX SC-21V disk controller	$4,800
160M Fujitsu winchester	$5,500
EMULEX TC-11P tape controller	$3,130
Kennedy 9300-3 tape transport	$6,060
Total starting from DEC package	$77,915

A similar system is available with everything from DEC; this with a TS11 tape drive (which we don't recommend), and an RM80, which is smaller than the Fujitsu. This system costs $94,010, (our price) $16,095 more than the package price; $29,570 more than the non-package price.

Medium 11/750 system.

It won't be possible to put more than 2M bytes of memory on this machine until 64K memory chips and controllers are available, so a truly "large" system doesn't make a lot of sense for a 750. To make a larger system, we add additional lines, disk storage, modems and memory up to 2 Megabytes:

1M TRENDATA memory	$10,000
160M Fujitsu Winchester	$5,500
ABLE DMAX/16	$5,000
VADIC Modems (5 triples)	$4,650
PRINTRONICS Line Printer	$7,000
Add-on cost	$32,150
Total cost starting from package	$110,065

To add this stuff on to the small all-DEC system we would add an RM80 for $17,300 and the other equipment (except for the the Fujitsu), for an add-on cost of $43,950, making the system price $137,960.

Small 11/780 Package.

For a system with more growth possibilities than an 11/750, faster processing (especially floating point), and higher i/o bandwidth, we recommend starting with a small 11/780. The system we would start with would include a VAX system package which would give:

11/780 CPU
512K memory
8 lines of DZ-11
LA120 console
Floating point accelerator
VAX/VMS license only

The $107,000 price here comes from a basic price of $143,900 from DEC for a package including RK-07s (SC-AXHHV-CK). We are assuming a 15% discount (to $122,570) and getting about $15,000 for the RK-07s.

We would then add-on:

1M TRENDATA memory	$6,780
EMULEX SC-21V disk controller	$4,800
160M Fujitsu Winchester	$5,500
EMULEX TC-11P tape controller	$3,130
Kennedy 9300-3 tape transport	$6,060
Total before tax and shipping	$133,270

An equivalent package from DEC would have the RM-80 and a TU-77 rather than the EMULEX/Fujitsu/Kennedy setup. The RM80/TU77 Based VAX package (SV-AXWBA-CA) would cost $171,062 with a floating point accelerator. We would not buy the TC-11 and Kennedy drive, the EMULEX controller or the Fujitsu disk, replacing them with the things in the package, and 16 more lines of DZ-11 for $3,807. Our small system price with 1M of TRENDATA memory would thus be $181,650, or $48,380 more than the non-DEC system.

Medium 11/780 System

As for the 11/750, we would add memory, disk, modems and a printer to the small system to get a medium sized system:

1M TRENDATA memory	$6,780
ABLE DMAX/16	$5,000
VADIC Modems (5 triples)	$4,650
PRINTRONICS Line Printer	$7,000
160M Fujitsu Winchester	$5,500
Total before tax and Shipping	$162,200

An equivalent package built on the Small DEC package adding another RM-80 instead of the Fujitsu, and DZ-11s rather than the DMAX would cost $39,537 more than the small all-DEC system, or $221,187; $58,987 more than the non-DEC system.

Large VAX Package.

For large VAXes we recommend 4 Megabytes of memory and 4 300M AMPEX drives split between EMULEX controllers on 2 UNIBUS adapters. You can also make this system with 2 300M and 2 Fujitsu's, evolutionarily. The system we would like to have would include a VAX system package which would give:

11/780 CPU
512K memory
8 lines of DZ-11
LA120 console
Floating point accelerator
VAX/VMS license only
UNIBUS expansion cabinet
2 BA11K boxes and backplanes
Second UNIBUS adapter

(This also assumes selling RK07's.)

We would then add-on:

3.5M TRENDATA memory	$23,730
2 EMULEX SC-21V disk controllers	$9,600
4 300M AMPEX drives	$60,960
EMULEX TC-11P tape controller	$3,130
Kennedy 9300-3 tape transport	$6,060
2 ABLE DMAX/16	$10,000
VADIC Modems (5 triples)	$4,650
PRINTRONICS Line Printer	$7,000
Total before tax and shipping	$255,490

To make the large VAX system into a DEC package, one would get a RM05/TU77 Based VAX package (SV-AXDBB-CA), which costs us $218,960 (after a 15% discount) and add three RM-05's for $88,500. We would add 3.5M of TRENDATA memory, would buy 8 300M DYSAN packs for $8,000 and 32 lines of DZ-11 for $7600. Adding the modems and the printer would bring the total system price to $358,440.

Sample system summary

In the summary which follows, "All-DEC" means that all mass storage and terminal interface devices are DEC supplied; all systems include second vendor memory, modems and printers.

Size	CPU	Mem	Disk	Tty	Tape	Modems	Lpr	Price	Ratio	Comments
Small	750	1M	160M	8	125ips	None	No	$64,440	0.82	Non-package CPU
Small	750	1M	160M	8	125ips	None	No	$77,915	1.00	
Small	750	1M	120M	8	45ips	None	No	$94,010	1.21	All-DEC
Med	750	2M	320M	24	125ips	5	Yes	$96,590	0.87	Non-package CPU
Med	750	2M	320M	24	125ips	5	Yes	$110,065	1.00	
Med	750	2M	240M	24	45ips	5	Yes	$137,965	1.25	All-DEC
Small	780	1.5M	160M	8	125ips	None	No	$133,270	1.00	
Small	780	1.5M	120M	8	125ips	None	No	$181,650	1.36	All-DEC
Med	780	2.5M	320M	24	125ips	5	Yes	$162,200	1.00	
Med	780	2.5M	240M	24	125ips	5	Yes	$221,187	1.36	All-DEC
Large	780	4M	1.2G	40	125ips	5	Yes	$255,490	1.00	
Large	780	4M	1.2G	40	125ips	5	Yes	$358,440	1.40	All-DEC
Large	780	4M	1.2G	40	125ips	5	Yes	$401,340	1.57	... with DEC memory

Conclusions

The second-vendor equipment in the packages here is all thought to work well on VAX hardware. You can reliably operate such a system if you can get maintenance for such equipment in-house, or through a organization associated with the vendor, or can stock some spares for non-redundant equipment. We have been able to configure VAX systems very cheaply because we can support, in house, good equipment which is available at a reasonable price. Even if you have no in house maintenance, agreements such as that between CDC and EMULEX make field supported VAX systems available at very lower cost.

Vendor References

ABLE	Async. interf.	415-932-0957	Eakins Associates (Joe Voelker)
AMPEX	Disks	408-733-2900	Ampex (Jess Clark)
CDC	Disks	415-942-0957	Eakins Associates (Joe Voelker)
EMULEX	Controllers	415-932-0957	Eakins Assoc (Joe Voelker)
KENNEDY	Tape Transports	408-245-9291	Electronic Marketing Specialists
PRINTRONICS	Printers	408-245-4392	Group III Electronics
TRENDATA	Memory	714-540-3605	Trendata
VADIC	Modems	408-727-6491	Moxon Electronics (Lee Berlind)
VARIAN	Plotters	408-733-2900	(Ted Downs)
VERSATEC	Plotters	408-985-5810	(Jim Muszalski)

A note from George Goble about 6250 tapes and SI disks

The following information from George Goble, Purdue University was received as this document was about to be printed via electronic mail. The mail is reproduced here after minor editing, and reflects his experiences and opinions:

I have tested a 6250 tape setup from AVIV corp, 6 Cummings Park, Woburn, MA 01801; (617) 933-1165. It is Telex 6253 drive (800/1600/6250 BPI) 125 IPS with a Telex Formatter and an AVIV 1 board UNIBUS interface. The UNIBUS interface has 4KB of FIFO, to help out with bus latency problems, which really seems to help out. The whole shebang costs about $32,000 + installation (about $1,000). It looks like a TU10. I am using the TU-10 UBA driver from 4BSD "newdev" with a little work.

The competition: System Industries (SI) has an STC 800/1600/6250 drive which will run on a VAX SBI interface, 11/70 Cache bus, or (almost ready) UNIBUS interface. There is only a 64 word FIFO. Mike Marsh seems to think the STC drive and the Telex are both good drives, with the Telex being slightly easier to maintain (we do self maint). The maint + 4KB FIFO + SI UNIBUS not quite ready are the factors why we went with AVIV for the drive, this does not mean the SI drive is bad. SI is the only ones with an SBI interface which should be used if you run tapes a lot on heavy load (more than just backups). The interface looks like a TU16/TU45 to the software. 6250 at 125 IPS pretty much hogs the entire UBA; it is faster than a disk!

I should point out that running any 6250 tape at 125 IPS on a UNIBUS IS GOING TO MAKE ABOUT ANYTHING ELSE GET DATA-LATES. Some devices like DMAX's and DMC-11's (with a lot of driver hacks) can and do run in this environment (our VAX). Even with a 4KB FIFO, there will be data-lates, although there will be a little breathing room. Running a UNIBUS disk with the tape will cause solid data lates, unless you have a smart disk controller with lots of FIFO (an SI 9400 has 4 sectors on the controller and runs with heavy UBA load without data-lating). Running a DEC disk (RK07) with almost no FIFO, will just not work. If you have UNIBUS disks with no FIFO, then the tape should go either on its own UBA or on the SBI.

