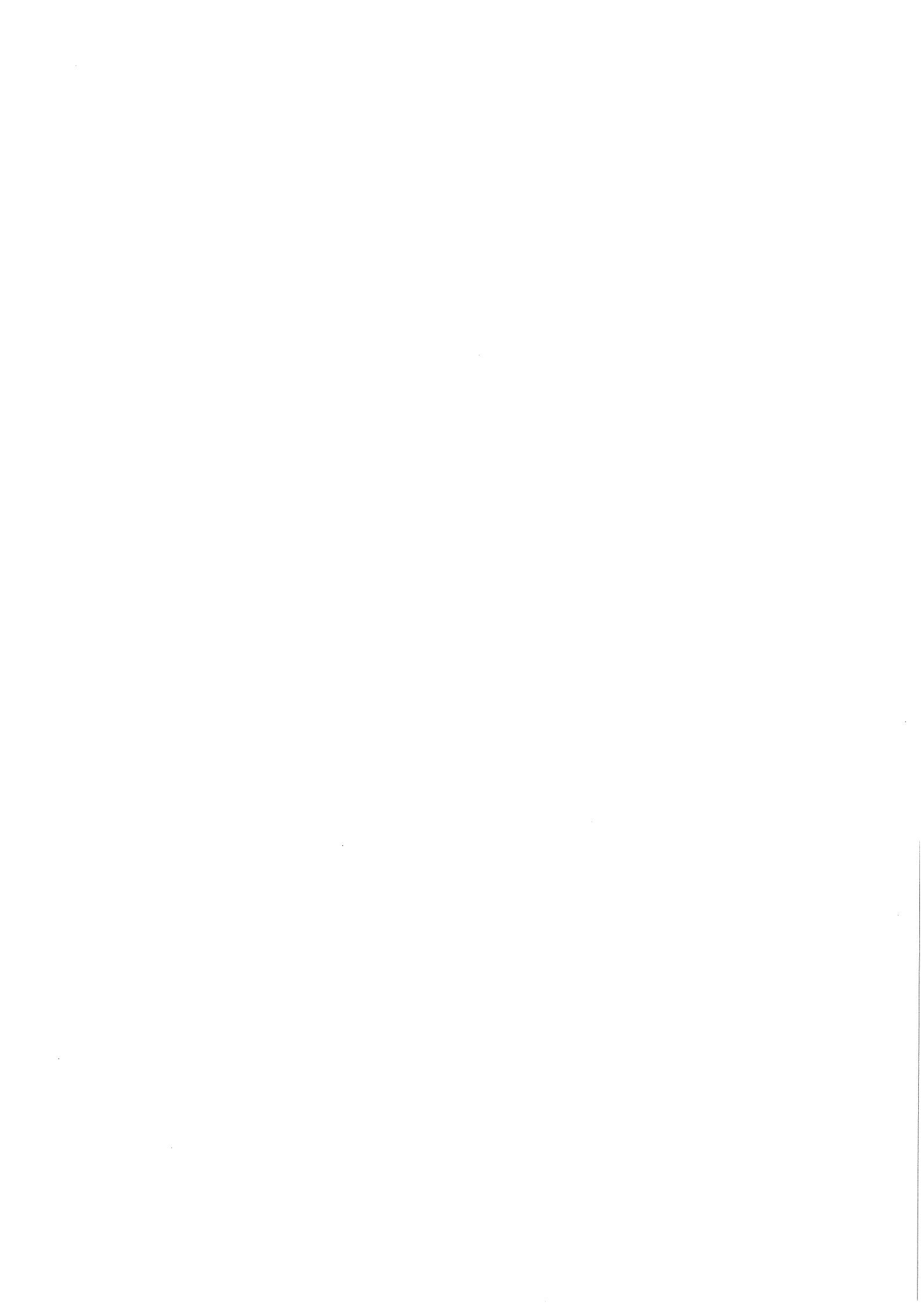


AUUGN

**Australian Unix systems
User Group Newsletter**

**Volume 7
Number 2-3**



The Australian UNIX* systems User Group Newsletter

Volume 7 Number 2-3

December 1986

CONTENTS

AUUG General Information	2
Editorial	3
Netnews	4
The Limits to UNIX	10
Using a UNIX system as a Multiprocessor Frontend	15
AUUG Constitution	22
Secretary's Report	27
Financial Summary 85-86	28
Budget 86-87	29
Management Committee Meeting Minutes	30
AUUG Summer 1987 Meeting Announcement	35
EUUG Spring 1987 Conference Announcement	36
\$echo magazine information	38
UNIX is a registered trademark	40
Letters to the Editor	44
AUUG Forms	45

Copyright © 1986. AUUGN is the journal of the Australian UNIX* systems User Group. Copying without fee is permitted provided that copies are not made or distributed for commercial advantage and credit to the source must be given. Abstracting with credit is permitted. No other reproduction is permitted without prior permission of the Australian UNIX systems User Group.

* UNIX is a trademark of AT&T Bell Laboratories

AUUG General Information

Memberships and Subscriptions

Membership, Change of Address, and Subscription forms can be found at the end of this issue.

All correspondence concerning membership of the AUUG should be addressed to:-

The AUUG Membership Secretary,
P.O. Box 366,
Kensington, N.S.W. 2033.
AUSTRALIA

General Correspondence

All other correspondence for the AUUG should be addressed to:-

The AUUG Secretary,
Department of Computer Science,
Melbourne University,
Parkville, Victoria 3052.
AUSTRALIA

ACSnet: auug@munnari.oz

AUUG Executive

Ken McDonell, *President*

kenj@moncsbruce.oz
Monash University, Victoria

Robert Elz, *Secretary*

kre@munnari.oz
University of Melbourne, Victoria

Chris Maltby, *Treasurer*

chris@gris
Softway Pty. Ltd., N.S.W.

Chris Campbell, *Committee Member*

chris@olisyd.oz
Olivetti Australia, N.S.W.

John Lions, *Committee Member*

johnl@elecvox.oz

Tim Roper, *Committee Member*

timr@uqcspe
University of Queensland, Queensland

Lionel Singer, *Committee Member*

lionel@pta.oz
Lionel Singer Group, N.S.W.

Next AUUG Meeting

The next meeting will be held in Adelaide.

Futher details are provided in this issue.

AUUG Newsletter

Editorial

Thank you to everyone who sent me comments on my first issue. The major criticism centred around the format of the Newsletter, especially the lack of contact addresses. I hope the information presented on page two rectifies this problem.

You may have noticed that this issue carries a strange number, 2-3, rather than the expected Volume 7, Number 2. Please see the letter to the editor from the AUUG Secretary for a detailed explanation of the new AUUGN numbering scheme.

Thank you to those who took time to contribute to this issue, especially to Robert Elz, John Lions, Mark Anderson and Peter Ivanov.

I have published the constitution in this issue and hope you have time to read it. I would like to hear more discussion on John O'Brien's proposal to incorporate the User Group.

