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

Well here it is !!!!

WHAT ??????

At the last User Group Meeting in August it was decided that a newsletter should be produced to be distributed among Unix users in Australia so here it is the inaugural Australian UNIX User's Group Newsletter (AUUGN for short - nothing phallic intended). This Newsletter will attempt to promote better communication twixt the growing number of UNIX users in Australia, BUT a Newsletter is only as good as those who contribute \$0 if at any time information that may be of use or interest to other UNIX users comes your way please forward it for inclusion in the next issue.

It is my intention to produce six Newsletters a year. If you do not wish to receive the Newsletter just let me know, On the other hand if you do wish to receive it and are not on the attached list please mail your address.

US UNIX User Group Newsletter

ITEM	DESCRIPTION	COST
1	Subscription to UNIX News Jan 1978 - June 1978	\$ 5.00
2	Subscription to Unix News July 1978 - Dec 1978	\$25.00
3	Subscription to Unix News Jan 1979 - June 1979	\$25.00

This newsletter early in its life was very good value, in the last year it has been very patchy which I suspect resulted from the vast number of people wanting it. Now that a fee is involved I beleive that it should be worth receiving. If you wish to subscribe then forward a cheque (payable to 'The Rockefeller University') to:

Dr Melvin Ferentz
 The Rockefeller University
 Box 8
 1230 York Avenue
 NEW YORK NY 10021
 USA

UK UNIX Users Group Newsletter Change of Address

Dr D B Anderson
Department of Electrical Engineering
University of Essex
Wivenhoe Park
COLCHESTER CO4 3SQ
UNITED KINGDOM

SITE INFORMATION

A great deal of interest in UNIX has derived from its implementation on different types of PDP/11's and other machines (VAX and INTERDATA), so a catalog of Australian UNIX sites with details of CPU, memory and disk and other hardware details would be invaluable both as a guide for software exchange and knowing who to ask when problems arise that are related to particular configurations.

The activities of each site is also relevant. For example, a site might be primarily involved in teaching, data processing or in software development. Software might also be primarily aimed at an office environment, student environment or a programmer environment. Apart from what a site does, others would probably like to know what it wants to do in the future. Perhaps you might also like to include a UNIX 'wish-list' (perhaps this is not a good idea, as people often like to fulfill the oddest wishes, witness PL/1, ALGOL78).

Dependent on the speed of response, your responses will be catalogued and published in the next issue.

MINIUNIX

Those of you who attended the August meeting will remember Chris Rowles's talk on MINIUNIX, especially MINIUNIX's apparent unreliability !!. Well perhaps this section should have been titled 'MINIUNIX CURED - Programmer makes good'. Apparently all the problems he discussed arose from incorrectly configuring it. MINIUNIX on 'panicing' attempts to re-boot the system but if the bootstrap on the system is not the standard one another panic will result, unfortunately the trap vectors have been clobbered by the previous panic and an arbitrary piece of UNIX handles the second panic !!!! This as you can imagine leads to all sorts of problems.

Their system is now supporting three simultaneous users reasonably well and efforts are under way to increase various parameters so that four users plus remote batch to the cyber can be supported !!

Guess who's getting a VAX

It is rumoured that the Computer Science department at SYDNEY's oldest University has got \$250,000 to spend on a VAX 11/780. I guess the \$64 question is 'how long before they'll be running UNIX ??'.

Berkley Pascal available for 11/34's and 11/40's

Berkley Pascal has been available for use with PDP11/70's and PDP11/45's for over a year now, it is being used extensively for teaching and research at this university on the two 70's on this campus. In the last week have received from Bill Joy at Berkley their latest Pascal compiler/interpreter which now will work on 11/34's and 11/40's as well as the bigger machines. More details in December issue.

BELL SYSTEM TECHNICAL JOURNAL

The July-August (Vol 57, No 6, Part 2) issue of 'The Bell System Technical Journal' was devoted entirely to the UNIX time-sharing system. The issue contains nineteen articles (400 pages) whose titles are:

- 'The UNIX Time-Sharing System'
- 'UNIX Implementation'
- 'A Retrospective'
- 'The UNIX Shell'
- 'The C Programming Language'
- 'The MERT Operating System'
- 'Portability of C Programs and the Unix System'
- 'UNIX on a Microprocessor'
- 'Document Preparation'
- 'A Minicomputer Satellite Processor System'
- 'Statistical Text Processing'
- 'Language Development Tools'
- 'The UNIX Operating System as a Base for Applications'
- 'Microcomputer Control of Apparatus, Machinery and Experiments'
- 'Circuit Design Aids'
- 'The Programmers Workbench'
- 'A Support Environment for MAC-8 Systems'
- 'No 4 ESS diagnostic Environment'
- 'RBCS/RCMAS - converting to the MERT Operating System'

This issue is definitely worth reading, it is not highly technical in nature and serves in my opinion as a good indication of the current and potential uses of UNIX both at BELL and elsewhere. For those of you who would like a personal copy these can be obtained for \$1.65US (including postage) from:

Bell Laboratories
Circulation Group, Room 1A - 217
Whippany Road,
WHIPPANY NJ 07981
USA

Ian Johnstone
AGSM
PO Box 1
Kensington 2033
AUSTRALIA

(02) 662-3752

Our List of Licensed Subscribers

??? should you be on this list ???

R Giles, Basser, University of Sydney, 692 3264
W Harris, Basser, University of Sydney, 692 3216
I Jackson, Basser, University of Sydney, 692 3524
B Kummerfeld, Basser, University of Sydney, 692 3272
P Lauder, Basser, University of Sydney, 692 2824
G Cole, Computing Centre, University of Sydney, 692 3491
B Rowsell, Computing Centre, University of Sydney, 692 3491
A Watson, Computing Centre, University of Sydney, 692 3491
C Morgan, Basser, University of Sydney, 692 3216
K Rosolen, Electrical Engineering, University of Sydney, 692 3276
C Rowles, Chemical Engineering, University of Sydney, 692 3001
A Hume, AGSM, University of New South Wales, 662 3752
C Maltby, Computer Science, University of New South Wales, 662 3788
K Schofield, Faculty of Commerce, University of New South Wales, 662 3470
B Lederer, Mechanical Engineering, University of New South Wales, 662 2835
C Doney, Computing Services Unit, University of New South Wales, 662 3832
D Horsfall, Computing Services Unit, University of New South Wales, 662 3590
C Webb, Computing Services Unit, University, 662 3590
Ms W Stibbards, Computer Centre, Royal Military College, 66 3686
R Elz, Computer Science, University of Melbourne, 341 5225
K McDonnel, Computer Science, University of Melbourne, 341 6459
R Gaylor, Psychology, University of Queensland, (07) 377 3226
J Noad, Prentice Computer Centre, University of Queensland, (07) 370 6019
R Kelly, Computer Science, University of Queensland, (07) 377 2185
D McNeil, Economic and Financial Studies, Macquarie University, 88 9291
E Oliver, Economic and Financial Studies, Macquarie University, 88 9073
G Cox, Computer Centre, Australian Atomic Energy Commission, 531 3650
G Peady, Computer Centre, Australian Atomic Energy Commission, 531 3085
D Richardson, Computer Centre, Australian Atomic Energy Commission, 531 3085
H McKenzie, Computing Research, CSIRO, (062) 43 3259
D Ryan, Computing Research, CSIRO, (062) 43 3261
J Smith, Computing Research, CSIRO, (062) 43 3260
D Blatt, Computer Science, University of Newcastle, (02) 68
R Miller, Computer Science, University of Wollongong,
J Reinfelds, Computer Science, University of Wollongong, (042) 29 7311x981
G Bullen, Computer Science, NSW Institute of Technology, 218 9450
J Cady, Computer Science, NSW Institute of Technology, 218 9451
Dr C J Barter, Department of Computer Science, University of Adelaide,

Martin Tuori and Sandra Wright
Defence and Civil Institute of Environmental Medicine
Box 2000, Downsview, Ontario,
Canada M3M 3B9

ABSTRACT

The RSX-11M V03, RSTS/E V6B, and UNIX V6 timesharing systems are compared on the basis of criteria pertinent to software research and development. It can be argued that UNIX provides a better base for groups seeking a comfortable and powerful system.