We have had very good luck with SI as a company with disks. We have two CDC9766's (300 MB RM05's) on two SI SBI controllers on each of our VAX's and they run great. There is no look-ahead register though and SEARCH gets turned into a SEEK. Quantity one price is approximately $36,500 for 300 MB (RM05 size) and SBI controller. Extra drive is approximately $17,000. SI tape drive price is approx the same as AVIV. Also SI is working on a Single board UNIBUS controller which is close to done.

System Industries, Sunnyvale, CA, (408)-732-1650

The SI 9400 controllers correct (upto 11 bit) ECC errors in the controller, no driver assistance needed and are easily made dual-port, multi-cpu, etc. SI also has Fujitsu 160 MB Winchesters with SMD.

We currently buy our 300 MB packs from DYSAN, all of them are completely error free (cost about $1K) That with SI 9400 controllers, make bad blocks non existant. Dysan available from Dave Reikert, Dekalb Data products, (404) 972-0612. Unfortunatly, the new winchesters now coming out will have several bad-spots, so something will be needed in the future.

A note (added in proof) from George Goble about VAX UBA and DMC-11

Mike Marsh has a 3 chip change which fixes the DMC-11 bus-hang problem on a VAX with DMA devices farther down the bus. Also I think the DEC UBA speed up ECO is UBA rev1A or maybe 2. This has to be asked for and it is on new machines. One can call Frank Owens, local DEC field service manager (Ft Wayne, Indiana) at 219-456-3541 for more info on both of these. The DMC fix was done by Mike Marsh (Purdue/EE) and is not yet a DEC official ECO. Whenever one puts in a new UBA processor card, the ROMS invariably fall out in 2-3 days and must be pushed back in.

Changes in the VAX system in the Fourth Berkeley Distribution, November, 1980

Bill Joy

ABSTRACT

This document describes briefly the changes in the Berkeley system for the VAX between the distribution of January 1980 (known as 3BSD) and this, the fourth distribution, of November 1980. It attempts to be summarize, without going into great detail, the changes which have been made, and is intended to be used by someone who is familiar with the system to quickly re-acclimate themselves to the new system.

Major changes

- There is a new format for .o files and also a new archive format. Old .o files must be discarded and the programs compiled from the source again. Old archives can be converted to the new format with *arcv*(8), but if they contain old .o files they have to be completely recreated. New libraries for the loader must have a table of contents, see *ranlib*(1).
- The C compiler and associated programs now take arbitrary length names, not chopping them off at length 7 or 8. Other changes in C are described under cc below.
- There is a new version of the C shell, which supports "job control" and which uses a new tty driver. See *newcsh*(1) and *newtty*(1) for details. Programming considerations for the new system mechanisms involved are described in *jobs*(3). Users of the older csh can use oldcsh until they have time to learn the new features.
- The standard output (in the standard i/o library) is now line buffered by default when the output is a terminal. So that essentially all programs may work without change in the presence of this extra buffering, a partial output line is flushed whenever the library goes to the system to read from the standard input.
- A number of performance enhancements have been made. Under normal timesharing loads, the system time has been decreased about 10%. The changes to the standard i/o library and improvements to the character handling routines to terminals have reduced greatly the cost of terminal i/o. The new version of the loader and the addition of table of contents to loader archives have speeded up loading of large programs by about a factor of 3. Improvements to the system data structures have decreased the time required for most system calls, increasing the amount of time available to user programs.
- A bug has been fixed in the C optimizer which caused incorrect code to be generated for the random number generator. A consequence of this is that the rand function in this version of the system produces different sequences. Fallout from this is the fact that the secret-mail facility encryption scheme is subtly changed by this bug fix. All users should re-enroll in secretmail using enroll; messages sent before re-enrolling can be read using /usr/old/bin/xget.

Section 1

This is a brief summary of the changes which have been made in section 1 of the manual. In general, a number of system maintenance commands have been moved to section 8 of the manual as they are of little interest to the average user. In addition, to reduce the bulk of the manual, a number of very-closely related commands have been merged together onto a single manual page; thus **ccat**, **compact** and **uncompact**, which had three separate pages in the last edition of the manual, now have only one, and the line printer commands **lpr, lpq** and **lprm** have been placed on a single page.

ac Now appears in section 8.

adb Now correctly initializes the maps when debugging the system. Control characters are now printed using the ^x convention when the C output format is used. The increment for dot in the **p** format is now (correctly) 4. A previously undocumented command is **$?**, giving the output of $r, preceded by the process id and signal stopping the process. The bug which cause the running process to always stop at location 2 when first run has been fixed.

analyze Now appears in section 8.

ar Now creates a new archive format, in which the critical information is stored in ASCII, and thus portable between the VAX, PDP−11, and other machines. See *old*(8) for information on dealing with the old format. The program *arcv*(8) converts old (binary format) archives to new format, in place. Beware, however: the format of **.o** files has changed, so archives of object files must be recompiled (see **cc** below).

arll Is now in **/usr/old/bin** see *old*(8).

arff Is now *arff*(8).

as A new option **−R** causes the assembler to make data segments read-only and shared; this replaces the cumbersome ":rofix" script used in the previous distribution. For very large assemblies there is an option **−J** which causes the assembler to use four-byte jumps to avoid "Branch too far diagnostics." Local labels have been added to the assembler defined by "*n*:" and referenced as "*n*f" or "*n*b"; see the assembler manual in volume 2c for details. All tables in the assembler have been made extensible so that there should be no problems with assembling enormous files. The assembler has also been sped up by 25-40%.

at A number of security problems with the *at* command have been fixed.

biff Is a new command which can enable immediate notification when mail arrives; if you give the command "biff y" then the first few lines of each piece of mail sent to you is printed on your terminal. This is a (different flavor) replacement for the MAIL variable of **sh** or the mail variable of **csh**.

cat New options **−n** numbers lines; **−s** crushes out multiple blank lines, replacing **ssp**; **−v** prints control characters in the ^x format. The commands *num*(1) and *see*(1) now just invoke *cat*.

cc A newer version of the C compiler supports more strongly typed structures. Field names need not appear in only one structure; rather the compiler insists on correct typing of pointers accessing structures in contexts where this would cause ambiguity, and producing a warning in other cases when the pointer does not have the correct type to access the field name.

A new type **void** may be used in a declaration or a cast to indicate that there is no result, or that a value is to be discarded.

The compiler has been changed to accept arbitrary-length names. For maximum portability it is better to limit names to 8 characters, 7 characters for externals, but this is no longer required. The longer names bring a new .o format, incorporating a string table, and consequently a large number of programs are changed. The critical ones exist in "old" versions to ease conversion pains.

A number of bugs in the compiler have been fixed. To avoid remaining bugs in the code generator, the declaration of register char and register short variables are currently being compiled as though the word register were not present. (This does not affect pointer variables.) This blindness is allowed in the semantics of the language, and is not as bad as it might seem at first. It is sufficiently difficult to maintain the semantics of C in operating on such quantities that the code is often better when they are not placed in registers. This applies to unsigned char and unsigned short variables also.

cd The *cd* command, in *csh* is now affected by a *cdpath* variable which works similarly to the command search *path* variable; see *csh*(1).

checknr A new command which checks macro bracketing structure in *nroff* or *troff* input.

chgrp Has been moved to section 8.

chown Has been moved to section 8.

cifplot Produces raster plots from CIF for integrated circuit designers.

clri Has been moved to section 8.

colrm Has had several bugs fixed.

cp It, along with *mv* now take −i flags which cause them to ask before overwriting an existing file; thus cp can be usefully *alias*ed to "cp −i" when using *csh*(1) to provide a safer environment.

csh Has had a number of changes; most importantly, the shell now supports "job control", allowing arbitration of a terminal between several active groups of processes. See *newcsh*(1) for a short description of new *csh* features. Users of *csh* automatically use the new teletype driver summarized in *newtty*(4). (An old version of csh, corresponding to that of the Third Distribution remains available as oldcsh to ease conversion to the new shell.)

ctags Now recognizes FORTRAN and Pascal routine definitions.

cu Is now functional on VADIC dial-out hardware with the necessary software to record outgoing calls; see *rv*(4).

cxref Is now the −x option to ctags.

date The internal date representation (maintained in the hardware) has been changed to be more compatible with the standard DEC system VMS, although VMS still poses problems (see BUGS in *date*(1)).

dcheck Is now in section 8.

dd Options have been added to do blocking and unblocking of tapes without converting to and from EBCDIC.

ddate Is obsoleted by the new dump program, which lives in section 8.

df The output format is greatly improved. It is now also possible to say "df directory" to get *df* to report the free space available on the file system containing *directory*.

diction Is a new program which finds bad or wordy diction.

diff Has new options to give lines of context and to create a merged conditional file with "#ifdef" controls in it. The command *diffdir* has been subsumed into *diff* and an option has been added to perform recursive directory differences.

diffdir Is now part of *diff*.

dmesg Is now in section 8.

dump Is now in section 8.

dumpdir Is now in section 8.