Please remember I need your personal contribution to the Newsletter to help it make it relevant to you.

AUUGN Correspondence

All correspondence regarding the AUUGN should be addressed to the Editor:-

John Carey
Computer Centre
Monash University
Clayton, Victoria 3168
AUSTRALIA

ACSnet: auugn@monu1.oz

Phone: +61 3 541 3671

Contributions

The Newsletter is published approximately every two months. The deadline for contributions for the next issue is the 13th of February 1987.

Contributions should be sent to the Editor at the above address. I would prefer contributions sent to me via electronic mail using my AUUGN footer macros.

Advertising

Advertisements for the AUUG are welcome. They must be submitted on an A4 page. No partial page advertisements will be accepted. The current rate is \$200 dollars per page.

Mailing Lists

For the purchase of the AUUGN mailing list, please contact Chris Maltby.

Disclaimer

Opinions expressed by authors and reviewers are not necessarily those of the Australian UNIX systems User Group, its Newsletter or its editorial committee.

Netnews

Here are some amusing and interesting excerpts from the netnews collected by Peter Ivanov.

John Carey.

From: steven@mulga.OZ (Steven Bird)
Newsgroups: aus.jokes
Subject: errno(2) codes
Date: 14 Oct 86 06:29:35 GMT

At the USENIX Association conference in Atlanta recently a contest was held to invent the most humorous/bizarre/etc UN*X error message of the errno(2) 'EERROR' type. This contest had been tried at an earlier European Users Group meeting, where the winning entry was:

ENOTOBACCO Read on an empty pipe

You get the idea. A partial [alphabetized] list of entries from Atlanta follows; if your pun/wierdness tolerance is low, you may want to abandon ship:

EBEFOREI	Invalid syntax
ECHERNOBYL	Core dumped
ECRAY	Program exited before being run
EDINGDONG	The daemon is dead
EFLAT	System needs tuning
EIEIO	Here-a-bug, there-a-bug,
EIUD	Missing period
ELECTROLUX	Your code could stand to be cleaned up
EMILYPOST	Wrong fork
END.ARMS.CONTROL	Silo overflow
ENOHORSE	Mount failed
ENONSEQUITUR	C program not derived from main(){printf("Hello, world");}
EWATERGATE	Extended tape gap
EWOK	Aliens sighted
EWOK	Your code appears to have been stir-fried
EWOULDBNICE	The feature you want has not been implemented yet

And finally, a sort-of 'period piece':

EMR.ED	A host is a host, And coast to coast Nobody talks to a host that's close, Unless the host that isn't close Is busy, hung, or dead.
--------	--

I would also like this new signal to be supported:

SIGNUKE	Nuclear event occurred (cannot be caught or ignored :-)
---------	---

From: rick@seismo.css.gov (Rick Adams)
Newsgroups: aus.general
Subject: major usenet problems in the USA
Date: 1 Nov 86 15:27:30 GMT

Sometime on Wednesday, a site in San Francisco, CA in the USA accidentally sent out over 200 newgroup messages. It was a mistake and the person responsible deeply regrets it. However, what he did was destroy the ability of a fairly large portion of sites in the US to receive news.

What happened is that the 200 newgroups messages caused the ACTIVE file to become too large and the news system to reject ALL articles received after that until the active file was repaired manually. Much of the damage has been repaired, however there are still several areas that are dead.

The newest version of news (2.11) and recent beta test versions (anything from me after March 19, 1986) correctly reject these invalid newgroup messages, but sites running older software were badly crippled.

Since virtually all sites outside the United States get their news through seismo, and seismo is running 2.11, the non-us sites did not receive these damaging messages.

However, since the US is screwed up, much news was lost. This includes the mod.sources postings of news 2.11 among other things. news 2.11 will be posted again sometime next week. It is hoped that the US will be back in reasonable shape by the middle of next week. You will certainly see a drop in the amount of news.

The really ironic part is that the damage was so extensive because certain US sites did their best to propagate news as fast as possible. If the news had been batched and sent daily, the damage would have been minimal.

---rick

From: dpw@unisec.UUCP (Darryl Wagoner)
Newsgroups: net.unix-wizards
Subject: Slaying Gould dragon with a wooden horse
Date: 27 Oct 86 03:45:17 GMT
Keywords: secure unix trojan horse gould

Like many others, I attended the Unix Expo in New York City this week. At the Gould booth there was a large sign challenging Unix wizards to break into their "Secure Unix" system. The also gave out a flyer that stated the following:

```
-----  
|  
|                                GOULD  
|  
|          ***    HACKER CHALLENGE    ***  
|  
|                                UNIX EXPO 1986  
|  
|                                OCTOBER 20,21,22  
|  
|There is a text file on our 6040-I, UTX/32S - SECURE UNIX*  
|system. We challenge anyone to find out its contents. The file  
|pathname is:  
|  
|                                /usr/unixexpo/securefile  
|  
|RULES:  
|  
|1.      You must access the system from one of two user  
|         terminals. Login as "guest1" or "guest2".  
|  
|2.      All winners who successfully break into the system will  
|         be placed in a drawing for a grand prize winner  
|         of a 19" color tv.  
|  
|3.      In the event of any conflicts, the decision of the GOULD  
|         show director will be final.  
|  
|*Unix is a trademark of AT&T  
|  
-----
```

The contents of the file was:

"gould makes firebreathing,very high performance super mini machines."

I will present the case history of how I broke it using the most classic of all hacker tricks. In addition I located other weaknesses in their system that would allow even the most novice hacker to break into UTX/32S.

Having only limited time and a public account to do my hacking, I choose to use the Trojan horse attack. They willing revealed the environment that a user is put in is a restricted environment either much like or exactly like the chroot(2) system call of Unix. Which, to the best of my knowledge hasn't been defeated. Therefore, it would have been a waste of time

to try to defeat the chroot. The Gould salesman readily offered to show me their environment which revealed that PATH was set to " ./bin:/usr/bin:..." The key being the current directory is at the beginning of the search path. I quickly created a 'ls' trojan horse and put it in the guest home directory. Then I asked if root could get to the guest directory and asked him to do so. He did a cd to the guest directory and did a 'ls' which fired off my trojan horse. I could have waited for him to fall into the trap. I was afraid that some one else would find my trojan horse and use my work. Before I got everything right, I had to enlist the unknowing support of root twice more, due to differences in Secure Unix.