INTRODUCTION

The point of view in this discussion is that of a user engaged in software development, problem solving, system programming or research. The services provided by these timesharing systems, and a person's interaction with those systems, will be discussed as follows:

1) Command Level:

- requesting program execution
- using command files, their syntax and flexibility
- accessing the file system, with provision for device independent I/O
- executing programs in combination

2) Utilities and Programming Facilities:

- languages available, and the quality of their implementation
- debugging and profiling tools
- editors, document preparation facilities
- software available, user group exchanges

3) System Calls / Executive Services:

- file I/O, raw device I/O
- communication with the user's terminal
- task initiation and synchronization
- real time tasks

4) System Programming and Development:

- the process of creating variants of the operating system
- modifying system parameters, accounts
- the 'privileged' state
- availability of source code for the O.S. and utilities

5) Documentation:

- organization and readability
- availability in machine readable form

6) Conclusion

COMMAND LEVEL

Of the time a user spends signed on to a timesharing system, most of it is probably spent in an interactive text editor. The second largest slice is likely spent in direct dialogue with the command interpreter. But the command interpreter is the cause of more frequent frustration, simply because it presents a wider variety of situations, and possible confusions. It is imperative, therefore, to have a clean, powerful command facility, which exhibits consistent behaviour.

The command facilities in timesharing systems allow the user to request the execution of a program or command file, and to specify arguments. A command language should allow for the selection of programs and command files from both local and global directories. By 'command file', or 'indirect command file', we mean a stored list of commands to be performed as a unit. The user can use this facility to assign a simple name to a complex request. Command files are the basis for 'batch processing', in which the system is instructed to perform some complex task, yet allow the user to continue working at something else.

In the UNIX 'SHELL', when a user types a program name, a search is made for that program in the user's working directory, in system directories, and in his personal utility directory. Command files are requested in exactly the same manner. The arguments to a program (or command file) are available within the program text as character string variables. The command syntax is uniform across executable binary programs, and command files.

In the 11M Monitor Control Routine

(MCR) the user requests 'installed' programs, his own programs, and command files using syntax unique to each form of action. When a task is requested by name, the list of installed tasks is searched. For a local program, "run progname" must be used; "\$progname" means a program (or command file) if it can be found in the system directory. "@filename" is used to initiate a command file. There is no good reason for the distinctions among these various action requests. Arguments to programs are retrieved by requesting a command line through a call to the executive. Command files cannot have arguments, although they may prompt the user for information. In such a case, the command file cannot be a batch process.

The command facility in RSTS is described simply as 'The Command Level'. Provision for command files exists in some system programs, but is not implemented in a general way at the user level. To run a program, the RSTS user types 'run progname'. Six special characters, and logical name assignments can be used to indicate directories other than the user's current directory. This is a useful shorthand; but the user is still expected to remember the contents of these various directories, and must ask for a program from one of them explicitly. Passing of arguments to programs is not present in a general form. Some system programs take arguments at the command level, like 'pip /li' on a single line. Other programs do not; for user programs, a technique using a 'core common string' will yield the command line.

In llm and UNIX, command file facilities exist which provide indirect command execution. These command files may be nested. In llm, this nesting is limited to a number of levels specified at compile time -- this provides protection against infinite recursion. An llm command file contains at most one line of input per program. To control a typical two-stage job (say compile and load) command files must be nested, using three files and thus scattering the instructions for this simple request. RSTS provides for batch job execution and pseudo keyboards, with which a reasonable command file facility could be arranged. The 'batch stream' command list is however not the same as the 'command level' command list. In UNIX, a command file can control any sequence of tasks -- it is directly equivalent to the corresponding series of commands typed at the user's terminal, and is thus a completely general and useful facility.

Although each of these systems

provide for grouping files in directories, only UNIX allows a full hierarchy (directories can contain directories). A file is specified by tracing a path to it through the file system tree, either from the root of the file system, or from the user's current directory. The node names can be arbitrarily chosen by the users (thus /user/bob/thesis/chapter1). Different mounted volumes are also indicated by node names (thus /sys and /user may be different disc packs). The UNIX SHELL provides a pattern matching facility which permits the user to access files using a shorthand notation -- a big help when the user is uncertain of the file name. Thus "*chapter?" means a name starting with anything (including 'null'), followed by "chapter", followed by a single character. The SHELL interprets this notation, replacing it in the command string with a list of names matching the pattern, before passing the command string along. A program or command file sees the command as if the user typed the list of names directly. The work is done centrally by the SHELL, so any new program benefits without additional programming effort.

In llm and RSTS, directories can be built only to one level. The syntax for file names is inherently numeric (dkl:[201,77]chap1), rather than allowing the user to use descriptive names. Under RSTS, system-wide logical renaming can circumvent this to some extent ('add logical dkl:[201,57]=bob'). A 'wild card' convention is adopted by some programs, which allows for substitution only in whole subfields (dkl:[201,*]chap11), and in RSTS for single letters as well (dkl:[201,57]chap??). This facility is built into programs locally (but not command files), and is therefore not available throughout llm and RSTS. If this seems like a small point, remember that a casual user might invoke the feature a dozen times in a normal working day.

The phrase 'Device-independent I/O' is used to indicate that a program designed to work with one peripheral may be used with a different peripheral without the need to rewrite the program. All three systems operate this way; the user uses a different prefix in the filename, for different types of discs, without any program change. Accessing a peripheral device in 'raw' mode (without imposed file structure) can be useful for simple inter-system transfers, backups, and examination of foreign volumes. In UNIX, raw device I/O is available by reading or writing a 'special' file system node, called "/dev/rkl" (for example). These nodes can be protected from indiscriminate access, as appropriate;

however, a casual user can copy a file straight to tape without difficulty ('cp myfile /dev/mt0'). Under l1m and RSTS, most user accesses to a device from the command level must be file-structured requests. Under l1m, the standard file copying utility, PIP will not permit a disc to be copied in raw image form, even by a privileged user. RSTS does allow image transfers. File structuring is an essential default, but is becomes a frustration when a user wishes to transfer information from one of the DEC operating systems to another. Each system has its own structure for file volumes, so it is usually done via DOS format magtape, native to neither l1m or RSTS.

It is commonplace to use two programs in combination, such that the output of the first is the input to the second. The classic example is a sort, followed by an update. In another instance, one might want to connect a text formatter to a program which constructs a list of the unique words contained in its input, and then print the resulting list on the line printer. To do this in l1m or RSTS, the user might have to rewrite the programs, using specialized executive calls, so that they communicate directly with each other in this manner. The usual solution is to use a temporary file on disc or tape. The commands would look something like this:

```
> runoff text.rno text.doc
> unique text.doc lp:
> pip
pip> text.doc/de
pip> ^c
```

In UNIX this problem is taken care of by a mechanism known as a 'pipe' between the programs. This is an in-core I/O buffer, which can be requested at the command level or at the program level. To the programs, a pipe behaves much like a file; existing programs can be used in a variety of new combinations, by command-level redirection of their I/O. Our example now looks like:

```
% nroff text.nr | unique | opr
```

The vertical bar, '|' indicates the pipe connection between two programs. Here the text formatter is connected to the unique word program, which is connected to the line printer spooling program 'opr'. There is no need to invent a name for the temporary intermediate, and it uses no peripheral storage.

l1m supports the following language processors: Macro-11, Fortran, Fortran-IV Plus, COBOL, Pascal, Basic, PL/I, and Coral-66. UNIX offers an assembler, C, Fortran, Pascal, Lisp, Macro-11, M11, Stage 2, APL, Snobol-3, Basic, and compiler-compilers called YACC and TMG. DEC's Fortran-IV Plus is available to run under UNIX. RSTS supports Basic+, Basic+2, Pascal, APL, Cobol, Fortran, Fortran-IV Plus, and Macro-11. These systems offer debugging facilities, with which a user can execute, examine, and change his running program. In UNIX, another facility is available for profiling an individual program to see where it is spending its time. This provides a breakdown of the percentage of time spent in each subroutine, and is useful for directing programming effort to improve efficiency. Such a facility does not exist in l1m or RSTS as delivered, but may be available from other users.