echo The echo builtin to csh now is the same as the standard version 7 echo. The special character sequences "\c" and "\n" no longer have their special meaning; instead the −n option is accepted.

efl Is a new FORTRAN preprocessor, providing C-like features.

error Is a new program which consumes error message output from language and other processors, analyzes them and (optionally) places appropriate diagnostics back into source files near the point of error.

ex A number of bug fixes and a few new features are available. See the incremental change report in volume 2c or read the file */usr/news/ex*.

expand Now has a companion *unexpand* command, reversing its effect.

flcopy Is now in section 8.

from Now takes a −s option, printing only headers from the sender who is the next argument.

gets Is obsolete; use "$<" of *csh*(1) or *read* of *sh*(1) instead.

grep The −i option now works with *fgrep*; several subtle bugs have been fixed in its algorithm.

iostat Now has a single output format which includes character i/o rates, seeks and transfers per second on each disk.

kill Is now built-in to *csh* and take job names "%..." as arguments. This eliminates much of the need to know process id's when using *kill*. It also takes symbolic as well as numeric signal names; do "kill −l" (lower case letter l as in lark) to see a list.

ld Has been speeded up substantially; several new options have been added providing incremental loading (−A), primitive load maps (−M) and symbol def/use tracing (−y). The loader now supports archives with table of contents, and expects all archives it encounters to have them (or it complains); see *ranlib*(1).

lisp The lisp interpreter has begun to support new "string" and "hunk" data types. A user must now type vertical bars to delimit atoms with embedded white space, as double quotes (") now delimit strings. The interpreter has benefitted from in-line expansion of frequently called small routines, and will interpret list structure 20% more quickly than before. A perusal of the index to the lisp manual in Volume 2c will show more new functions which Franz has in common with MACLISP.

liszt The quality of the code produced by *liszt* is substantially better than the previous version; *liszt* also now provides macros for UCI lisp compatibility as well as Maclisp compatibility, and a facility for constructing a cross-reference with *lxref*(1).

ln Now subsumes *lnall* when given multiple arguments.

ls Has new options −R for a directory-recursive version, −F to highlight directories and executable files, and now indicates *mpx*(2) files in long format outputs with first output character **m**.

lxref Is a new lisp cross referencing program.

mail All mail is now forwarded through the facilities of *delivermail*(8) using the mail aliases data bases in the file */usr/lib/aliases* whose format is described in *aliases*(5). Gateways to UUCP, ARPA and Berknet machines are provided, with normal naming conventions, i.e.: "name@site" for the ARPANET, "machine!name" for

UUCP, and "machine:name" for the Berknet. Personal distribution lists created and placed in the file *.mailrc* are now expanded in sent mail so that the recipient can reply to members of personal aliases. Systemwide aliases contained in the "aliases" file are applied to *all* incoming mail; this allows correct functioning when reply is made to a mailing list, which was not expanded to its constituents when the mail was first sent.

make Now understands about Pascal files, running *pc* using the options PCFLAGS. A number of limitations have been removed to allow very large programs to be controlled.

makewhatis
Has been replaced by *catman*(8).

man Now interfaces with a preformatted version of the manual; this allows the manual to be presented much more quickly. The preformatted manual is maintained automatically; when pages are changed and then requested they are reformatted. The *catman*(8) command can be used to reformat all manual pages whose preformatted sections are out of date.

mkfs Is now in section 8.

mknod Is now in section 8.

more Has been improved, providing more commands and has also been changed to be more (sic) similar to *ex*(1). A new interface *page* is like *more*, but clears the screen before presenting each pageful.

mount Is now in section 8.

msgs Now provides options for saving messages in files and responding to messages. There is also an option (**-p**) to arrange that long messages be piped through *more*(1).

ncheck Is now in section 8.

newaliases Rebuilds the random access version of the mail forwarding database. After you edit the file /usr/lib/aliases which contains the forwarding information, you must run **newaliases**.

nm No longer prints symbols destined for *sdb*(1) by default. Instead, when the **-a** option is given, they are printed symbolically (rather than in octal as before.)

pc Is a new Pascal compiler, which allows separate compilation and mixing of Pascal code with routines written in C and FORTRAN. It is possible to use *sdb*(1) with Pascal routines, and all standard language features, including procedure and function parameters are implemented.

pi Has had some options changed so that they are similar to *pc*'s options (since the latter is constrained by the demands of compatibility with *cc*(1) and *ld*(1)).

pr Now has an option **-f** to use form-feeds in the output rather than multiple blank lines to eject pages. This can be used for greater top-of-form alignment reliability on unreliable printers or to reduce bulk of the output.

prof Now suppresses routines which were never called and which accumulated no time. An option to sort the output by number of calls has been provided. *Prof* can also combine the statistics from a number of saved monitor data files in a single output report, producing a new summary data file.

ps The formats for ps have been improved to include more information about job state in all output formats, and cpu and memory consumption by jobs (in the **u** and **v** output formats.)

pwd Is superseded for *csh*(1) users by *dirs*, which is built-in and significantly faster.

ranlib Is used to create loader table-of-contents for loader archives. The loader insists that such table-of-contents exist or it complains, as loading is very slow without these indices.

renice Is now in section 8.

restor Is now in section 8.

sdb Now knows about Pascal as well as C and FORTRAN 77.

spice Is no longer part of this distribution, and must be obtained separately.

stty Now has options to control the features of the new tty driver, summarized in *newtty*(4) and fully described in *tty*(4). In particular, it is often desirable, when using the new driver with *csh*(1) to place the command "stty crt" in the *.login* file to cause control characters to echo legibly and intelligent erasing of characters from the screen to take place except at very low speed.

style Is a new command, which analzes the style in which a paper is written.

sync Is now in section 8.

tail Now has a **−f** option which follows a file as it grows, replacing **tra**, and a **−r** option, which prints out the last part of a file backwards, line by line.

tar Now normally saves and (optionally) restores modes of directories. It also uses the raw tape drive by default, so the file option is not normally needed.

tp Has been fixed to work with the tape drive as it should.

tra Has been deleted; use "tail −f" instead.

tset Has a more general scheme for mapping non-hardwired terminals to types. The command used in startup files to set the terminal type can now be simpler; see *tset*(1) for details.

ul Subsumes the old "iul" command as the option **−i**.

units Currency tables have been updated.

uuencode Along with *uudecode* provide the ability to mail binary files through *uucp* or other mail facilities.

vmstat The format has been changed to include statistics on the rate of system calls, context switches and device interrupts. The format also includes information about page-cache hits (pages of programs which recently exited which were found lying in the free list.) See *vmstat*(8) for more details.

vpac Is now in section 8.

wc Now will estimate transmission times over communications networks.

yes A new command, which is excessively obsequious.

Section 2.

The important changes to section 2 define the mechanisms needed to support the job control facilities of *csh*(2) defining new system calls **getpgrp**, **killpg**, **setpgrp** and providing a version of the signal system call interface with clean semantics; this is described in *sigsys*(2). For a general introduction to the new job control facilities from a programming standpoint see *jobs*(3).

The system calls in section 2 have been identified as to origin: those in section 2V are peculiar to the VAX Virtual Memory version of the system. Some of these (e.g. **vread** and **vwrite** which are likely to change soon are so marked.) Calls resulting from the job control facilities are in subsection 2J.

getpgrp Is a new system call, which returns the process group of a specified process.

ioctl There is a new *ioctl* call FIONREAD which returns the number of characters immediately available from any readable file descriptor, most usefully terminals.

killpg Kills all processes in a process group

reboot Is used to halt the processor or initiate automatic reboots.

setpgrp Changes the process group of a specified process.

sigsys There are more signals, needed by the new job control and limit facilities. This is the new signal interface, which can be setup to handle signals reliably, blocking them off when they occur so that they can't happen again before they can be reset and also allowing them to be blocked, but not lost if they occur (protecting critical sections.)

vadvise Allows processes with very abnormal paging behavior to advise the system so that it can have a better chance of dealing with their virtual memory needs.

vhangup Is used to simulate hangups on hard-wired ports and to prevent other users processes from scribbling on a terminal after they log off.

vlimit Is used to provide the resource limiting *limit* facility of *csh*(1) which allows the amount of CPU time or memory used by any process to be controlled, as well as the maximum size of a file created, maximum size of a core file created, etc.

vswapon Is used to inform the system that it should use the interleaved portions of the paging area, since at boot it normally uses only that portion of the paging area adjacent to the root file system.

vtimes Provides additional virtual memory information as well as the traditional information provided by **times**.

wait3 Provides options which allow return of vtimes like information about the virtual memory utilization of terminated processes as well as allowing one to determine if any children have changed state without blocking if none have.