At this point, let me point out that in order for Gould to archive this level of security they had to strip out a lot of the things that makes Unix powerful (ex: suid bits) and isolated users into a chroot environment. UTX/32S also seems to have many cross checks with the different /etc/passwd and /etc/group files. The first attempt was to add my own "admin" account to the top level passwd file. This failed because the user id I chose wasn't in the group file. Another trojan later, I had my own group in the group file. Still the system complained about my group not being valid, but it did let me log in as an administrator. Then a very strange thing happened. I couldn't execute "cshsu(8)" (Gould's answer to su, but less secure). The real admin couldn't execute cshsu either. I returned the next day and asked if they had found out how I had broke in. With their audit file, I expected that they had. The answer was that I had broke something and they had to reboot; that caused the audit file to be removed. (note: if you ever want to cover your tracks on UTX/32S just crash the system.) Well, this gave me new hope that I could break it with another, better horse. With the next horse I copied the file in question to an area that I could read. (Besides making a copy of the file I could have also planted a worm or virus. Of course no one would do such a thing :-)

Then I showed them the content of the file in question. Well they lost their cool to say the least. I was happy to explain how I did it. They informed me that I had not really broke the system but just tricked the system admin and that the method that I used was immoral. I tried to argue with him about fifteen minutes without success. In hope of reasoning with him I enlisted the help of a impartial third party. (Who I haven't ask if I could quote so I will withhold this persons' name). This person listened to both sides and concluded that I had broken the system with a classic hacker technique.

The question I have for the net is: Is using a trojan horse a legit way to break into a system? What is your opinion?

SUMMARY of Gould UTX/32S System

I am not even sure that it can still be called Unix since SUID bits have been removed. After all that is what Dennis M. Ritchie patented as the Unix protection scheme. But as far as being secure, I will say that it is or could be as secure as any other unix system. It takes more forethought to break standard unix. It takes away one of the most powerful features of unix. The cshsu should have stripped out the current directory from the path like su(1) does. For that matter, the shells should have removed the current directory or at least put it at the end

just for good system hygiene. The tty driver should have a kill character to allow login to be killed to prevent trojan horses. There is also another hole I will not going into at this point.

--

Darryl Wagoner
UniSecure Systems, Inc.; Newport, RI; (401)-849-0857

{allegra|gatech|linus|raybed2|ihnp4|cci632} !rayssd!unisec!dpw

From: RISKS@CSL.SRI.COM (RISKS FORUM, Peter G. Neumann -- Coordinator)
Newsgroups: mod.risks
Subject: RISKS DIGEST 4.28
Date: 13 Dec 86 03:47:46 GMT

.
. .
. .
. .
. .

From: "Lindsay F. Marshall" <lindsay@ccheviot.newcastle.ac.uk@Cs.Ucl.AC.UK>
Date: Fri, 12 Dec 86 08:24:19 GMT
To: risks@csl.sri.com
Subject: An Amusing Article on the Taxonomy of "Bugs"

From "The Computer Bulletin" December 1986 by John Lansdown

One of the things Brian [Reffin Smith] touches on in his contribution [to the book "Science*Art"] is the creative potential of software and hardware bugs. He suggests that these might be distinguished from the more bothersome variety by being called, 'pugs'. I have often thought that bugs are as important to us in computing as snow is to Eskimos so, like them, we should distinguish the many different sorts with different names.

To give a few instances. There are some bugs which waylay unsuspecting computer users and beat them into the ground - often fatally: following Brian's example, these should be called 'thugs', particularly as they often arise through the programmer's or manufacturer's misunderstanding of theory. Some bugs are tiresome but the intrepid user can dismiss them as of no consequence: whilst not being thuggish in themselves, obscure others that sometimes are and hence make debugging particularly difficult: these obscuring bugs should be called, 'fugs'.

Some people claim to write totally bug-free programs - if their programs don't work it is not them that are to blame. The manual, the system or, more likely, the unintelligent user is at fault. Bugs in these programmers' code should be called 'smugs' or, perhaps, 'humbugs'. Bugs which put the system to sleep whilst it still appears to be working or, conversely, make it hyperactive - resulting in reams of unrequired printing or an endless sequence of error messages - should be called 'drugs'. Finally, those which give rise to that undesirable condition known as deadly embrace (brought about by such things as incorrectly designed database lockout mechanisms) should be called 'hugs'.

Only by properly naming these types of errors can we hope to study their true effects and ramifications. But what should such a study be called? I'd be happy to hear your (printable) suggestions.

[Such a challenge will not go unheeded.

'slugs' might be low-level bugs (like viruses?) that move slowly from one place to another, especially in systems having no shell.

'dougs' might be named in honor of Bell Labs' legendary Doug McIlroy (who with Bob Morris was responsible for EPL, the Multics development-language supersubset of PL/1). Doug used to make multiple patches to the live image of the compiler (which predated the official PL/1 compilers, by the way) ON-THE-FLY, oblivious to compilations in progress. I remember some horrendous (and of course completely nonreproducible) compilations resulting therefrom.

PGN]

The Limits to UNIX

J. Lions

University of New South Wales, Kensington, NSW 2033.

ABSTRACT

The continuing survival of the UNIX system for many years is hardly in doubt. However much of the novelty has already worn off. The initial motivation behind the growth of UNIX has been diluted, as UNIX has become institutionalised and moved into computing's 'main stream'. This is well exemplified by changes in the UNIX Programmer's Manual. While the UNIX system supports 'bottom-up' design methods, traditional data processing now favours 'top-down' design, which is not particularly well-supported by UNIX. Many newcomers to the UNIX scene have been puzzled and upset by this. What is really needed is a system that supports 'fully integrated, up-down-and-sideways, full-duplex' design methods. UNIX is evolving in that direction, but it isn't there yet.

Introduction

My involvement in computing goes back over 25 years, and with the UNIX system, more than 11 years. This gives me some reason to consider myself an 'old-timer', and as retiring president of the Australian UNIX systems User Group, to reminisce, and to give my view on 'how things were better in the good old days'. I am also concerned to understand the criticisms of UNIX by some newcomers to the UNIX scene, and to explain why there has not been the explosive growth in related software development that some people have been predicting.

The really good days for UNIX numbered about 4500 by my estimate, from 1 January 1970 to somewhere in 1983. By 1983 UNIX had become institutionalised, and the official title had become System V for ever, for reasons known only to marketing people. In my view, the success of UNIX has been in no little part because it did not kowtow to the accepted wisdom of the world's most commercially successful computer company. I accept that IBM often does something right, but this has been more often in the line of bamboozling their customers than in producing the best possible computer products.

Manuals

One problem with marketing people is that they often do not really understand what sells their product.

Right from the beginning, the UNIX Programmer's manual was on-line. This was indispensable for providing all users of the system with up-to-date information. This was absolutely imperative because the idea of making do with a moth-eared copy of last year's manual would have been useless. There are several stories of Ken Thompson's legendary ability to create whole new commands overnight (as with *grep*) or over a week-end (as with the *bas* compiler). Because disk-space was at a premium, and because the manual was displayed initially by printing on an ASR 33 or 37 (remember those) brevity and conciseness was at a premium.

Early editions of the UNIX Programmer's Manual exemplify some of the best written computer documentation I have ever encountered: the working of commands was described exactly, without ornamentation. You may have to read something three times before its full significance sinks in, but even so, this is far preferable to having the same information presented imprecisely in three different ways. Wry understatement was used to convey the limitations of programs in a vivid way. For example, the following all appeared in the Fifth Edition manual:

bas Has been known to give core images.
cubic Too much playing of this game will cause it to disappear.
mount Mounting file systems full of garbage will crash the system.
spline A limit of 1000 points is enforced silently.
wump It will never replace Space War.