A powerful text editor is perhaps the most essential program development tool. TECO is a popular editor, and is available for l1m and UNIX, and RSTS. ED, the most commonly used editor under UNIX, is a descendant of QED. A version is available which provides for macros and multiple buffers. All three systems offer document formatters, although NROFF under UNIX provides a macro definition facility which greatly expands its capabilities, at the cost of being somewhat harder to learn. l1m does not come with a sorting program. UNIX does have a sort-merge program; and one is available for RSTS as an extra package.

The user community of an operating system can provide useful software and an exchange of ideas. The availability of software by this route can cut man-years off the development effort. The user groups for RSTS and UNIX are very active, and can provide a vast collection of software (some 100's of programs); the RSX community is not so active -- this observation comes from our survey of the software libraries and discussions with other users.

SYSTEM CALLS / EXECUTIVE SERVICES

An operating system should provide the programmer with convenient means for file I/O, device I/O, communication with the user, and task control. The three systems provide much the same overall services. The UNIX system calls are easy

to deal with, being a simple list of operations all at the same level. They look and behave like subroutine calls. All the documentation on system calls is contained in an alphabetically-ordered section of the UNIX programmer's manual, in a consistent format. The user can quickly find the description of a particular service. Under l1M, the executive services branch out into a variety of levels and functions, thus making them more difficult to learn. To find the description of a system service in the l1M manuals, the user must hunt around through tables of contents, indices, and the text itself. The RSTS system calls are grouped together in the Programming Manual, but are in no clear order. Some of the RSTS calls require that the user construct complex string expressions. For example, to set up a 'chain' to have a file spooled for printing:

```
AS=SYS (CHR$(8%)+("PROGNAME"+CHR$(13)
+CVT$(1000%)+ "Q OUTPUT.DAT"+CHR$(13))
```

Task synchronization is provided in an operating system by some means of signalling between cooperating processes. In UNIX this is done by the 'signal' system call, in l1M by 'event flags', and in RSTS by 'send/receive' system calls, which are implemented using message queues. None of these is entirely satisfactory--the usual complaints are the lack of a target (in the first two) when requesting these services, and the need for the receiver to be actively listening. In l1M, user programs which use synchronization must agree to certain system-wide protocols on the meaning of the flags; otherwise they may mangle each other.

Efficiency in I/O operations is of great importance; l1M permits an application program to perform asynchronous I/O -- i.e. to continue execution, even though I/O may not be complete. An event is noted, and possibly an asynchronous trap initiated at I/O completion. This can be useful in some situations. UNIX does not have this explicit provision -- asynchronous I/O does occur automatically on block-structured devices, transparent to the user program.

Real time tasking is of interest to those doing process control, data acquisition, and interactive control. UNIX is not intended as a real time system, although some installations use the system clock interrupt to monitor real time devices, such as a data tablet. On l1/45's and 70's, supervisor mode can be used to support a single real time task. l1M is intended as a real time system,

and claims a certain guaranteed response time; it is also intended for use in environments where a mixture of real time and program development tasks may be happening on a single machine. RSTS is not intended for real time use; it does have the useful feature that it can run RT-11 and RSX-11M as tasks, thus allowing some software for those systems to be used. But corresponding real time performance is not provided, of course. Several points should be kept in mind when considering real time performance. If a parallel processor is available, the host system may not be called upon to do real time processing. Secondly, there is a big difference between servicing a single real time task and servicing several simultaneously.

SYSTEM PROGRAMMING AND DEVELOPMENT

The SYSGEN procedure of l1M must be walked through each time a fundamental change is required in the operating system. This is a laborious exercise, for which only part may be relegated to an indirect command file. It takes a full day when you're green, and even accomplished system hacks require a couple of hours. Many of the system's parameters, however, can be changed on the fly, including accounts, memory partitioning, and terminal characteristics. In RSTS, this process is similar. It can be broken into a number of discrete parts, however, so that a complete SYSGEN is not always necessary. A RSTS sysgen may take as little as 1 hour. Again, accounts, terminal speeds, etc. can be changed on the fly.

In UNIX, a new version of the system is created by compiling part or all of the system source; this takes as little as 5 minutes. It is a rather painless task, as it requires attention only to the details to be changed. As in the other systems, the softer system characteristics can be modified during timesharing.

System requests which are potentially dangerous are protected by being 'privileged'. A user must provide the correct password to be allowed to make these privileged requests. Other requests which execute privileged system calls, but should nonetheless be available to the casual user, can be made so. All three systems have these capabilities. In RSTS and l1M, a user signed on without privilege may obtain privilege only by signing off, then signing on to a privileged system account. To return to

the unprivileged account, the user must again sign off, then sign back on. In UNIX, a user may become the 'super user' at the time of logging on, or from an unprivileged account (without having to sign off). This is as temporary a condition as the privileged user would like; upon exit from the super user state, the user is returned to his previous (unprivileged) state. This may seem like a small detail -- but it is exercised frequently.

The availability of source code makes an enormous difference to the effort of tailoring a system to the users' specific needs. A systems programmer can deal effectively with a large system only if he has the source code to read, and can understand it easily. If the source is not available, his hands are effectively tied. The llm system provides source for only some of the programs supplied, written in Macro-11 assembler. RSTS is supplied with source for programs written in BASIC (most of the utilities), and a few of the executive sources in MACRO-11. The UNIX system comes with the source for everything, written in the high level language C, with a few routines in assembler. The C language source code is easy to read, understand and modify.

DOCUMENTATION

The documents describing the llm system and services are poorly organized. They are heavy introductory reading, thrusting massive amounts of detail upon the user, without providing clear statements of function. The indices for the manuals are generally short, thus making random access of information very difficult.

The RSTS documentation is also lacking in these respects. Users complain that successive versions of the manuals are not always consistent, so that updating becomes a problem. Although the 'System Users Guide' is a useful introduction for beginners, there is no decent description of system fundamentals -- device handlers, etc. The indices are short, making it difficult to find information easily.

The documentation for UNIX is organized into 8 sections: commands, system calls, subroutines, peripheral devices, system file formats, user-maintained programs, user-maintained subroutines, maintenance programs. Each section is organized alphabetically. Each entry in

the manuals provides a one to three page description of a particular function, and its use. The key to the UNIX manual is its organization -- a functional breakdown into these sections, and a simple consistent format. It provides quick random access, and is very comfortable to use. The UNIX documents are distributed in machine readable form, and can be accessed at the user's terminal. This arrangement encourages continuing integrated documentation of changes and additions to the system. On-line documentation also allows for commands which search the table of contents of the manuals; this can help to quickly track down a particular piece of information. Other documents are provided which present utilities and language descriptions in a longer and more conventional format.

CONCLUSION

This report is not a completely detailed, technical examination of these operating systems. It is a look at those characteristics which contribute to user/programmer efficiency, comfort and satisfaction. Hopefully the criteria presented will aid the reader in gauging these and other interactive operating systems, and provide a vehicle for further discussion. Every interactive system user knows the feeling of reluctance to change over to a new system and working environment. But operating systems will continue to change; perhaps we should be striving for systems which can change gracefully. A user community should ask, "Is the cost of having to become familiar with a new system worth the increased productivity and comfort it offers?" The answer of groups which have moved to UNIX is consistently "Yes!"