Section 3

The major change to the standardly available library routines is in the standard i/o library handling of the standard output. The standard output is now line buffered by default when it is directed to a terminal. In order that old programs which do terminal i/o continue to work (in almost every case) without change, the standard i/o library flushes a partial line in the standard output buffer whenever a read is made from the standard input.

curses Is a new library of routines to do terminal independent cursor motions. This is a higher level set of routines based on the low level *termcap*(3).

dbm Has been changed to use 1024 byte blocks internally. If you have old data files and recompile a program which uses **dbm** on them, you must recreate the files in the new format.

getfsent Is a new package of routines for locating entries in the file system information file, **/etc/fstab**.

jobs Is a descriptive page on use of the new job control facilities.

malloc Hasn't been changed yet (unfortunately), but we have discovered that it has extremely poor behavior if used to allocate a large number of blocks in a large virtual address space, as it tends to page fault on a large number of pages whenever you allocate more storage when the free list is nearly empty. Very large applications should write their own free storage management routines based on *sbrk*(2) until a new version of **malloc** is ready.

nlist The namelist has been changed to embed a string table. The definition of the namelist in <a.out.h> now includes a **union** construction which the C compiler will not let you initialize. Thus if you wish to use **nlist** you should use <nlist.h> instead of <a.out.h> as it doesn't include a **union** construction and initializations of **struct nlist** objects are thus possible.

rand Generates different sequences, due to a bug fix in the C optimizer (i.e. the old routine was flaked out due to the bug.)

regex Is a new library routine which allows compilation and execution of regular expressions.

stdio Has been changed to have line buffered standard output to terminals by default (as described above).

sigset Is the user-level interface to the new signal facilities, providing reliable and convenient handling of signals.

string The routines **strcatn strcpyn** and **strcmpn** are now decreed to be spelled **strncat, strncpy** and **strncmp** in the official version (to be compatible, e.g., with PDP-11 UNIX), but both versions still exist (temporarily).

termcap Is the new name for the previously misnamed **termlib** library.

Section 4

The system now supports additional i/o disciplines and devices:

bk A "line discipline" which allows efficient, high-speed, input for uploading of programs and networking over asynchronous communications lines.

dh Abel DMAX-11 dh emulators.

hp Includes RM-05 support (including the SI MASSBUS disks).

lp A driver for a standard lineprinter interface.

rv Racal/vadic call units (via user-software simulation of a dn-11).

tm Support for UNIBUS TM/11 tape drives (e.g. Kennedy transport on a Emulex controller.)

tty Includes a new tty driver which is standard for users of *csh*(1) providing additional functionality.

up A driver for a EMULEX SC-11 or SC-21 UNIBUS disk controller with storage module (e.g. Ampex 9300) disk drives.

va A varian raster plotter.

Section 5

The changes impacting most users are the new archive and object file formats. The new object file format requires recompilation of all .o files, as the new format includes a string table for names while the old table had only fixed space for 8 character names. The new archive format is portable, and can be easily converted to using *arcv*(8).

a.out Now includes a string table. New macros are provided for dealing with executable files, and the names of the headers and the include files have been changed. You should read this section if you have programs which work with the bits in these files.

acct The system now maintains all the information fields in this file.

aliases This file gives forwarding information for mail. The **delivermail** utility, which is invoked by the mail handlers, uses this information to forward mail and to implement distribution groups. The program **newaliases** must be run whenever this file is edited, since the **delivermail** program actually uses a binary version of this file.

ar Now uses a portable format. Old formats may be converted using *arcv*(8), although converting old loader archives is pointless, since the **.o** files must be regenerated anyways.

core The maximum size of a core file can now be limited with *limit* of *csh*(1) or by calling the system call **vlimit**.

environ Has acquired a USER variable with the login name of the user.

fstab Is a new data structure, kept in the file **/etc/fstab**; it greatly simplifies the system startup script **/etc/rc**, the interface of the dump program, and is a central figure in the automatic reboot procedure. It is possible to reconfigure the disk drives simply by changing this data base. No file system dependent commands need appear in **/etc/rc**.

stab Describes the symbol table entries for the symbolic debugger which appear in **a.out** files.

termcap Has been changed to describe even more terminals and functions.

ttytype The standard naming for terminals in this file is now to give long names; thus dialups should be described as "dialup", not as "sd" like before.

vfont There are now rotated versions of all the fonts for use on output devices which run 11 by 8.5 inch (sideways) paper. The Hershey fonts have been extensively edited to fix problems and fill in missing and mangled characters.

Section 6

There are a number of new games, including the popular ZORK game as well as more mindless and more popular games, such as **snake**, and human-humiliating games such as **boggle**.

aardvark Is one of several dungeon-exploration games.

boggle Is an implementation of the Parker-Brothers game, in which you and the computer look for words and you are humiliated by the computer.

chess Is the PDP-11 chess program running in compatibility mode.

cribbage Is a respectably good cribbage player.

fish Plays Go Fish.

fortune Has been refreshed with new and interesting quotations and fancy-pants options.

mille Is an implementation of the French card game, Mille-Bourne.

monop Is the game of monopoly.

mpu Is another exploration game.

snake Is a mindless game in which the computer chases you around the screen and, usually, ends up eating you.

zork Is the classic successor to adventure for die-hard explorers.

Section 7

hier Has been updated to reflect changes in this distribution.

Section 8

There is a major organizational change in section 8: the maintenance commands have been moved here from what was section 1m (which no longer exists) and have also (almost all) been moved into the directory **/etc.**

The procedures for system reboot have been substantially changed. The system now has provisions for automatic reboot and file system repair after a crash. The system has been changed so that the disks are kept more consistent, so that the repair program can easily tell whether the inconsistencies it finds are simple and reasonable to fix. If it finds unexpected inconsistencies, then the automatic reboot will fail and the new *fsck* interactive repair program can be run.

adduser There is now a shell procedure **vipw** which locks the password file for editing, so that you can't get messed up by people changing their password while you are adding users.

arcv Converts to the new archive format (as discussed previously.)

bproc Is defunct; see **reboot** which replaces it.

catman Is a new command, which creates the preformatted version of the on-line manual from the nroff source for the manual pages. It also creates the data base for *whatis*(1) **/usr/lib/whatis**, replacing **makewhatis.**

crash At the point of a system crash, the system now will automatically reboot itself if the auto-reboot switch on the machine is set, print the registers and the top few locations of the stack, and will then normally "preen" the file systems and come up multiple user again. The system does not (yet) automatically dump core to disk when a crash occurs, so the procedures described in **crash** are still used to take a crash dump; in order to get one, you have to be running with auto-reboot disabled when the crash occurs. Note that you must now use **dd** to copy crash tapes, as the tapes are blocked "bs=20b".

cron Now runs the executed commands as **root**; before it ran them as **daemon** which kept a number of reasonable **crontab** entries from working properly.

dcheck Is obsoleted, in all normal usage, by *fsck*(8).

delivermail

Is a new daemon which acts as a postman and routes mail destined for foreign networks. It also interprets distribution, forwarding and alias lists from the file **/usr/lib/aliases**, which replaces the function only partially performed previously in **/usr/lib/Mail.rc.**

dump The dump program now is much more robust. It restarts after bad tapes are encountered, and will keep a selected group of operators informed about what it is doing. A **dumpdates** file which is kept in ASCII replaces the old binary format **ddates** file, making it easy to edit. **Dump** uses the new **fstab** file system table which records the desired frequency of dumps, and a new option **w** will tell the operators what needs to be dumped.

flcopy Has a new option **−h** to start halfway through, writing a disk from an existing floppy file without reading the disk.

fsck Is a new, intelligent, interactive file system check program. It is normally run automatically at reboot with the option **-p** to preen all the file systems and fix normal minor incomplete operations. If any unexpected problems arise, the automatic reboot will fail and **fsck** should be run manually. There is a document explaining the use of fsck and all the errors which can occur in standard UNIX systems. Most errors are prohibited from happening in this version of the system unless there is hardware or software failure; **fsck** was designed with adversity in mind to deal with these exceptional cases and is very easy to use to fix up addled file systems.

getty Is the new name for the old **getty.vm**.

halt Halts the processor cleanly when no reboot is desired, regardless of the setting of the auto-reboot switch on the machine.

icheck Is largely replaced by **fsck**.

init Now understands how to initiate an automatic reboot. When passed appropriate information as parameter at boot time, it will immediately run **/etc/rc** with parameter **autoboot**, causing it to run a disk check. If **/etc/rc** exits with a non-zero exit status, the reboot we be considered a failure, and a single user shell is given to the console as before. The form of **/etc/rc** is such that interrupting the automatic reboot also yields a single-user shell on the console. **Init** has also been fixed so that "**kill 1**" attempting to bring the system down single user will not hang irretrievably if some processes refuse to die because of hardware or software malfunction. Instead, **init** gives a single user shell and a warning that something is amiss.

mkfs It is critical that all newly created file systems have **lost+found** directories as created by...

mklost+found
Which creates empty directories into which **fsck** places detached files which it reconnects.

mount Now takes an option **-a** to mount all normally mounted file systems as specified in **/etc/fstab**.

ncheck Is obsoleted in normal usage by **fsck**.

old Is a front for a number of old commands dealing with old formats for **.o** files and archives. These commands are kept in **/usr/old** and its subdirectories and can be used to keep your sanity until you have time to convert your programs to the new formats.

reboot Describes the new reboot procedures, based largely on the automatic reboot facility. Also describes the cold-start and emergency-start procedures (which came from the old **bproc** manual page.)

sticky The system has been changed so that text pages of processes which have exited are retained in the free page pool in core (at least for a while), so that they will not have to be read from the disk again if they are needed quickly. This tends to make heavily used commands behave much as **sticky** was intended to make them behave on swap based systems.

swapon The system supports paging interleaved across disks and controllers, but bootstraps running only on one disk. This command makes the additional pieces of the paging area available.

umount Now takes a **-a** option, to attempt to unmount all file systems mentioned in **fstab**.

vipw Is a new script which lock the password file while editing it. If the password file is being changed (e.g. by the *passwd*(1) command), then **vipw** complains, and avoid the race condition.