You may deplore the software deficiencies thus revealed, but you cannot deny the eloquence of their portrayal! Today's manual makes much duller (and less informative) reading as all the personal touches are methodically expunged.

We, like the original system designers, found the on-line programmer's manual indispensable. The fact that it existed, the fact that we could change it and keep it up to date, the fact that we could print copies for our own use without difficulty, and the fact that we didn't have to depend on printed copies when they weren't available, were all fundamental in allowing us to operate independently of the university's central computing facilities (this was ten years ago). People complain about its terseness, but it remains that the original manual page format has proved durable and effective.

So what have the marketing people done? First they made some observations and a deduction something on the following lines:

IBM is the world's most successful computer company
IBM's manuals are mostly long-winded and indigestible

Everybody else's manuals should be the same.

As a result, well-meaning technical writers at A.T. & T. have taken the old UP manual apart and reorganised it so that it now satisfies nobody (at least that I know). This is in the name of progress, in the interests of becoming more 'user friendly'.

And what has IBM done for an encore? For their versions of UNIX, they have decided to remove the on-line version of the manual entirely (it takes up too much disk space)!

Commands

The manual has of course grown for many reasons: there are now many more commands, each with many more options. Each of these additions makes its contribution to the manual's size. The freedom with which new options and commands could be added has always been one of UNIX's strengths: many people have independently made their contribution and left their mark on the system. I have even scratched my initials in one or two places.

The initial UNIX command set was capricious because it was never planned to conform to any grand design. Many of the commands resulted from original research, e.g. into finite state automata and regular expressions, by their contributors.

Not so obvious has been the contribution of those who formerly laboured to clear the dead wood from the system, to cut the grass, rake the paths and to weed the flower beds. Although this used to be done frequently (many commands disappeared entirely between the Fifth and Sixth Editions of UNIX), it is now almost impossible to delete anything: everything has become institutionalised. Now the cry is 'upward compatibility', which is another way of saying 'stick to the mistakes of the past, whatever the cost'. This is not necessarily all bad, as IBM has shown by surviving comfortably for the last twenty years, but even IBM won't be able to keep it up indefinitely.

For better or worse, many parts of the UNIX system are now frozen and beyond renegotiation. We ought to appreciate that there was nearly ten years of probing and experimenting before this happened. Perhaps it was really enough, but I doubt it. The most important parts, e.g. the system call interface to the file system, may be mature, but other matters such as mechanisms for passing messages and sharing memory between processes (which were not developed by Ken Thompson, because he had not really decided the best approach to them) seem to be less than completely satisfactory.

The User Interface

The interface between the user and the UNIX system has been criticised more than once. This criticism is often uninformed and based on only fragmentary evidence. The standard interface pleases many users well; and displeases others, who classify it as other than 'user friendly'.

The user command language syntax evolved slowly and deliberately over a period of years. However else it may be described, it was not accidental! The important interactive commands are the shell and the text editors (most of the other commands are not interactive at all). The most frequently used commands have the shortest names. These are deliberately not English words and some are only vaguely mnemonic. They are invariably distinctive. Because shell command names are never less than two characters, they are completely distinct from the even more frequently used single letter commands reserved for even heavier interactive use, most notably by the editor. Although the command names may appear strange at first sight, they are quickly learnt and soon become familiar. One well-known name, *grep*, is even likely to become an accepted English verb!

An important aspect of the user interface of the UNIX system is the way commands can be combined ('mixed-and-matched') to form new commands with capabilities not envisaged by their original designers.

Until recently, the user interface has been attuned to the most popular style of user terminal, the 'glass teletype'. Improved terminals and faster data communications allow alternative approaches, e.g. based on menu-selection techniques. The UNIX system interface is capable of adapting to the new opportunities afforded by such terminals, as anyone able to afford a bit-mapped, mouse-driven terminal in the class of the Bell Laboratories' BLIT terminal already knows.

UNIX is a DBMS

It is sometimes forgotten that *UNIX is a data-base management system*. To understand this, one needs to remember one UNIX file can be equated with one data-base record in a conventional system. This may not seem ideal if your application is in traditional data processing. However it provides an excellent base for managing the bits and pieces needed to develop elaborate suites of programs. It has been proved often enough that UNIX is a favourite DBMS for program developers. And if you develop a few additional tools (as I have been doing recently), it can also provide a versatile DBMS for an office automation system. The effective marriage of UNIX for program development and traditional mainframe systems for DP program execution was demonstrated long, long ago with the development of the Programmer's Workbench.

The race between PICK and UNIX is not a race between equals. Those who predict that PICK will win have a narrow view of what computing is all about, namely traditional data processing. Traditional DP has dominated most organisations thinking about computing for so long that many people think that that is all there is. Most of the obvious DP applications have already been implemented, and so the real future lies with non-DP applications. I don't see developers for expert systems, or CAD/CAM packages, or text publishing systems being much involved in the UNIX versus PICK debate.

Packages

The package software markets for UNIX systems has disappointed many who consider the last few years development of packages for the IBM PC to have been successful.

I think this is easily understood: the UNIX system appeals to precisely those people for whom the IBM PC package solution is *not* a solution. Many package users are now concerned with how to integrate one package with another, e.g. getting the output of a word-processing package for use as the input for an electronic mail package. The 'obvious' solution to all this is more 'more highly integrated' packages. This is a losing strategy, unless an integrated solution is planned from the start. And one can't keep going back to the start over and over again — life is too short, if nothing else!

The UNIX philosophy of design implies a bottom-up approach. Much of the UNIX philosophy lies in the software tools approach that uses:

- a set of well-chosen, carefully engineered and tested program tools.
- means for invoking and managing assemblages of these tools to solve problems not envisaged by their original designers.

Thus UNIX and LEGO have a great deal in common. And like LEGO artefacts, UNIX artefacts are not always as smooth and as pretty as one might like. The software tools approach is powerful, and suitable for people exploring new applications and techniques. Nevertheless, new ideas are still possible and needed. In particular, any particular user interface can be 'papered over' and made to appear quite different if the application demands it.

Traditional DP likes tidy world with everything arranged in neat hierarchies. In such a world, top-down design is king. Because the problems of traditional DP are understood, such an approach is both feasible and desirable. Languages such as COBOL, Pascal, Modula-2 and Ada support this view. Large organisations like this approach. Software suppliers to large organisations like this approach.

Top-down designers like long names, bottom-up designers like short names. It is clear from names such as 'ed', 'cat' and 'grep' to which school of thought that Ken Thompson subscribes.