* This is DCIEM Report Number 78X1

UNIX PANEL DISCUSSIONS

summarized by

M.M.Taylor
Defence and Civil Institute of Environmental Medicine
Box 2000, Downsview, Ontario M3M 3B9

ABSTRACT

Two panel discussions on UNIX were held. The first was mainly an introduction to the provenance, capabilities, and future of UNIX. The second was more technical. The sessions were not recorded, because of a recorder malfunction. This report discusses the question "What is UNIX and where is it going?" in a general manner. The topics covered by the individual participants are listed. Following the panel discussions, a Birds-of-a-Feather session was held to discuss the potential benefits and problems of forming a DECUS CANADA UNIX SIG.

Session I: What is UNIX, and who uses it? Participants:

M.M.Taylor -- DCIEM, Toronto (Chair)
D.R.D.Macfarlane -- Bell Northern
Software Research (Toronto)
M.I.Tuori -- DCIEM, Toronto
J.Kornatowski -- University of
Toronto
D.Haig -- Queens University, Kings-
ton

UNIX system summary description
-- D.R.D.Macfarlane
UNIX Users Group -- M.I.Tuori
New User Experience -- J. Kornatowski
Word Processing under UNIX -- D.
Haig

SESSION II: Where is UNIX going, and how should it develop?

Participants:

D.R.D.Macfarlane -- Bell Northern
Software Research, Toronto (co-
chairman)
M.I.Tuori -- DCIEM, Toronto (co-
chairman)
S.Wright -- DCIEM, Toronto
W.Pase -- I.P.Sharp, Ltd., Ottawa
R.Howard -- Waterloo University

RT-11 for UNIX users, and vice-
versa -- Wright
Varieties of Pascal -- Wright
Secure UNIX, EUCLID, and DOD-1
development -- Pase
Standardization in UNIX file sys-
tems -- Tuori
Waterloo work on UNIX -- Howard
Probable directions of version 7
UNIX -- Howard

UNIX was initially designed and implemented by a pair of DEC users on a PDP-9. Their names (Ken Thompson and Dennis M Ritchie) are now enshrined as the names of two key directories in most UNIX systems. UNIX was very soon thereafter implemented by them on a PDP-11, the machine series on which most subsequent implementations have been done. The initial version of the operating system proved so popular that special arrangements had to be made for its distribution to non-commercial users. Eventually, demand forced Western Electric to permit distribution at a high fee to commercial users, as well. The fee was intended to discourage commercial users from applying, but this discouragement was not successful. Western Electric maintains, through its licencing agreement, tight control over dissemination of its UNIX operating systems and the software supplied with them.

Western Electric now licences four distinct levels of UNIX for different purposes and for different members of the PDP-11 family: micro-UNIX for the LSI-11, mini-UNIX for machines without memory management, standard UNIX for machines with memory management, and Programmers' Work Bench (PWB), for developing operating systems and utilities for foreign machines. Each level of UNIX contains the capabilities of the lower levels.

Although most UNIX systems running to date have been obtained from Western Electric and run on PDP-11s, other UNIXes have been made. These, in general, conform to the user interface of Western Electric UNIX, but may be very different

internally. Interactive Systems Corp sells and supports an "enhanced" version of Western Electric UNIX, but will provide its own version for the VAX 11/780 shortly. UCLA has developed an independent "secure kernel" UNIX for the 3-mode PDP-11s, in which some system calls were disallowed because of possible security problems, but which otherwise looks to the user like standard UNIX. UNIX systems have also been developed for non-DEC machines.

A major convenience feature of UNIX as distributed by Western Electric is that the entire source is available. Most of it is written in a high-level language, "C", rather than in assembler code. It is thus relatively intelligible and additional facilities can be readily implemented.

The future development of UNIX seems assured, independently of Western Electric UNIX. The US Department of Defence (DOD) and the Canadian Department of National Defence (DND) are cooperating in the development of a certifiably secure UNIX operating system. Such a system would be unclassified, and should be running by late 1979. UNIX is expected to become a DOD standard operating system for most popular machines. As a complementary development, DOD is supporting a major new language development, DOD-1, intended as a standard system programming and concurrent real-time application language. DOD-1 will be available for UNIX systems.

UNIX was initially intended as a word processing system, and developed into a powerful time-sharing system only when it became apparent that the structure lent itself to such a development. Accordingly, the word processing capabilities of UNIX are one of its features. The facilities range from a simple runoff-like routine to a system that can handle multi-font and mathematical typesetting.

Most users who like UNIX enjoy its flexibility and human engineering. The human engineering is far from perfect, but it is usually thought to be much better than competing systems. Naive users can readily do complex things, and system programmers like its simplicity and power. The simplicity of its architecture was a primary reason for its choice as a basis for the DOD-DND secure operating system. The same simplicity makes it a natural candidate system for experimentation in operating system concepts, distributed processing, and so forth. In its Programmer's Work Bench incarnation, it is a good tool for

developing software for foreign computers and operating systems, and even in the straight UNIX version, much of the software for DEC operating systems can be made to run unchanged.

In any user-supported major system, there is bound to be a standardization problem. In UNIX, just about every installation tailors the operating system to its own taste. This creates the possibility that user created software will not be compatible across installations, and to some extent this incompatibility does occur. Usually, however, when a piece of software depends on an installation-dependent facility, that facility is distributed with the software and can be installed on the receiving system -- if it is not in conflict with some other idiosyncratic modification already installed. The UNIX user community is aware of these problems, and moves are afoot to standardize on one version of the main user program (the shell), which serves mainly as a command language interpreter. Inasmuch as each user can have his own shell, regardless of the installation standard, such a standardization across installations provides users with opportunities without forcing them into conformity. As a joke, one installation produced a shell which mimicked the IBM time-sharing user interface, complete with realistic delays and waits. Any user familiar with the IBM system could use that shell instead of the standard UNIX shell.

BIRDS-OF-A-FEATHER SESSION

Chair: M.J. Bennett -- University of Western Ontario

The final session was designed to investigate the desirability and feasibility of forming a UNIX SIG within DECUS CANADA. There are both problems and benefits from forming such a SIG. The main problem concerns conflict of interest, both with DEC and with the Western Electric Licencing agreement. It was agreed that Jack Richardson and Mike Bennett would sound out the relevant people in DEC to see if any insurmountable problems exist. The potential conflict between the organization of a DECUS UNIX SIG and the restrictions of the proprietary licence agreement were resolved by determining that any SIG would be open to any DECUS member, regardless of whether they had a UNIX licence, but that proprietary information would be discussed only at closed sessions, following the model of the

DECsystem 10-20 SIG.

It was generally felt by the meeting that there was considerable benefit both to DECUS and to the UNIX users of having a UNIX SIG within DECUS CANADA. UNIX users have generally broad interests within DECUS, and would wish to meet in connection with the DECUS CANADA Symposium. Several users reported difficulties in dealing with Canada Customs on the distribution tapes, and it seemed that a Canadian distribution centre would be a useful entity. Initially, and perhaps for a long time, such a centre would have to be at a user site, since no facilities for library distribution exist at Kanata, and since DECUS CANADA has no UNIX licence that would permit them to have the distribution software. This would present a significant problem to any DECUS UNIX SIG if it wished to have the software distribution integrated with the DECUS Library.

In view of the problems and potential conflicts, the meeting decided not to form a SIG, but appointed a committee to study the question and to serve as liaison for Canadian UNIX users who are members of DECUS. Martin Taylor was selected as chairman of this committee for the time being. Other members include Mike Bennett, David Macfarlane, Sandra Wright, Jim Dawe, and R Beyar. The meeting closed with the hope that the problems will prove to be resolvable, and that a DECUS CANADA UNIX SIG will be functioning before the Edmonton Symposium.