Department of Psychology

RG:rc

University of Queensland
St. Lucia
Queensland
Australia 4067
Telephone: (07) 377 1111
Telex: UNIVQLD AA40315
Cables: Brisbane University

21 January 1981

Peter Ivanov
Dept. of Computer Science
Electrical Engineering
P.O. Box 1
Kensington NSW 2033

Dear Peter,

I am enclosing copies of a BBN C/70 glossy and a tech report on Steve Johnson's C machine just in case you don't have either. They might be appropriate for AUUGN. An article comparing the various C/UNIX hardware alternatives might be quite interesting.

Yours faithfully

Ross Gayler

BBN Computer

BBN Computer
Corporation
33 Moulton Street
Cambridge, MA 02238
617-491-1850

13 November 1980

Ross Gayler
University of Queensland
Department of Psychology
St. Lucia, Brisbane 4067
Australia

Dear Mr. Gayler:

BBN Computer's C/70 has been DESIGNED to execute the C language. Its operation codes, addresses and data types are matched to those used in the C language. It uses 8 MAPs and 1024 registers to achieve extremely fast context switching. These features alone allow the C/70 to execute sample C routines as fast as the 11/70, yet contain significantly less hardware, and thus cost less to build.

Here are some examples; the C/70 executes a function call in 6.2 microseconds, compared to 17.1 on the 11/70; a system call in 270 microseconds, compared to 390 on the 11/70. Disk I/O is comparable: the C/70 executes random reads in 15 ms per block compared to 14 for the 11/70.

A C/70 with 256Kb of main memory, 8Kw of microprogrammable memory, 32 asynchronous terminal ports and an 80-Mb Storage Module Disk has a list price of $56,700.

I hope the enclosed material answers your questions. If you have more, please don't hesitate to call me or your local salesman.

Yours sincerely,

BBN COMPUTER CORPORATION

Martin Oakes

Martin Oakes
Director of Marketing

MO/js
Enclosure

A Subsidiary of
Bolt Beranek and Newman Inc

RMIT

DEPARTMENT OF COMPUTING

Faculty of Applied Science

Royal Melbourne
Institute of
Technology

GPO Box 2476V
Melbourne Vic 3001
Telephone 3452822
Telegraphic Address
'Meltech' Melbourne
Telex AA36406

MR:kh

February 12, 1981

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

Dear Peter,

Thank you for your letter regarding UNIX and the HP. I am planning to proceed along two fronts, these being the software tools with a UNIX style shell and the port of UNIX to the HP.

I have already started work on the tools with the aim of getting most of them running on the HP and maybe the Cyber. Our license for UNIX has not yet been processed so work on this is planned for the future.

Thanks once again for all the help you have given me and if you hear of anyone who is interested in the software tools I would be very interested in talking to them.

Yours sincerely,

Mark Ross

Mr. M. Ross,
Lecturer, Department of Computing

THE UNIVERSITY OF NEW SOUTH WALES

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

February 20, 1981

SCHOOL OF ELECTRICAL ENGINEERING

Ross Gayler
Department of Psychology
University of Queensland
St. Lucia
Brisbane 4067
Queensland
AUSTRALIA

Dear Ross,

Well, by now you no doubt know that I received your subscription, because you have got volume 3 number 1. You also know that I straightened out John Lions about the Queensland people and the manuals, with a little help from Dr. Newell. And probably by now your university has already made some decisions about whether 'to VAX or not to VAX'.

All I can say is "doesn't time fly when your having fun!".

Seriously though, I am sorry I have not replied earlier but there has just been so much happening around here that now is the first time I have had in two months to sit down and plough through my mail.

I received a letter from Rick Stevenson telling me of his exploits with level 7. He also asked if I new of someone in this country with a copy of the Vrije pascal for level 7. Well I sent him a copy of that and have asked him to send me a copy of his V7 changes.

In case you dont know, Johnny Lions has despatched the following manuals to Queensland:

1. To the Department of Computer Science
   - two level 7 manual sets
   - one beginners manual
   - one beginners companion manual
2. To each of Electrical Engineering, Human Movement Studies and Mining and Metallurgy
   - one level 7 manual set

He has sent no source code manuals as these are now distributed by Bell Labs. I have enclosed a few J.L.'s form letters telling people where to apply for a copy.

I have enclosed a full list of all AUUGN's past and present subscribers and I would have supplied you with a set of labels to assist you in circulating your questionnaire, but I feel I have missed the boat. Should you still require these, call me at work (02-662-3781) or at home (02-349-6256) and I will send them that very day (even Christmas day).

Sorry once again for the delay.

Yours sincerely,

Peter Ivanov

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

Department of Psychology

University of Queensland
St. Lucia
Queensland
Australia 4067
Telephone: (07) 377 1111
Telex: UNIVQLD AA40315
Cables: Brisbane University

RG:js

11 February, 1981.

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

Dear Peter,

The VMS/UNIX fracas is not yet over here, in fact the most interesting part is yet to come. Future attractions include the release of the report(s) of the technical subcommittee which did not reach a consensus, a public forum of potential users, and the climactic decision of the Computing Policy Committee. It promises to be more entertaining than the Festival of Sydney and Moomba combined.

Could you get me some technical details on the BFI-II and the Motorola 6809 C compiler? We're getting a new building and I'm responsible for some of the interior decoration. I rather fancy having a room or two done out in wall to wall BFI-II's. The 6809 C compiler is relevant not only to the BFI-II as we have 6800 based microcomputer systems used for experimental control which have been designed so that the 6800 can be replaced by a 6809 with minimal effort.

As you probably know, Chemical Engineering here are getting an 11/44 system which will be running UNIX. Our equipment subcommittee (no prizes for guessing the identity of one member) has just approved the money for us to replace our 11/34 processor with an 11/44. We probably won't get it for 6 to 8 months though.

Rick Stevenson sends his apologies for not sending you a tape of his level 7 mods. He is very dubious that a tape written by his controller would be readable elsewhere. I think he said something about the CRC being bit reversed and taken modulo 1. He will send the tape to you, via Chris Rowles, either when he gets a new controller or when he can get the stuff to our machine and onto the TS-11 (1600 bpi).

Yours sincerely,

Ross Gayler

THE UNIVERSITY OF ADELAIDE

DEPARTMENT OF MECHANICAL ENGINEERING
Chairman: Dr. J.M. Pickles
Professor: R.E. Luxton

In reply please quote: RG/HJK

ADELAIDE,
SOUTH AUSTRALIA 5001
Tel.: (08) 223-4333
Telex: UNIVAD AA89141

16th January, 1981.

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

Dear Peter,

Thank you for your prompt reply to my enquiries re the Australian UNIX User Group Newsletter at the users conference last year. Enclosed please find a cheque for $48.00 to cover the purchase of back issues of Vols. I & II of AUUGN and a subscription to Vol. III. We would prefer copies on bond paper but will accept micro-fiche if necessary.

At present it is almost certain that Electrical Engineering will be setting up a UNIX system during the course of the year and there is a strong possibility that we will set one up in this department as well. I have had some contact with John Field at DMS here in Adelaide who is planning on running MINI-UNIX soon. As far as I know these are the only users here in South Australia but I would appreciate any information you might have regarding others.

Yours sincerely,

Rod Curtin.

ROD CURTIN

הטכניון — מכון טכנולוגי לישראל
TECHNION — ISRAEL INSTITUTE OF TECHNOLOGY

הפקולטה למדעי המחשב
DEPARTMENT OF COMPUTER SCIENCE

December 25, 1980

Dr. Piers Lauder
Basser Dept. of Computer Science
Sydney University
Sydney - AUSTRALIA

Dear Dr. Lauder,

I was most interested in reading your paper about "Share Scheduling Works", in the recent AUUGM.

We here at the Technion have a Vax 11/780 with 1MB memory, two RM03's disk drives, a TU77 tape drive, a DUP-11 synchronous communication adapter, and three DZ-11 line drivers for a total of 24 terminal lines operating standard terminals at 9600 baud. We are operating the system using the Third Software Distribution of the Berkeley Virtual VAX -11 Version. The system supports our research staff and about 200 students working with vi, pi, asm.

I was very pleased to find out that your system is able to support 75 users simultaneously. We have found a delay when operating under full load and are thus somewhat afraid to add more terminals. I would be most interested in knowing what is the equipment configuration at your location? What is the load mix?

Is it possible to obtain a copy of the programs that you wrote to implement the Share Scheduling Algorithm? If so how can I obtain a copy of these programs?

Thank you for your time and consideration.