The Three Design Waves

Fred Brooks, in his book *The Mythical Man-Month*, suggests that a person or group implementing a particular system are most likely to get it right on their third attempt:

1. their first system won't be right because they don't yet understand the problem;
2. their second likewise, because although they now think they understand the problem and its solution, they underestimate the true complexity of the problem;
3. the third will be right because the novelty has worn off, and getting it right is the main objective.

We may turn this around and say that software designers come in three waves:

1. the first group of designers are solving new problems in new ways. They are not sure, when they start, what their final solutions will be. They favour a bottom-up approach, and they find UNIX much to their liking, and develop many interesting systems. However most of these systems are not 'commercial'.
2. the second group of designers are much less innovative in general. They work in structured environments with well-defined goals. They produce many systems, some of which are commercially successful. However these systems tend to be too rigid and difficult to integrate with other systems. Many of these systems currently reflect the views of traditional DP managers, whether these are appropriate or not. These are the systems, that according to IBM, are at 'the leading edge'. They may or may not have much to do with UNIX.
3. successful members of the third group are few, but there are many package implementers who are trying to join them! They are developing reliable, efficient, well-engineered, integratable software packages. We hope they will stick to UNIX, but we can't prove it. However we can be sure that their customers will not stick to MS-DOS.

Put another way, the first group like to use standard components to build no-standard assemblies, the second group like to build standard assemblies and find themselves using non-standard components, while the third group wants to build standard assemblies out of standard components.

Clearly package designers have been attracted to UNIX because, imperfect though it may be, it still provides the best base for group three designers. Many of the obvious gaps in the command set have been filled, but not always in the best possible way. There are irregularities in the range and handling of command options that affect different people in vastly different ways. The best that can be said here is that the problems are being addressed, and at least some of them will be solved.

In the office automation area, I see developments occurring on three main fronts:

1. shipping data around (electronic mail and all that);
2. presenting information to the user and obtaining a response (windows, Blit-ting, etc.); and
3. organising large amounts of data (the data base problem).

I see that UNIX systems are being used for much R & D on the first two fronts; certainly it cannot be said that the UNIX system community is falling behind. However I don't believe that as much as could be done has been done on the third front, that of organising data. Of course the conventional monolithic DBMS solution is available under UNIX, but as I have already said, UNIX suggests an alternative approach from the traditional one. Recently I have made a few inventions of my own to help me navigate around the vast array of files that I seem to have accumulated. I have found that I need more than just a hierarchical filing system and file identifiers that are just unique strings. While what I have found is obvious in retrospect, it is not exemplified in the standard UNIX system. Without the opportunity to say more here, let me say that I think I have found an area where new developments are still possible that will assist package developers in the future.

Conclusions

It is not possible to conclude a discussion of this kind in advance. Clearly the fast growing onrush of UNIX youth is almost spent. The question now is whether its momentum will carry it on to the final goal, to become the definitive software architecture, or whether it will fall short of the finishing line, to be overtaken by some other ugly duckling that is now languishing under some wayward tortoise shell. The race is on, but the betting is still open.

Using a Unix system as a Multiprocessor Frontend

Multiprocessor Group
Department of Computer Science
Monash University

ABSTRACT

The connection of a VAX 11/750 running UNIX and a capability based multiprocessor is briefly described. The multiprocessor uses the VAX as an intelligent I/O subsystem.

1. INTRODUCTION

We describe parts of the physical and logical connection between a VAX 11/750 (Kermit) running UNIX 4.2BSD and a capability based multiprocessor. The purpose of the connection is to provide the multiprocessor with an intelligent I/O subsystem. The two main tasks of the I/O subsystem are to act as a disk server and as a communications port for users and their Command Shells (CS) residing in the multiprocessor.

The multiprocessor currently consists of four National Semiconductor NS32000 series processors and 2.5 megabytes of shared memory. Each processor has approximately eighty percent of the power of Kermit. The multiprocessor also has an I/O adapter which is connected to an I/O bus. Various Dual Port Memories (DPM) can be connected to this bus to provide a communications channel from the multiprocessor to other systems.

The architecture of the multiprocessor implements password capabilities (Anderson, Pose, Wallace 86). A password capability is like a ticket which authorizes its holder to gain access to objects in a predefined manner. Types of objects which a capability does refer to are processes and data files. All objects are persistent entities. Password capabilities are values and thus can be stored in any communications media.

2. DUAL PORT MEMORY

Kermit and the multiprocessor are connected by a 128kb dual port memory. The bus connecting Kermit and the multiprocessor can support a sustained transfer rate of 3Mb per second.

To Kermit, the DPM appears to reside in two, physical address spaces. Read/write operations in one address space produce the expected results. However, a read operation in the other address space causes a read/clear operation on the DPM. A read/clear operation is one where the contents a memory location is read, and then cleared atomically. Read/clear operations are used to construct semaphores. Unfortunately, due to various hardware constraints, it was not possible to implement a read/clear capability for every location of the DPM. Consequently, only every fourth word of the DPM is capable of supporting a read/clear operation and of that word, only the most significant bit is changed. When read/clear operations are carried out by a process, it sees the effect as a change in sign of the addressed location.

The DPM is logically divided into two parts. The first part is used by the disk server for transferring pages to and from the multiprocessor. When the multiprocessor's disk controller

is constructed, this function of the DPM will be removed.

The second part of the DPM consists of a number of buffers which may be allocated to UNIX processes. Part of each DPM buffer can be used by UNIX processes to gain the attention of a process in the multiprocessor. The procedure is as follows: The process puts a password capability along with a message into a predetermined portion of a DPM buffer. Having dumped the capability and message into the buffer, an associated flag is set. This flag also resides in the buffer. Periodically, the buffers of the DPM are scanned by the multiprocessor and seeing a set flag attempts to send the message to the process pointed to by the associated capability.

3. I/O OPERATIONS

Once the DPM was installed the question of how to access it from kermit arose. There appeared to be two major approaches: manipulating a process's page tables so that the DPM could become part of its address space, or making the DPM visible as a file and provide the necessary routines to perform operations on it.

While it would have been possible to change kermit's kernel so that the DPM could become part of a process's address space, this option was not taken up for two reasons. First, it was felt that the changes required too much effort with the time available. Second, because of kermit's varied duties e.g. ACSnet operations, extensive manipulation of the kernel could not be allowed to interfere with the machine's uptime. The end result was that the DPM was made to appear as two special files (DPmem, DP0mem) in the device directory.

Operations on the DPM are performed by ioctl calls. The various operations are read a block, write a block, read/clear a word, and wait. The wait operation is one where a process invoking it blocks until the contents of the specified DPM location becomes nonzero. Reading from the DPM via DP0mem is a read/clear operation.

One quite unfortunate aspect concerning the implementation of ioctl calls is that it suppresses the natural advantage of a DPM compared to local area nets. Unlike a LAN, very small amounts of data can be made visible to the other machine quickly and with little cost. For example, making a word visible to both machines consists essentially of a write operation. Ioctl calls add an unwanted overhead to the transfer of small amounts of data; especially if the data is not contiguous. Considering that parts of the DPM are being used as communication ports for terminals, small, sporadic amounts of I/O is not uncommon. Hence the percentage overhead of an ioctl call can be significant.