For the time being, enquiries from DECUS CANADA members and others interested in the possibility of a UNIX group within DECUS CANADA should be addressed to

Dr. M.M. Taylor
D.C.I.E.M.
Box 2000
Downsview, Ont.
M3M 3B9
(416) 633-4240

or

Mouse Watching by a Cat

by

Peter Ivanov

As supervisor of the largest UNIX system in Australia, I read with some amusement the section on UNIX security in the July UK Newsletter and decided to share with you some reminiscences about "UNIX cracking" from my colleagues and myself. The incidents described in this account are NOT fictitious, although some may seem so.

Firstly, however, I must say that Ian and Mike from UKC really only touched the surface of the problem and unfortunately showed admirable restraint in NOT resorting to "inelegant expedients" which in my experience can make a system about as stable as a teacup in a typhoon.

Our system in Computer Science at UNSW (see equipment summary) currently supports more than 550 student users, a small proportion of whom would very inelegantly stab the system in the back given half a chance. Whether through malice, incompetence or chance all users are dangerous to varying extents and a system cannot be called "secure" unless it at least resists (if not defeats) all attempts to bring it undone! Thus security, in my book, encompasses a number of aspects, some of which are

- a) Protection against depletion of system resources (such as disc space, proc slots etc),
- b) Protection of individual users information (files) from corruption or observation by other users, and
- c) Protection of privileged or proprietary software from those users not granted access.

Obviously when a system is cracked in such a way as to give the "cracker" super-user status that is the end of all security but if any aspect mentioned above is cracked, the results could be just as serious.

Now to some story telling.

We obtained our first UNIX system (level V) in 1975 and the first few tales date from this period. From the very first days "pseudo login" programs appeared, NOT in order to steal names and passwords for our little 11/40 system I hasten to add, but to crack the Cyber-Kronos system with which we shared the terminals. Soon it was quicker to see a second year student to get more money put in your Cyber account than to see the computing centre. It is obviously very difficult to defeat a well written "login" program and about all one can do is try and break its grip on the terminal.

Soon the "computniks" tired of "Cyber cracking" and turned their attention to UNIX. A super-user accidentally left the source mounted "readable by others" for about 30 minutes. In this time user file space soared (copies of source in various disguises) and a bug was discovered in "login" where password length was not checked properly and enabled a password of specific length to be entered followed by its known encryption. It took two days to clean up all the set-uid-root shells and spare source AND ALL IN 30 MINUTES!!!!

Another old favourite usable with shells which search in the order "x, /bin/x, /usr/bin/x" was to leave a dummy command (eg "ls") in any writeable directory (even your own) and wait for a super-user to blunder past. A simple "chmod, chown, unlink, exec" sequence worked wonders. As with fishing most of the fun was in selecting the correct bait and tackle.

A variation of the above method works well on sloppily maintained systems where "writeable-by-others" commands are some times available for even the shortest periods of time. By over-writing the command with one which does "that little extra" you once more wait for a super-user to execute it for you. The shell problem is easily fixed by changing the search order for uid zero, but its variant is more difficult and will be discussed later.

Inevitably, holes in existing code are always popping up. Our local classic was the "lpr-lpd" combination. Our "/dev" directory used to be fairly rigidly protected and in order for lpr and lpd to access user files and "/dev" they were set-uid-root. All was well until people discovered the remove (-r) after printing flag on lpr. I leave you to contemplate what could happen and assure you that it did.

Looking over a super-users shoulder can give exhaustive encryption programs a very good head start. Early on we discovered that passwords should be at least 10 characters long and, if possible, totally meaningless. Fortunately, Australia abounds in 10-30 character aboriginal place names that few would dare to pronounce.

In late 1977 our prayers for a larger PDP for teaching purposes were answered and I was given the rare opportunity of supervising building modifications, cabling, installation, maintenance, software development and making the afternoon tea for the workers, all of which, believe it or not, affect security.

The reason why building modifications and installation are important was summed up beautifully by a salesman of "secure systems" who said

"This system is guaranteed secure as long as it is
not removed from the concrete box....."

Several people I know could, given access to the front panel, crack any machine on campus in less than 30 seconds. To lay hands on our 11/70 one must pass through four lockable doors, the last of which has a unique key and is always locked.

Terminal laboratories should be located nearby and be laid out in a systematic manner so that during brief, irregular and frequent visits, particularly out of normal hours, budding computniks may be identified by sight and login name. They may then be watched and, when they have progressed sufficiently, asked around for a cup of tea and given something useful to do in return for "certain favours". This way they get to further their skills and we get cheap programmers.

Local software developments have resulted in a system as secure as humanly possible. AUSAM, described elsewhere, has implemented resource limits (procs, disc space, page limits etc) so well that I can recall only once running out of disc space, caused by a bug in a super-user program.

Other software changes are:

- a) "Bug" programs to watch computniks and warn of their presence.
- b) Programs to scan file systems setting modes and owners, and reporting on "funny" files (those with names containing unprintable characters or starting with '.', or having set-uid-root modes).

- c) Alteration of "init" to fork a "login" instead of a super-user shell in single user mode.
- d) Alteration of "login" to cope with the "no password file" situation.
- e) Changes to a vast array of programs (work still in hand) to create files mode 600 or 700 so that users are protected by default. This is a partial solution to the shell variant mentioned earlier but unfortunately one must still depend on super-users never being clumsy.
- f) Fixing "sgtty" to disallow calls setting modes on a tty not owned by the user. This practice was being used to acquire terminals and access to other users accounts by setting incorrect baud rates or parity and forcing the unsuspecting victim to leave because he thought the terminal had stopped working.
- g) Modifying "passwd" so it asks for the new password without echo so that users passwords are not visible on a "ps".

Finally some random points:

- a) We only have one super-user, root, and refrain from using this login name on any terminals except those over which we have absolute control, in or near the 11/70 room.
- b) Our shell searches in the order "/etc/x, /bin/x, /usr/bin/x, x" for uid zero and placing all super-user needed commands in "/etc" actually makes ones life easier. Also placing "su" in "/etc" completely removes any worries about "using the wrong one" when super-user status is required since "/etc/su" must be used.
- c) Periodically I run off a complete "ls -ali" of the mounted system and take it home for some sunday morning reading, along with lists of all set-uid files and copies of "my computniks" latest creations.
- d) To combat "login" programs a "grep login" of the whole system will usually obtain the desired results unless unusual measures have been taken to disguise the programs presence.
- e) When confronted with a user who has obviously been acting the fool (eg sending billions of nulls to some poor buggers terminal or stealing other users login names) he should be immediately "excommunicated". That is all his files should be made inaccessible and his initial shell should be changed to give a curt message to "see the system supervisor" before exiting. Nothing hurts a computnik like no computing. When he comes grovelling simply tell him what he did wrong (not what he is accused of doing wrong, note), promise that if it happens again the removal will be final, and give him back his fun. Naturally watch him very closely for the next few months.

At this point I was going to say something trite about our system never having been cracked but alas I cannot. During a normal "ps -agl" last week, the first year computniks were discovered running a setuid shell made up to look like a "getty". They were sprung in the act and under the threat of excommunication revealed that we too are sloppy. They had found a writeable command thanks to a poorly written run file, had compiled a special version of the command and had waited for a super-user to execute it. They were rather peeved that they only had about 30 minutes to explore before being caught but swore that they had done nothing nasty in that time.