Sincerely yours,

Shlomo Goldberg

Shlomo Goldberg,
Computer Science Dept.

cc: Peter Ivanov, AUUGN

CABLES: TECHNION, HAIFA 32 000 / TELEPHONE: 225111 :טלפון / 32 000 טכניון, חיפה :כתובת למברקים

Dr. Shlomo Goldberg
Computer Science Department
Technicon City
Haifa 32000
Israel

Department of Computer Science
University of Sydney
Sydney
New South Wales, 2006
Australia

6th January, 1981

Dear Dr. Goldberg,

Thanks for the interest you expressed in my paper on the Share Scheduler. I see you are also interested in driving your CPU further! Perhaps the best I can do for you is to describe our system in a little more detail, and some of the steps we have taken to increase the number of supportable users. I also enclose a 600 foot tape in "tar" format, with a few programs that might be of some help to you.

The Vax in the Department of Computer Science is configured as follows:

Floating point accelerator
3.25 Mbytes memory
3 * RM03 68Mbyte Disk drives
4 * KMC11-A Unibus I/O processors
14 DZ11 8-line multiplexors

On an average day there will be about 70 users logged in, most running pi and em (a modified version of "ed" from QMC, London). About 20 of these will be running f77 instead of pi, and a few will be using SYD11 (a pdp-11 simulator). There will be few academics using the system during student sessions, as their "share" is usually too small! The average user consumes about 40 Kb of memory. Typical response times with this load are a second for editor commands, and about 20 seconds for a "pi" compilation of a 150 line program.

You can see that we are not short of memory, and in fact the system hardly ever swaps processes at all. Our disk activity averages 50 transfers/second (out of an theoretical maximum of approximately 150 transfers/second), so we are not disk bound. We are not running the Berkeley System, so your disk figures will probably be higher due to paging traffic as well.

We are, in fact, very short of CPU time, and nearly all our efforts have been directed to minimising system CPU overheads.

Our first concern was the very high overheads of the Vax associated with terminal interrupts. To this end we installed KMC11 support for the DZ11s. Our measurements indicate a saving of approximately 5% CPU time per installed KMC11 (each of which supports 32 DZ11 lines). Each interrupt on the Vax Unibus take 13us of CPU time to happen, and add to that the service time of the driver. The Vax can be expected to support at most 10 9600bd lines with 100% CPU degradation (DEC figures) (or 10,000 interrupts/second). The KMC11, by doing block transfers, considerably reduces the number of interrupts (approximately by a factor of 10). Also, may I suggest that you reduce the speed of as many terminals as possible. We find students to be quite happy at 2400bd.

We then did some profiling of the system code and found many areas where improvements could be made. (Many of these are in the latest Berkeley release also). In particular we hashed disk block numbers in the buffer cache reducing the search time. We made many of the latest Berkeley changes independently, most to do with replacing CPU expensive "calls" instructions with in-line code, or "jsb"s. The best way to get these benefits would be to get the latest Berkeley updates (which also include a Pascal compiler!).

The Share system depends on our AUSAM resource control modifications to

Unix developed over the last 4 years. To control users on a per-user basis requires the presence in the kernel of some sort of per-user structures (lacking in most Unix distributions). I have included in the tape as much source that could be useful to you as possible to do with the Share scheduler and per-user structures. You will have to install these structures in your system to support the scheduler in some way. There is a chance that we will be doing this work ourselves for the Berkeley distribution in the near future, but no promises.

In managing about 1400 users, we also discovered a need to have a random access "passwd" file. This was done under the "AUSAM" changes mentioned above. I haven't included any source for this, but if you should feel you might like to have it (before any official distribution for the Berkeley system), please let me know.

Good luck!

Yours sincerely,

Piers Lauder

Department of Computer Science

Head of Department
Dr. L. H. Reeker
B.A.(Yale), Ph.D.(Carnegie-Mellon)

University of Queensland
St. Lucia
Queensland
Australia 4067
Telephone: (07) 3773952

28th November, 1980.

Peter Ivanov,
AUUGN Editor,
Dept. of Computer Science,
University of New South Wales,
P.O. Box 1,
KENSINGTON. N.S.W. 2033.

Dear Peter,

Please find enclosed payment for a subscription to AUUGN, and copies of all back issues. Ross Gayler informs me that $34 will cover the cost, but let me know if it isn't enough.

I am currently modifying V7 UNIX to put in most of the UNSW mods, and a few of the local ones - with an emphasis on 11's without separate I and D space (we have an 11/23 and an 11/34). It will run with both BIG-UNIX and MAPPED-BUFFERS - our system currently has 40 buffers and contains RK05, RP02, RX01, TM11, LP11 and DH11 drivers! There is still a fair amount of work to be done (mainly on utilities), however I am sending a copy of the system in its current state to Chris Rowles as soon as our tape controller gets better.

We also have a few of the larger V7 utilities working (grudgingly) on our 11/34 - awk, portable C compiler etc.

Yours faithfully,

R. Stevenson

Rick Stevenson.

P.S. Do you know of anyone in Australia who has Vritje Pascal for V7 UNIX?

THE UNIVERSITY OF NEW SOUTH WALES

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

February 20, 1981

SCHOOL OF ELECTRICAL ENGINEERING

Rick Stevenson
Department of Computer Science
University of Queensland
St. Lucia
Queensland 4067
AUSTRALIA

Dear Rick,

Thanks for the subscription, but I dont know how Ross arrived at the price of $34-00. The correct sum should have been $48-00, but dont worry, I will come back to what you can do for us to make up for the extra $14-00.

Enclosed you will find a copy of the latest Vrije pascal distribution for UNIX level 7. The tape original tape we have was in TAR(1) format with a block size of 512, written at 1600 BPI, but I was not sure that a TM11/TU10 would read it. Actually I was pretty sure it would not, but you can correct me if I am wrong.

So what you will find in the package are:

- A tape written with TAR using a block size of 20b at 800 BPI.
- A printout showing (at the top) that the tape is readable here, and the output of a 'tar -tvb 20' command.

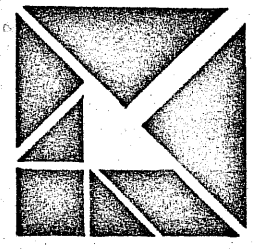
You mentioned in your letter that you have done considerable work on level 7 for small machines. If you wouldn't mind what you can do in return, for the extra $14-00 and the tape I have sent you, is send me a copy of your modified level 7 system on a new 2400ft mag tape.

You should also receive the newsletters you requested about the same time as this package. Keep up the good work and merry Christmas.

Yours sincerely,

Peter Ivanov

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



the Time-Machine Ltd.

ט"מ בע"מ

Haifa,12-11-1980

Mr. John Lions

University of New South Wales
Dept of Computer Science
Kensington 2033
Australia

Dear Sir,

There is some interest at our site for communication between PDP (+ Unix) and a CYBER 171 computer. From Vol 9 of Software practice and experience, I learn that your site solved the problem by developing your own U-200 emulator. I would like to know more about it and its availability,if there are some formalities involved,please let me know.

Looking forward to your reply,

Sincerely yours,

Gershon Shamay
Manager Software Development

Eder St. 49a, P.O.B. 72, Haifa, Israel. Phone: 04-246033.

THE UNIVERSITY OF NEW SOUTH WALES

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

PLEASE QUOTE

February 20, 1981

SCHOOL OF ELECTRICAL ENGINEERING

Gershon Shamay,
The Time-Machine Ltd,
Eder St. 49a,
P.O.B. 72,
Haifa,
ISRAEL.

Dear Gershon,

John Lions has asked me to reply to your letter of 12th November 1980, requesting information on the availability of the UNIX/U200 emulator package. As it happens, you are in luck. Earlier this year I sent two distribution tapes of the U200 package to Israel. They went to:

Emanuel Wind
Computer Science Department
TECHNION - Israel Institute of Technology
Technion City
Haifa
ISRAEL

Eran Gabber
Computer Laboratory
Department of Mathematical Statistics
Tel-Aviv University
Tel-Aviv
ISRAEL

You should be able to obtain a copy quickly from either of these two sources.

In your last letter you mentioned that you had level 7 UNIX running but with a few nasty restrictions. Rick Stevenson, whose address appears below, writes that he has a version of level 7 running on his 11/34. It uses the UNSW produced BIG-UNIX and MAPPED-BUFFER options from level 6 UNIX, and supports drivers for RK05, RP02, RX01, TU10, LP11 and DH11, and still has space left over for FORTY buffers. He also says he has a few of the larger level 7 utilities running (grudgingly) including AWK and the portable C compiler.

I think you will agree that it may be worth your while writing to him for some ideas. His address is:

Rick Stevenson,
Department of Computer Science,
University of Queensland,
St. Lucia,
Brisbane 4067,
Queensland,
AUSTRALIA.

Further on the subjects of upgrading to level 7 and the U200 system, we are (this very day) going over full time to level 7 on our PDP11/70 and have almost completed rewriting the 'Batch System' to run on level 6 or 7 UNIX without the extensive 'shared -data' kernel changes needed for the old U200 system. Should you be interested in obtaining this newer system, write to me again in a few months, allowing time for the bugs to be squashed, and I will send you a tape.

Until then, merry Christmas and a prosperous new year.