In an attempt to reduce the number of ioctl calls to update a terminal's DPM buffer, it may be thought that large amounts of the buffer can be moved to an I/O process's address space in one go, the relevant parts updated, and then written back. Unfortunately, such an approach can have problems. Consider FIG. 1. It depicts a logical representation of a particular DPM buffer. The buffer contains sections for input and output, control variables for each of these sections, and two semaphores for these sections. Two processes, one for input and one for output, may be manipulating variables pertaining to their section at any one time. It is tempting for the input process to read a section and all the preceding variables in one go, update the pertinent variables in its address space, and then write back the lot. However, when the time comes to write back the updated variables for a section, "stale" values may be written back for variables of the other section. Hence a form of cache incoherency arises.

The data structure defining a DPM buffer can be rearranged so that the movement of blocks of items does not cause incoherency. To do so is not very elegant and considerable care must be exercised if the definition of the data structure is changed. On the other hand, the hardware of the DPM forced us to arrange the data structure due to the placement of semaphores in any case. Currently, we suffer the extra ioctl call. The extra call does not seem to make a noticeably adverse impact on performance unless Kermit must labour under a particularly heavy load; a not uncommon situation given the machine's varied duties. With the implementation of address mapping the problem would not have appeared.

4. I/O PROTOCOL

We now describe the intended protocol for establishing a communication channel between a process in the UNIX domain and a user's Command Shell in the multiprocessor. Currently, another version of the protocol is in use for test purposes. This version will be replaced by the protocol described below when the relevant information is gathered. The labeled arcs in FIG. 2 help depict the actions of the protocol.

There are four processes involved in making a connection. Two processes in the UNIX domain (Multid, Multidaemon) and two in the multiprocessor domain (CS, Admin). Admin is a process which holds a capability onto the entire DPM. Both Multid and Multidaemon are capable of seeing the entire DPM from the UNIX domain and are setgid processes.

To establish a connection with a CS the user first invokes the multi dialogue program (multid). Multid then strikes up a conversation with multidaemon via a socket. Multid requests a DPM buffer from multidaemon (arc A). Multidaemon looks for a free one and then sends a message to Admin in the multiprocessor asking it to check the tentative allocation and to derive some capabilities onto the DPM buffer (B). Admin does so and if all is well, gives a string (random) back to multidaemon (C). Multidaemon gives this string with the appropriate buffer number to multid (D). Multid then proceeds to send a message to the CS (E). The message contains the string and the number of the buffer being allocated. The CS, on receiving the message, quotes the string to Admin along with the buffer number (F). If the string is correct for that buffer Admin gives to the CS the capabilities to the DPM buffer (G). A two way communication link between the CS and multid is now established. When information is to be transferred to the CS, multid takes the information, places it in the allocated DPM buffer, and then sends a message to the CS. The CS then picks up the information from the buffer directly. Typically, the unit of transfer will be a string terminated with a newline character. When the CS wants to send information to multid, it dumps it in the DPM buffer and ensures that a particular location is made nonzero. Multid, waiting on that location, proceeds to get the information. Multid also has a facility to transfer the contents of files residing in Kermit into objects residing in the multiprocessor. We do not intend to go into all of the functions of Multid. Suffice to say that currently Multid is used to make connections to processes in the multiprocessor and act as a channel for terminals.

One important aspect of the multiprocessor that must be preserved is its autonomy and secure nature. Any security problems inherent in the UNIX domain must not be exploited to enable processes in the multiprocessor domain to compromise a user's interaction with their CS. One problem the protocol is designed to avoid is the presence of "black boxes" in the multiprocessor domain. By black box it is meant some process employed by someone to inspect, transparently, the data traffic between a user and their CS.

Assume the existence of a black box interposed between a CS and Multid. The black box is situated in the multiprocessor domain. Attempts by unscrupulous users to use such a black box will not work. Any communication between a user's CS and Admin cannot be intercepted and Admin will only release the capabilities to a newly allocated buffer once. Either the CS will get the capabilities to the DPM buffer or the intruder will. If the CS gets them the intruder will not be able to see the contents of the DPM buffer and thus its purpose is defeated. If the intruder decides to retrieve them then the CS will fail when it asks Admin for them. Thus the CS cannot talk to its user and will be unable to authenticate itself. Consequently, the user will be alerted to the black box. The black box cannot successfully emulate the user's CS as it does not know the codeword used by the CS to authenticate itself to the user. Any security breach will have to occur in the UNIX system itself.

5. FURTHER DEVELOPMENT

The size of each DPM buffer is to be increased and the UNIX domain will take on many editing tasks. Thus traffic between UNIX and the multiprocessor will consist mainly of commands and pages of data.

Kermit has been described in the context of a frontend to the multiprocessor. However, an equally valid view is that of the multiprocessor as a backend to Kermit. It is quite feasible to implement processes on the multiprocessor which, when called upon to do so, take on tasks set for them by Kermit. Kermit will of course have capabilities to those processes so it will not have any special privileges compared to any other user of the multiprocessor. Although the interconnection strategy and protocols are quite different, the AMOEBA system (Tanenbaum, Van Renesse 85) is capable of offloading work from UNIX systems to it.

Another promising area being considered is using the multiprocessor to support realtime applications developed in the UNIX domain.

6. CONCLUSION

Despite the tightly coupled connection between the VAX and the multiprocessor, there is no loss of autonomy for either machine. Unlike network connections such as ethernet, there is no hardware method for a machine to change the execution flow of another.

It was a mistake to make the DPM visible to UNIX processes as two separate files. Providing a mapping of the DPM into a process's address space, while requiring more effort to implement, is much more efficient and more suited to the random access nature of the DPM. Hence the kernel of Kermit will have to be adjusted. Single word manipulation of the DPM can then proceed in a very efficient manner.

To provide truly secure communication paths between a user and processes in the multiprocessor, the UNIX I/O subsystem is not adequate. There are a number of deficiencies in UNIX which make it unsuitable as part of a highly secure communications channel. An I/O subsystem based on multibus compatible components is being constructed. The I/O subsystem will handle only I/O and editing functions. Consequently, its kernel will be much smaller and more easily certified as secure than a full UNIX kernel. Also, the I/O subsystem is physically situated near the multiprocessor. Hence the security of the I/O subsystem will fall under the jurisdiction of the administrator of the multiprocessor.

One method that can be employed to avoid successful attempts at black boxing in Kermit is to have end to end encryption between a user and its CS. That is, when

communication is established between a user and their CS, all I/O between the two is encrypted according to some previously agreed upon key. To use such a technique however requires special hardware in a user's terminal, something which is unlikely to be available in most current installations. The user is pretty much forced in these types of situations to provide their own encryption equipment to interpose between the terminal and the untrusted I/O subsystem. Such a tactic is feasible. Chips are already available capable of implementing fairly strong algorithms such as the Data Encryption Standard (Gligor, Lindsay 79). It is not hard to envision a small "pocketsize" hardware device capable of being interposed by a user on commencement of a session between a terminal and the cable to the system. Such a device removes the threat of black boxes. Since the encryption process takes place only for I/O seen by a user, the burden on the process in the system performing the encryption/decryption procedure should not be overwhelming.