I suppose the only thing I can say in defence is that these gentlemen have recently cracked several other machines on campus, but we asked them around for that cup of tea months ago.

en route to Vancouver
Friday 4 August 1978

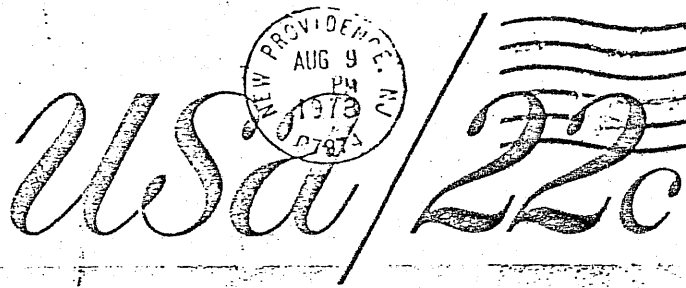
Dear Ian & Co.,

I got back into Unix mode on Wednesday with a visit to Interactive Systems in Santa Monica, ^{and Berkeley on Thursday} IS are prospering, with new offices being built all around them in an old warehouse building. Things are taking a little longer ^{business-wise} than they had hoped. They have a clause ^{in their Unix license} which allows them to sell Unix binaries at \$14,000 a time. Problem is that this is not attractive to companies ~~with~~ with 3 or more machines who get a better deal from the standard Unix license. Their biggest customer ^{ITT} is * licenses are part of the hassle. WE has temporarily stopped licensing new installations at UC Berkeley because preparing textbooks using phototypesetting ^{etc} in a possible commercial use. Berkeley computer centre has four 11/70's currently (but the fourth is idle pending WE decision). ~~They~~ They are using their CDC equipment for batch - with a version of SCOPE which departed from CDC in 1969! The EECS dept has ^{two} 11/70's ^{one} for Ingres, which seems to be working well using almost standard Unix V6 underneath. The Berkeley Computer Centre may buy software from Interactive Systems - interesting!

* Editor: everybody seems to be into editors using cursor addressing ^{mechanism} ~~mechanism~~ Benroughs. IS uses a Robin Elmer terminal with their own ROMS which cost them \$2500 each. It's very spectacular with elaborate windows to look at files - even a dual window feature. They are into phototypesetting in a big way. They also have a message system which is sophisticated - allows automatic filing, records replies, etc. - an integral part of the way their company operates. They have a VAX in house - are building a UNIX style interface to DEC software (e.g. like mid-way in Richard Miller's transformation from Interdata to UNIX). Berkeley is due to get 3 VAX systems shortly. Will write their own C compiler to avoid more hassles with WE. // Berkeley edited "ex" uses cursor addressing of hearse-size ADM-3A terminals (which they buy for \$750). This is pretty nice and saves a lot of message traffic (Bill Joy implemented it after being frustrated using a 300 baud terminal at home). Incidentally the EECS 11/70 has 72 terminals attached but dies when about ≥ 45 are active. Their disks are older and mixed, their comp centre has

P.S. Could you keep a copy of this for me please?

Dr J. Lewis
26 Morris Place
Madison NJ 07940



Ian Johnstone
Australian Graduate School
of Management

University of New South Wales
Kensington 2033
AUSTRALIA

AEROGRAMME
VIA AIRMAIL
PAR AVION

(2) Second fold

From Berkeley:

- * keep clear up to 1000 octal in case of interrupts from stray sources: ie don't pick up PS, PC from miscellaneous code
- * they have automatic re-boot out of "panic: trap"
— satisfactory about 85% of time.
- * check power-fail diagnostic to measure time left to do things after power fail trap
- * use "escape" to ~~control~~ stop and start terminal output

Additional message area

Some trouble with high density disks (300 megabyte) attached to Unibus — a problem associated with Unibus writes being recorded in cache memory — a hardware glitch which slows things down * WE is being sued for infringement of their consent agreement, by distributory software * New York Univ user group is not quite dead but overwhelmed by numbers — also there is now a real problem with all the different kinds of licenses: binary only, PWB etc * 500 people went to NY meeting * next national meeting will be in Santa Monica Calif next "spring" * a tape with 50 changes appeared "on the street" (near Murray Hill) last year. Includes major changes to text swapping apparently but as far as I can ascertain not for extra buffers. * Both Berkeley and IS would be interested to see your tape. Sorry about the writing but it is rather bumpy. Hope all is ok. This trip sure beats teaching. John

26 Morris Place
Morris NJ 07940
18 August 1978

Dear everybody,

Well I made it at last the promised land. However the plumbing is not all old-plated and contrary to rumour there is not a PDP11 in every room. On my first day here I met a whole lot of people whom I am now in the process of re-meeting and getting their names straight. So many things have occurred in quick succession, it is hard to get them all into perspective or even a reasonable order.

First the next release of UNIX is planned for October. This will be an internal release for the Bell systems (over 300 of them) and they are in a position to call at least some of the tune since they are paying the piper. Whether a general outside release will follow almost immediately is not at all clear. (Incidentally it is likely that future academic UNIX license agreements will not contain such a strict interpretation of academic use as the PWB/UNIX agreement, since the latter is viewed as a special product with a different origin and market.)

The new release will be call UNIX/TS, to distinguish it from UNIX/RT (which is a new name for MERT). So in future there will be at least two flavours of UNIX. (PWB/UNIX will continue to be based on UNIX/TS). Many features of the new release are described fairly fully in the July-August issue of the Bell System Technical Journal (Vol. 57, No. 2, Part 2). A copy of this is being sent to each UNIX licensee so that a few copies will be around eventually. (I have already sent one copy to MWA.) This BSTJ constitutes a major addition to the available documentation, so that many of you may want private copies. They have apparently printed 5000 extra copies to start with. If you wish to order a batch I will try and help from this end.

The most obvious new feature is the new shell - elaborated along the lines already expected. The paper by Steve Bourne (p. 1971) is the place to look. Editor changes are relatively few, and a few ideas from outside could be usefully picked up. Some of the QMC features such as "!!" have not made it yet. The idea of changing directories within the editor has aroused some interest. The "wa" command used to be available as "W" but it will be replaced by a better idea which will go something like "w !cat >> filename". This is only a better idea when one realises that the editor file can be piped to any process started by an arbitrary shell command. There is a similar construct for "e" also.

On the subject of typesetting, the Computer centre here has three phototypesetters and is resisting pressure to acquire a fourth, for then they would have to hire another operator as well. Typeset copy is used regularly around here - for the weekly calendar of events, for address lists, - even for lecture notes. However there are bastions of conservatism: it is not used in the telephone directory (but a line printer listing is) and except for the most recent issue, not in the BSTJ. "troff" as it now stands is very closely wedded to the Graphics Systems phototypesetter - in particular only four fonts. I was talking to Brian Kernighan yesterday and he said that they were looking very seriously at a Morgenthaler CRT typesetter, with a much wider choice of fonts, point sizes and up to ten times as fast - its cost is now down to \$40,000 (compared to \$15,000 for the GSI dev-

ice). Once this new device is ordered, a new version of "nroff/troff" will have to be written. However I suppose it is another case of "don't hold your breath".

The UPM seems to have grown significantly. I counted 154 entries in Section 1 (section numbers are no longer in upper case Roman numerals) and the permuted KWIC index runs to 25 pages. New commands include:

filters for output to various terminals eg. Diablo
accounting routines
awk: perform operations on lines matching patterns (a rival for "sed")
bs: a compiler/interpreter for modest sized programs (replaces "bas")
cu: call Unix (cf "ll")
deroff: remove nroff etc. constructs
three versions of "diff"
three versions of "grep"
fsck: file system consistency check and interactive repair
"cut" and "paste"
shutdown
spell with a "-b" flag for Piers (and me); based on a dictionary with 25,000 words!

** a failure
you "spill"
as too much
chemistry!*

A number of commands including "roff" seem to have disappeared for ever. New system calls include:

access: to files;
acct: turn accounting on or off;
chroot: change root directory on a per process basis (seems to be used for testing)
alarm & pause: replace "sleep" (new signal 14 SIGALRM)
syscall: indirect
umask: controls access permissions for newly created files on a per process basis
uname: returns name of current system version

There are a whole list of new routines in Section 3. All DHR's proposed I/O routines are now becoming standard and the old putchar, putc, printf, etc. are on the way out (vestiges remain). Access to the password file (still searched sequentially) is via "getpwent" etc. (Sixteen bit user ids are in, along with 16 bit group ids.) The remaining manual sections are not dramatically different. There is a new section 9 which documents the contents of a number of ".h" files (the number of which has grown dramatically).