Yours sincerely,

Peter Ivanov

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

EUUG

EUROPEAN UNIX USER GROUP

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

7th January, 1981

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

Dear Peter,

Just finished reading AUUGN, Vol. II, No. VI (I never saw no.'s IV and V - I've muttered to Bruce about this), and I thought of so many little things that I wanted to mention that I'm going to re-read it, pen in hand. Get a coffee (or a beer) I'm not reknowned for my terseness!

Firstly, your policy statement. Can't say I like the idea of making the newsletters a public journal. It's something our group, along with both the American and Canadian decided against, and something that we've had to emphasise more than once to DECUS. With respect to bug fixes, I have been trying for ages to get more detail into them as you should see in our next issue. Our group now represents over 100 installations. More and more of these are non computer science, application oriented sites, and simply do not have the background to extrapolate from a 'general description of a modification' into the requisite C coding, particularly in the kernel.

I notice the balance sheet for your Canberra meeting, and for the first time realise fully that you do not have any form of committee. Beware of growing too fast - all the work will fall on your shoulders. We have found that having an ongoing bank account (at least) allows us, by adding a little on meeting registrations, to accumulate some funds. These then can be used to buy in stocks of useful items (BSTJ's, UNIX reference cards, UNIX manuals, magnetic tapes, etc.). Our members were initially worried that meetings would become too formal. This has not been the case. They are as 'free-form' as ever. Only behind the scenes, is life a bit more organised; with my cheap, headed paper I actually get replies from the large commercial concerns (even DEC!). I hope you take this in the spirit in which it is intended. I worry that, with growth, your group and your newsletter will collapse, as we thought we might, and as USENIX certainly did for almost two years.

Very interesting papers. Piers Lauder has obviously been busy! - when does he intend to move to Murray Hill? Trouble is, none of the papers give any idea of the availability, i.e. are they generally available, how much effort is required to install them, how much kernel space they require, do they need special system calls ..... Of course, maybe these items pall into insignificance when you work in a CS Department and have a VAX (bit of the green eyes there).

On the subject of cattle-dogs, particularly software ones, I have twice attempted this, once using free-form input, the second time using a simple restricted form. Whatever, the DB and input forms are not the problem: its getting the .....'s out there to put pen to paper. I've come to the opinion

Peter Ivanov 7th January, 1981

Cont./....

that no-one using UNIX, uses software! Regardless of this, my best wishes in any attempt.

With particular reference to the letter of Ken McDonnell of Monash Univ.; he is quite right in saying that possibilities should be investigated before serious or detailed design is attempted, but the thought of

"stored once per item .... set of indices ....
hierarchical classification of descriptors .... index terms ..."

horrifies me. Is the problem so large and the searches required so complex that RDB's need to be considered? Are people expected to be so particular in their search spec. and the DB data so all encompassing that on a successful hit the machine will pay the licence fee, suck the software down the network and install it! An entry consisting of 10 items, only 2 of which may be searched, one of these having no value, the other being 'chosen by the author', would seem an ideal way to start! Give me a system that will pick up all references to

"cheap reduced swapping"
or "extend address space"

and I will go away smiling. As an aside, I might suggest special hardware, we've just built some that dogs straight (no indexes) search on the equivalent of 10 Mb of text in 20 seconds, and its getting faster. In closing the subject of your cattle-dog I have actually got a constructive suggestion. In reading the letters section I came across a letter from TM in Israel requesting information about V7 for 11/34's. By the time that letter was published I had already sent him a copy of our V7 small machine package (as I have yourself). I also find from your newsletters that similar work is being done by 'Ross Gaylor' of Queensland. This goes to show (a) the need for a DB of software, (b) that the DB should not just contain existing software, but also packages intended, or currently being developed.

My 'grandpappy' always used to say that you were wasting your time slapping a man's back if you hadn't kicked him in the .... first! A bit excessive! I think I'm justified in saying well done. Another very good read. Keep it up. I look forward to the next.

Yours sincerely,

R.A. Mason

P.S. I'VE JUST HEARD THAT
INFOPRO SYSTEMS
56 CLIFFSIDE TRAIL
DENVILLE
NJ 07834
USA
ARE CURRENTLY COMPILING A 'UNIX SOFTWARE LIST' WHICH WILL BE MADE PUBLIC.
YOU MIGHT WANT TO MAKE CONTACT WITH THEM.

THE UNIVERSITY OF NEW SOUTH WALES

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

February 20, 1981

SCHOOL OF ELECTRICAL ENGINEERING

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

Dear Alan,

I have just finished reading your letter, and drinking several orange juices. I can see why you are not renowned for your terseness. Well neither am I.....

I shall treat your points in order.

I can understand your dismay at AUUGN becoming 'public' but I really do believe that this is the only way to go. First, I bet that EUUGN (if that's its new name) is read quite widely by non-license holders, no matter how careful you are in checking your readership's credentials. I personally have had feedback from machine manufacturers to articles they read in AUUGN well before the recent change in policy. Needless to say I have not been sending them copies! Also numerous enquiries from marketers of UNIX-like systems who have no UNIX license as such cannot be ignored for ever. As for the American and Canadian Newsletters not being public, I have not seen a US newsletter for years and my 'policy statement' is obtained in part from the 'statement of policy' to be found in every CUUGN I have received which states the CUUGN is 'to be considered a public journal' and 'authors take complete responsibility for their material'.

On the topic of bug fixes I agree that detail is what is needed but my view is that detail may be included as long as reams of surrounding source code and program operations are not exposed. Small bugs may be corrected very well with a short description which would mean nothing to someone without a UNIX system and large bug fixes (and these would be very large and complicated) may be handled by a description of the problem and naming of a contact to obtain the fix from. Its all a matter of scale.

Yes, we have no committee. No one. Except me that is. The user group as such started on this campus in 1977 as UNIX spread from its first conquest in Electrical Engineering to even permeate the bowels of the Library. We got together, and as more people used UNIX more people came to the meetings, until now people travel right across the country (rather a long way really) to attend a meeting. Future meetings are discussed at each meeting and institutions that offer to organise future

gatherings do all the work involved, with a little assistance from me in the provision of mailing lists etc. And so we have survived without a committee.

I know about the work falling on one person's shoulders and I truely feel that I have just about enough work to do. I have a University account (they do all the book keeping and auditing etc) and have managed to build up a tidy reserve which I have already used to pay for software, pay peoples conference fees in return for good summaries of what goes on, etc. etc., as well as the expenses of producing the newsletter. I love the idea of the letter head, but the mind boggles about logos I might use for AUUGN. I do take the point that things may need to get a bit more organised but so far we have had many meetings, and they only seem to get better. Rest assured we are NOT about to collapse.

Piers Lauder is a very productive fellow, and is at the US SF meeting right now. He will be visiting Ianj at Bell, but as to job prospects you should ask him. None of the papers was terribly detailed since they were only summaries of what was detailed at the conference, and at least he wrote more than you get from the US meetings. Though the point about detail is well taken I only print what I am given and more detail can be obtained from Piers. I assure you that detail does not become insignificant when one is surrounded by a VAX, 11/70, 11/40, 11/35, 11/34, 11/03, millions of micros etc etc (getting greener??). Indeed when it comes to installing some of Piers's more recent, not quite bug free stuff, detail is paramount.

The cattle-dog looks like becoming a pet project of mine, and I do seem to have sufficient hands to get several pens to paper. As far as new software for the catalogue goes, if I have any real contact with the person(s) who write it, I will impress upon them the fact that it only gets included if they right it up. As to the form the catalogue will take, I have of recent years, much to my dislike, become more and more an administrator. Overly complicated schemes do not appeal to me, so the DB will be simple but I am sure we will be able to send you away smiling. I would be willing to include 'software under development' in the catalogue, but dont you think that the building of special hardware is complicating the project just as much as Ken McDonnel's 'stored once per item ...etc'.

I sent a letter some time ago to TM in Israel with more detail about work done in Queensland by Rick Stevenson on V7 strip downs. You will be able to read these letters in coming AUUGNs although the next issue may be a little to large for them to be included. In case you had not noticed, I dont produce AUUGNs bigger than about 80 pages, since this is just under the 250gram postage limit. Volume 3 number 2 should be a big issue, with writeups of the latest US meeting. Also in the next auugn you will see that backissues are available on micro-fiche, and if you cannot extract copies of some AUUGNs from Bruce Anderson I would be happy to send you a few little fish to fill the gaps. I have not yet received my tape of the V7 strip down but I am pleased to know

its on the way.

I plan to visit the US for the mid-year UNIX meeting and have been toying (just) with visiting the UK. We shall see how the finances hold out. Regards to your grand-dad.

Yours sincerely,

Peter Ivanov

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

INFORMATION SYSTEMS

Bunker Ramo Corporation
Trumbull Industrial Park • Trumbull, Connecticut 06609 • (203) 377-4141

January 14, 1981

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

Dear Mr. Ivanov,

NKER
AMO

I am taking the liberty of drafting this letter for my good friend and colleague, Peter Stevens, who wrote to you a while ago regarding the AUUGN. (Peter is presently busy at a training course, while I have time on my hands).