REFERENCES

ANDERSON M., POSE R.D., WALLACE C.S. 86: A Password Capability System, *The Comp. Journal*, VOL 29, 1. 1986.

GLIGOR V.D., LINDSAY B. 79: Object Migration and Authentication, *IEEE Trans. on Software Engineering*, VOL SE-5, 6. 1979.

TANENBAUM A.S., VAN RENESSE R. 85: Distributed Operating Systems, *Comput. Surveys*, VOL 17, 4. 1985.

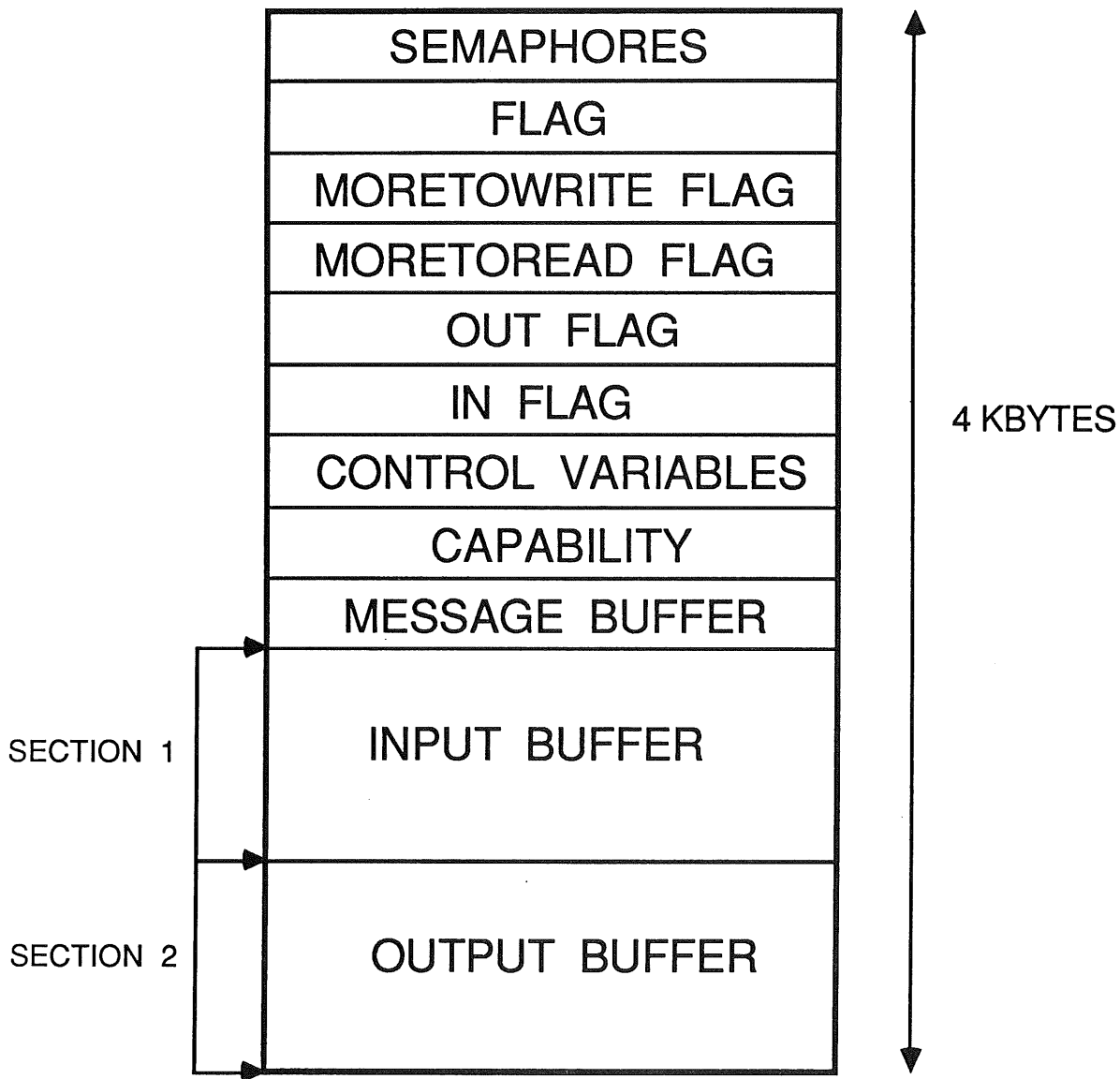


FIG. 1 — FORMAT OF TERMINAL BUFFER

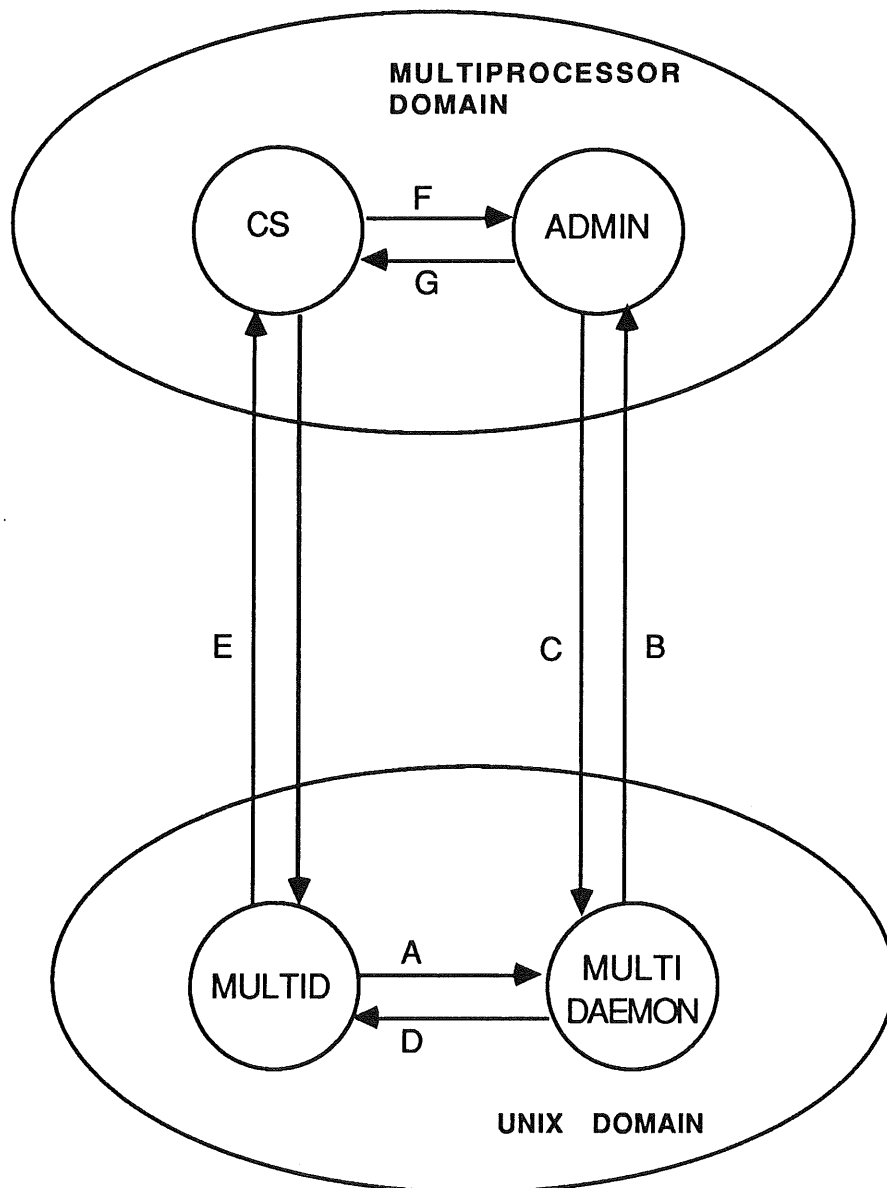


FIG. 2 — I/O PROTOCOL

Adopted 27/8/84
Rules and By-Laws for
the Australian Unix systems User Group

I. RULES

(1) The association shall be known as the **Australian Unix systems User Group**, abbreviated hereinafter to AUUG.
[UNIX is a trademark of A.T. & T. Bell Laboratories.]

(2) Office of the Association.

The office of the AUUG shall be at Room 343E, School of Electrical Engineering, University of New South Wales, Kensington, New South Wales, or at such other place as shall from time to time be determined by the Management Committee.

(3) Definitions.

In these rules, unless otherwise stated:

“he”, “him” and “his” shall also be construed to mean “she”, “her” and “her” respectively;

“Financial year” means the period from 1 June to 31 May;

“By-Laws” shall refer to the By-Laws of the AUUG;

“General Committee Member” shall mean a general member of the Management Committee;

“mail” shall imply the transmission of information in written or printed form, first-class pre-paid, via the general post or public or private courier service;

“unfinancial member” shall mean any member whose most recent term of membership has expired and who has not yet paid the subscription for the next twelve month period;

“voting member” shall mean any member entitled to cast a vote.

(4) Aims.

The aims for which the AUUG is established are to promote knowledge and understanding of the UNIX system, and of similar or related computer systems.

For the furtherance of these aims and to achieve its purposes, the AUUG may carry out any or all of the following activities: conduct technical meetings, conferences, discussion groups, panels, lectures and other types of meeting; prepare and distribute a newsletter and other publications; collect software and distribute said software to its members for their use; verify licenses of members for the purposes of administering the services of the AUUG; subscribe to or cooperate with or affiliate with or amalgamate with other associations formed elsewhere with similar aims; accumulate assets; and establish and promote other activities not included in the above list consistent with its aims for the benefit of its members.

(5) Eligibility for Membership.

Any individual or organisation who subscribes to the aims of the association, and who agrees to be bound by its rules and regulations and who has not been previously expelled from the association shall be eligible to join the AUUG.

(6) Application for Membership.

An application for membership shall be in writing on the form approved by the Management Committee and shall provide such information as shall from time to time be prescribed by the Management Committee.

(7) Commencement of Membership.

Membership shall become current on the first day of the month following the date on which a valid membership application accompanied by payment of the appropriate entrance fee plus annual membership subscription is received by the Secretary, and shall continue for twelve months from that date.

(8) Renewal of Membership.

Upon completion of the initial membership period and any subsequent periods, membership may be renewed for a further period of twelve months by payment of an additional annual membership subscription.

(9) Termination of Membership.

A member may resign his membership at any time by giving notice in writing to the Secretary. No member who resigns shall have any claim for a refund of subscriptions paid.