Facilities here certainly are more lavish, eg. one office, two people, four phones (obviously one needs an extra phone for one's terminal). Up till now I have been using a tty 43 (this listing) which is pretty nice if you want hard copy but nothing fancy (apparently supply is having trouble keeping up with demand in the wide world). However being part of the telephone company tends to keep one if the fold as it were - everything is on a dial-up basis, limited to 1200 baud, and there is a noticeable preponderance of hard copy and lack of CRT terminals. Moreover there is no coffee room and people don't take coffee breaks. At lunch they tend to take the full hour and not to talk shop. On the other hand the whole building is air-conditioned which is highly appreciated right now as the weather is hot and very humid.

Widespread adoption of Unix throughout the Bell system may yet be the death of its reputation for solidity and reliability. There are now many, many people with their fingers in the pie and the new system is still in a state of flux. No one has yet called "enough" to changes and 'improvements'. There is a great long list of trouble reports from Bell installations of the kind which should be fixed and forgotten, eg.:

RMDIR can't remove a directory specified by a path name of more than 36 characters
SIZE cannot handle zero length files

lot of problems are now arising because the new shell pays particular attention to the exit status returned by commands - apparently some commands have been doing it wrong for years. Regarding the resident code, I haven't been able to find out too much yet. Handling of text segments has been vastly improved to eliminate unnecessary swapping; they have picked up our suggested change for zombie processes; files can of course be bigger; a method of getting more buffers without sacrificing segment five is being implemented; Ken Thompson says that up to 20% of cpu time may be going into "wakeup" in Level Six systems. The changes they have made to speed up the search for ready-to-run processes were definitely worthwhile. Some performance figures I have seen show "wakeup" still at the top of the list of most used procedures, but with only 8.9% of the time now.

The main new directions being pursued relate to portability. Bell doesn't want to be locked into one supplier for a variety of reasons. Quite a few references to the Interdata 8/32 appear in the UPM now. A VAX version of Unix is now running at Holmdel and performs 20% to 100% better, depending on the application. This is before any special advantage was taken of the VAX architecture, eg. for memory management. Code strings are about the same size, or a little less, than on the 11/70. so - if you really want to run large programs - the VAX now looks to be preferred. Further implementations on other equipment are certainly possible. In particular IBM has apparently looked very hard at Unix already. However there are more than a few legal problems inherent in that one, so again, don't hold your breath. More good news (?): the next release of UNIX will contain a Fortran 77 which bears comparison with Fortran IV+ for execution speeds. However the compiler is very large ... just makes it into "i" space.

Well that is about the lot for now. I've still got a lot of reading to do, and much material to digest. As a learning project, I am coding a modification to "login" and "passwd" to force people (where required by the administrator) to change their passwords at regular intervals - security is starting to become more important ever since somebody phoned up one of the systems, logged in as "ken" (which is an account on most systems), without a password) wrote to somebody on the system late at night "hi - it's me - what's the current root password" and got it! More to the point "ken" volunteered an interest to visit Australia .. possibly in conjunction with the next IFIP (i.e. world computer chess championship) so start saving your pennies, and putting together an official invitation

Cheers for now,

John Lions

P.S. (i) Have you seen Ritchie's & Kernighan's book on C yet?
It is very good

(ii) Bell has two patents on UNIX
(a) set-user-id bit
(b) on "typo"!



RUTGERS COLLEGE • DEPARTMENT OF MATHEMATICS • HILL CENTER FOR THE MATHEMATICAL SCIENCES
BUSCH CAMPUS • NEW BRUNSWICK • NEW JERSEY 08903 • 201/932-2390

September 7, 1978

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

Dear Ian:

Many thanks for sending the tape of goodies and especially for all the work that you and Peter Ivanov and perhaps others put in.

I have found myself a nice friendly small group of Nuclear Physicists and I am now a local user.

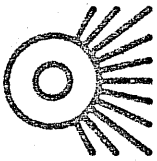
We saw John Lions and his family on Saturday at our beach house--yes, we are 50 yards from the beach. He reckons that much of what you have done at UNSW is well ahead of Bell. Apparently his notes on UNIX are a best seller and copies are locked in safes, etc. The local people at Rutgers were very pleased to have just received a copy of John's notes. I shall write again later when the locals have tried out some of your goodies.

Many thanks.

Yours sincerely,

David Hunt

1b



Bendigo College of Advanced Education
P.O.Box 199/Bendigo/Victoria 3550

Telephones
Edwards Rd. Campus
(054) 431877
Osborne St. Campus
(054) 439433
Mc Crae St. Campus
(054) 437944

DEPARTMENT OF INFORMATION SCIENCE

September 15, 1978

Mr. Ian Johnstone,
A.G.S.M.,
University of N.S.W.,
KENSINGTON, N.S.W. 2033

Dear Ian,

At the recent DECUS conference in Canberra, I was given your name as the distributor of UNIX in Australia. We run a PDP 11/45 RSTS/E system at present and would like to buy UNIX, at least for experimental purposes.

Could you send us the information we will need to order UNIX and would you list any options which must be purchased separately. We would also like to receive any literature that you may have on UNIX including such information as the devices it supports on PDP 11 systems.

Yours faithfully,

Ian R. Gillard,
Lecturer in Department.

IG:jr

18th September, 1978

Ian Johnstone,
Australian Graduate School of Management,
University of New South Wales,
Kensington 2033
Australia.

Dear Ian,

I spoke to Sandy Frazer today (friend of Ian Jackson's and Dennis Ritchie's boss (perhaps boss is not quite the word ... say department head)). Frazer is interested in Computer Systems including statistics communications (Greg Chesson is working on data comm. protocols and trying to set up a system which can be made to work on any system which has a C compiler).

Two other areas of interest: working with VAX and implementing Unix on 370, this presents the greatest challenge and any one who can pull it off will win a medal. Ritchie & Thompson don't like it because 370, error recovery will be the tail that wags the dog. VAX/UNIX is working, and Phase I is complete. Phase II is to take advantage of VAX architecture to do things better is now beginning and is to 'finish' by the end of this year. This work is being done at Holmdel (about 30 miles from here). By year end, the support group where I am, will have a VAX and possibly the centre of activity will shift closer to here.

.....

The Bell Laboratories User meeting last week was interesting in a number of ways - UNIX is such a big thing now that the cosy club atmosphere is disappearing - but people are very friendly and many of the troops are grateful for all the help they can get. I assure you that software distribution at UNSW is in great shape by comparison with here: some people have two or three 11/70's running 1976 model software - I may be able to boot some of your files to them even if the official channels won't accept them. Actually I have got "pack, unpack" running with STDIO ... may be able to get them into official use. Have to sort out "smdate" or it will screw "make".

.....

Cheers

John Lions.

UNIVERSITY OF CALIFORNIA, BERKELEY

BERKELEY • DAVIS • IRVINE • LOS ANGELES • RIVERSIDE • SAN DIEGO • SAN FRANCISCO



SANTA BARBARA • SANTA CRUZ

COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING
AND COMPUTER SCIENCES
COMPUTER SCIENCE DIVISION

BERKELEY, CALIFORNIA 94720

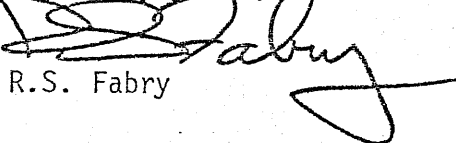
September 18, 1978

Mr. Ian Johnston
University of New South Wales
Graduate School of Management
P.O. Box 1
Kensington, New South Wales
Australia 2033

Dear Mr. Johnston:

Thank you very much for the tape containing UNIX version of the UT-200 emulator and batch subsystem for UNIX. Our CDC 6400 runs a homebrew operating system which supports both UT-200 and a local protocol for remote batch. We are currently using a UNIX link based on our local protocol because at the time the UNIX link was developed only the local protocol was supported. I hope we will be able to try your UT-200 protocol version and to compare the two schemes. It is quite possible that we will be switching to a standard CDC operating system in the next year or two. In such a case we will surely attempt to use your system.