At the time of that first letter, both Peter and I were employed at TIE Communications, Inc. in Shelton, Connecticut. We have both since left to join the Information Systems Division of Bunker Ramo which is located only three miles away.

At TIE, we had just finished setting up a Version 7 UNIX system on a PDP-11/44, with RM02 disk and TS-11 tape drive. DEC supplied us with a modified V7 which they have developed for the/44; in addition to the 15 or so lines of code changes necessary for running on the /44, this version incorporates support for the TS011 and DZ-11 and an enhanced disk driver which allows a mixture of RP and RM disks. It can also be configured to run on non-I & D machines.

Assuming you have a V7 license, DEC will supply you with their version free or at nominal charge, in the form of a standalone boot tape like the one you get from Western Electric. The person we dealt with was:

Fred Cānter
DEC
1000 Continental Blvd.
Merrimack, N.H. 03504
U.S.A.
(603) 884-7852

We found him to be very helpful and cooperative. But he tells us that DEC won't release their version outside the U.S. without the intervention of their marketing organization, so you should "consult your DEC representative". I don't think they will give you any trouble.

Unfortunately, it looks like TIE may not be running UNIX much longer, as their last person with any UNIX expertise (and our last contact there) has just tendered his resignation.

Here at Bunker Ramo, Peter and I have been "given" an old /45 on which we are to bring up a V7 system. Right now we are still in the process of procurring disks and terminals, but we should soon be on the air. Our goals include connecting to an HP-64000 Micro Development System, an IBM 4300 and a VAX 11/780; eventually, we hope to support UNIX (or at least UNIX tools) under VAX/VMS which is the operating system our development people choose to run.

If you should happen to know of anyone who has done such work with VMS (and isn't selling it as a very expensive product), we'd love to hear about it. And although we no longer have access to a /44, if there is any other way we can be of help to you, please don't hesitate to ask.

Yours sincerely,

Kelvin Delbarre

Kelvin Delbarre

encls: order and payment for AUUGN

THE UNIVERSITY OF TEXAS AT AUSTIN
AUSTIN, TEXAS 78712

Computation Center
512/471-3241

February 10, 1981

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

Dear Dr. Ivanov:

Ian Johnstone of Bell Labs has given me your name as editor of the Australian UNIX Users Newsletter. Since I have taken on a similar task with respect to the U.S. UNIX Users Newsletter I would like to propose that we exchange Newsletters.

To start things off, I've enclosed with this letter a copy of each of the Newsletters I've mailed since becoming editor.

Sincerely,

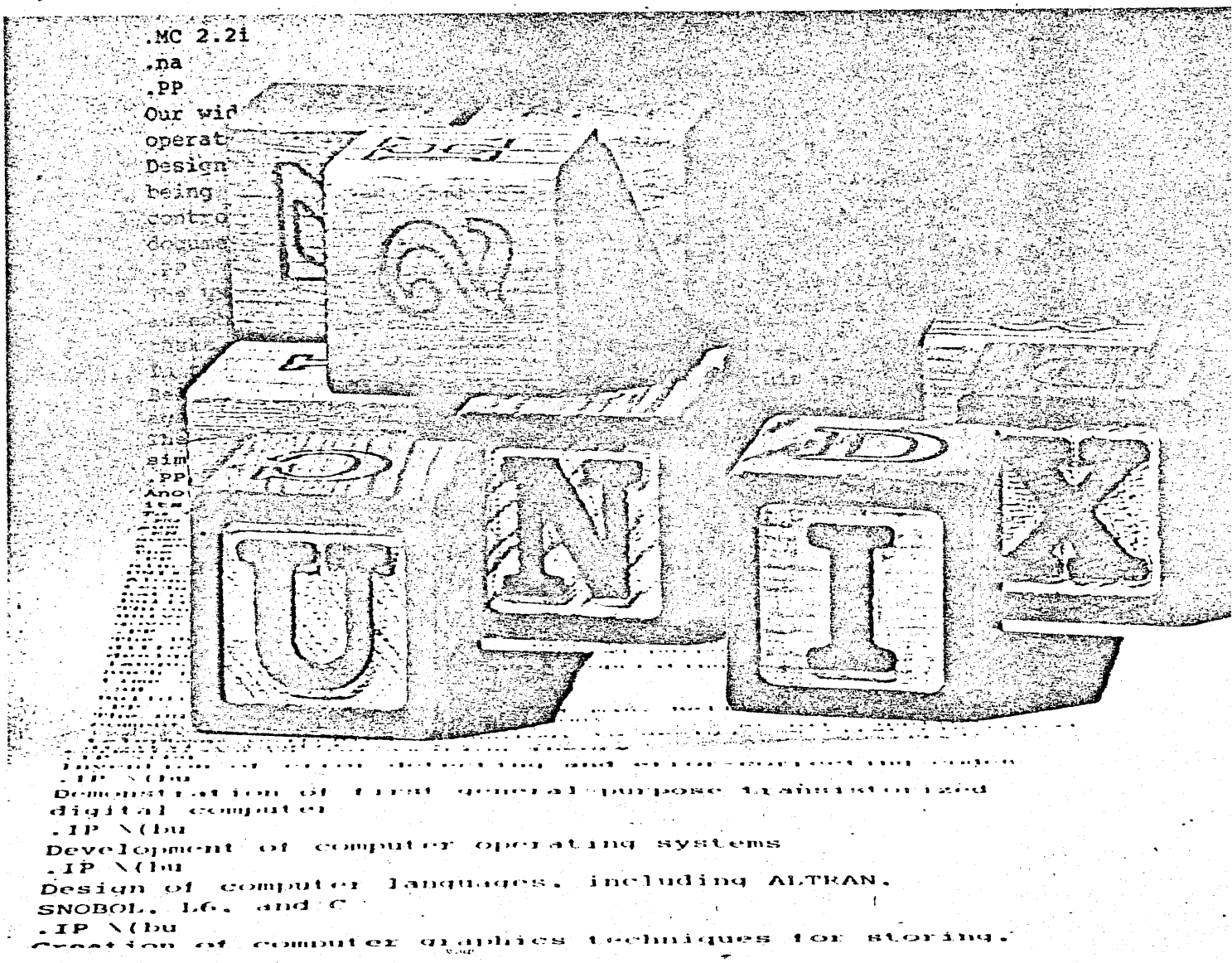
Waldo Wedel

Waldo M. Wedel
Newsletter editor

WMW:mp

Enclosures

An easier way to use computers... from a pioneer in computing technology.



The computer printout in the picture above contains th instructions and input needed for the *UNIX* system to format and produce the text on the opposite page.

Our widely accepted *UNIX*™ operating system is simplifying the use of computers. Designed to handle a variety of applications, it is being used to manage and maintain the telecommunications network, control experiments, develop software, process text, prepare documents, and teach computer science.

The *UNIX* system allows users to take small programs and assemble them like building blocks to perform complex tasks. In text processing, for example, the command "Spell Bell Labs Ad" tells a computer to proofread this ad against a dictionary filed in its memory. The program that performs the task was created by simply combining several smaller *UNIX* programs.

Another useful feature of our *UNIX* operating system is its ease in typesetting mathematical expressions. To typeset an equation like

$$\sum_{i=1}^{\infty} \frac{1}{x_i} = \pi$$

someone need only type: "Sum from i=1 to infinity 1 over x sub i=pi." The computer does everything else.

Since 1969 the Bell System has installed more than 1100 *UNIX* systems. Along with other Bell Labs innovations in computing technology, these systems are enhancing the way the nation's telecommunications network is planned, designed, built, and operated. Through licensing agreements with Western Electric, universities have installed over 800 *UNIX* systems, and government and industrial facilities are using over 400.

The *UNIX* operating system can be used with computers of different manufacturers because it is small, cleanly designed, and written in a general-purpose programming language. Such portability in a computer operating system saves time and money.

Building on the past

The *UNIX* system is just one of many Bell Labs advances in computing science and technology over the years. Among our innovations:

- Application of telephone switching concepts and technology to early computers
- First demonstration of remote computer terminal and data link
- Conception of electronic analog computer
- First design of AND and OR gates for diode circuitry
- Formulation of Information Theory
- Invention of error-detecting and error-correcting codes
- Demonstration of first general-purpose transistorized digital computer
- Development of computer operating systems
- Design of computer languages, including ALTRAN, SNOBOL, L6, and C
- Creation of computer graphics techniques for storing, manipulating, and presenting information
- Development of Fast Fourier Transform
- Design of central processors for switching systems having virtually no downtime

Looking ahead

Computing technology is having a major impact on the telecommunications business. It's increasing the Bell System's ability to provide new services and handle existing ones more efficiently. Today, for example, the nationwide telecommunications network links thousands of software-controlled electronic systems, making it by far the world's largest distributed processing network. We and our partners—Western Electric and the telephone companies of the Bell System—are putting technology to work so that the network will continue to evolve and expand its information-handling capabilities. The object, of course, is to meet the fast-growing and changing needs of Bell System customers.

For information about employment, write:

Bell Laboratories, Room 3C-303, 600 Mountain Avenue, Murray Hill, N.J. 07974

Please include a resume.

An equal opportunity employer.

Bell Laboratories

From Science: Service