Sincerely yours,


R.S. Fabry

RSF:jn

P.S. I have just learned that we will probably fire up your system to access the CDC 7600 at LBL!

26.9.78

Dear Brian & Ria,

.....

Sorry to hear the Cybers are playing up. We have a PDP10 here at Essex, although most of our stuff is on PDP11s, under Unix, of course! We had a Unix User's group meeting here yesterday - the European user's group. Mostly English & Dutch, and a couple of Frenchmen. Unix is not terribly popular outside England and Holland - in Europe, that is! Most universities seem to have it in England. BUT the main thing about the meeting is that UNSW is the undisputed leader as far as UNIX software goes! The number of references to NSW was staggering. Discussing NSW software etc. The final motion was to standardize on NSW software all throughout Europe! Really gave me a feeling of pride. Andy Tanenbaum from Vrjie was there - has put together a new super high speed Pascal compiler - several times faster than Berkeley Pascal. Almost as fast as C! Optimization etc. Double precision. We got a copy and people have generally agreed to use it instead of C for system program! Even Tanenbaum, who is actually an ex-patriot yank, made frequent and reverent mention of the NSW software! Pass this news on to Murray and John L, if you wouldn't mind. Mention was made of John Lion's and Greg Rose's papers in SIGART on operating systems - whatever issue it was! I'm sure if John Lions were to take a trip to Europe here, he'd be treated like royalty - well, amongst the Unix community anyway!

.....

Phil McCrea

UNIVERSITY OF ABERDEEN

HEAD OF DEPARTMENT
A. M. MURRAY, B.Sc., Ph.D.



TEL. No. 40241

DEPARTMENT OF COMPUTING SCIENCE
KING'S COLLEGE
OLD ABERDEEN
AB9 2UB
Scotland, U.K.

28 September 1978

Dr. Ian Johnstone,
AGSM, University of New South Wales,
P.O. Box 1,
Kensington,
N.S.W.,
Australia 2033.

Dear Ian,

Thank you for your letter of 24th August and all the material, which arrived safely. I passed this on to Peter Collinson who has now given it to Bruce Anderson, our new Newsletter Editor, for publication in our next issue. In particular I think that "Mouse Watching by a Cat" will go down very well with all readers!

We very much appreciate all the work you have been doing and, at our meeting at Essex on the 25th September, the UNSW software was much in demand. People were particularly interested in the new C compiler, which apparently falls slightly short of that described in Kernighan's new book, but which we might well make a standard.

As you see, I have given up as editor, after a two year stint, but it has been most interesting. Any further material should be sent to

Dr. D. B. Anderson
Department of Electrical Engineering
University of Essex
Wivenhoe Park
COLCHESTER CO4 3SQ.

Yours sincerely,

A handwritten signature in cursive script that reads "Peter Gray".

Peter Gray.

UNIVERSITY OF GLASGOW



Computing Science Department,
THE UNIVERSITY,
GLASGOW, G12 8QQ.

27th September, 1978.

TEL: 041-339 8855

EXT.

478/7458

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

Dear Ian,

I am writing to let you know that your magnetic tape arrived safely and that I have succeeded in extracting about half of the software so far, onto two RK05's. I am giving a copy of these immediately to Peter Collinson of University of Kent, who has spent most time in the U.K. in implementing your previous distribution. Peter is the new chairman of the U.K. Users group (as from this month), but we at Glasgow will continue to act as software distribution centre, at least for the time being.

Your new version of 'tp' is a great boon, and will make life a great deal easier for everyone involved in software distribution. The main problem, since very few installations in the U.K. have magnetic tape, has been to split up the software into sensible chunks which will fit on an RK05. I haven't had time yet to look at the documentation in detail, so am not entirely clear what is new or changed since the last distribution. At any rate, I have tried to arrange the software in order of likely interest to U.K. installations, so that people will not necessarily require to send us four RK05's.

Once I am happy that I have successfully extracted and catalogued all the software I shall return your tape under separate cover, with the collected U.K. software (first mailing) written on it. This includes, you will be glad to hear, a BCPL compiler from Kent. I should perhaps also mention (I don't think it was included in the list in the documentation you sent) that there is an implementation of the POP-2 language (developed by Popplestone & Burstall at Edinburgh University for research in artificial intelligence) available for Unix from Stephen Hardy of Sussex University. An early version of this has been used very successfully here in Glasgow for research in game-playing, but I believe there is a new release imminent. There is also an implementation of Wirth's MODULA language available from the University of York. If/

If you are interested, you should write direct to Ian Cottam, Dept. of Computing Science, University of York, Heslington, York YO1 5DD. The MODULA compiler is written in BCPL, and a BCPL compiler is provided with the system. A new release is also reported to be imminent of the Vrije University of Amsterdam PASCAL system, the first release of which was on the Third U.S. mailing. For more details you could get in touch direct with Andy Tannenbaum, Vergroep Informatica, Vrije Universiteit, de Boolelaan 1081, Amsterdam, The Netherlands.

We are very disturbed about the software distribution situation in the U.S., which appears to be in a state of chaos. So far as I know, neither your previous distribution, nor the first U.K. distribution, sent to New York In January has been sent to a single site in America. Of course there are problems of scale, which probably justifies the recent proposal to institute a charge. If this goes ahead we shall probably suggest that a single U.K. site (e.g. Glasgow) takes out a software subscription, and then redistributes to other U.K. sites. Since in any case most people here can only accept software on RK05's, this would make particularly good sense. In the meantime, you can rest assured that your software will at least be broadcast within the U.K. to all interested installations.

Thanks again for the tape. We look forward to receiving your future distributions, and your comments on PWB/Unix in due course.

Yours sincerely,

Alistair C. Kilgour

Alistair C. Kilgour.

TELEPHONE
345 1844
TELEGRAMS
UNI MELB PARKVILLE



University of Melbourne

COMPUTER CENTRE

Parkville, Victoria 3052

25 September 1978

Mr Ian Johnston
Australian : Graduate School of Management
University of New South Wales
PO Box 1
KENSINGTON NSW 2033

Dear Mr Johnston

I have been told that you run a most effective UNIX based computer installation, and I am writing to ask you to assist one of my staff in a study of the operating system.

If you can accommodate my request, I would like to send Mr David Millsom, a programmer at the Computer Centre here, to visit you and your installation for several days. I would like Mr Millsom to evaluate UNIX and to report to me on the following aspects of it :

- . Timesharing performance
- . System Security
- . Ease of Modification
- . Support, especially for new hardware products
- . Batch Processing and Spooling facilities
- . File Backup Facilities
- . Terminal support and control facilities
- . Word Processing facilities

I imagine that Mr Millsom would need to spend some time talking to people at the School, and some time using the system. I appreciate the magnitude of my request, and I could well imagine that you might not be able to provide these resources. On the other hand, Mr Millsom is an experienced DEC System-10 and RSTS/E systems programmer and may be able to contribute to your site.

Please let me know if a visit can be accommodated and if so, what dates might be suitable.

Yours sincerely

Geoffrey M Hudson
Acting Controller

TELEPHONE
345 1844

TELEGRAMS
UNIMELB PARKVILLE



University of Melbourne

DEPARTMENT OF COMPUTER SCIENCE

Parkville, Victoria 3052

4th October, 1978.

Mr. Ian Johnstone,
AGSM,
University of N.S.W.,
Kensington,
N.S.W., 2033.

Dear Ian,

I would appreciate it if you could send me a copy of the latest UNSW Unix distribution tape, for which purpose I enclose a 2400' tape.

I would be also grateful if you could pass on the smaller tape enclosed to the students who are working on the C version Tektronix routines (and whose names I unfortunately forgot to obtain) and ask them if they can send me a copy.

Thanks in advance,

Yours sincerely,

Robert Elz.