A member who has been unfinancial for more than two calendar months shall be deemed to have resigned his membership, and shall no longer be entitled to any privileges enjoyed by members.

Former members who have resigned will be entitled to rejoin the AUUG on the same basis as new members joining the AUUG.

(10) Amount of Guarantee.

Each member of the AUUG undertakes to contribute to the assets of the AUUG in the event of its being wound up while he is a member or within one year after he ceases to be a member for payment of the debts and liabilities of the AUUG contracted before he ceases to be a member, and for the costs, charges and expenses of winding up and for the adjustment of the rights of contributories among themselves, such amount as may be required but not exceeding fifty dollars.

(11) Expulsion of Members.

Upon receipt of a petition so requesting from twenty or more members, or half the membership, whichever is less, the Management Committee shall call upon any member to explain any alleged misconduct, and the Management Committee shall have power to suspend or expel any member who in its opinion has either been guilty of misconduct or has acted prejudicially to the interests of the AUUG or who has wilfully infringed any of the Rules or By-Laws of the AUUG.

(12) Annual General Meeting.

The Annual General Meeting shall be held within the second half of each calendar year. The date and general location of each Annual General Meeting shall be determined at the preceding Annual General Meeting but either the date or location or both may be changed by the Management Committee if it proves impossible or highly inconvenient to meet at the location previously selected or on the date previously selected.

(13) Ordinary General Meetings.

An ordinary general meeting of the AUUG shall be called by the Management Committee in conjunction with any technical meeting or conference or other function where attendance by a quarter or more of the voting members is expected by the Management Committee.

The business that may be conducted at such a meeting shall be as prescribed in the By-Laws. Notice of such meetings together with the agenda shall be sent to all voting members by mail at least four weeks before the date set for the meeting.

(14) Extraordinary General Meetings.

Upon receipt of a petition so requesting from twenty or more members, or half the membership, whichever is less, the Secretary shall call an Extraordinary General meeting of the AUUG for a date no later than two calendar months after receipt of the petition, and the business of the meeting shall be confined to matters described in the petition and to other matters specifically provided for in these rules and recorded in the written agenda sent to all members by mail at least four weeks before the date set for the meeting.

(15) Quorum for a General Meeting.

For each general meeting, the quorum shall be twelve members personally present and entitled to vote.

(16) Officers.

The Officers of the AUUG shall be: the **President**; the **Secretary**; the **Treasurer**; the **Returning Officer**; the **Assistant Returning Officer**; and the **Auditor**.

(17) Management Committee.

The management and control of the business and general affairs of the AUUG shall be vested in a Management Committee of seven members, namely: the **President**; the **Secretary**; the **Treasurer**; and four **General Committee Members**.

(18) Elections.

The election of Officers and General Committee Members shall be by postal ballot, under conditions defined by the By-Laws.

The term of office for all Officers and General Committee Members except the Auditor shall be for one year, from July 1 to June 30.

(19) Auditor.

The Auditor shall take office after the end of the Annual General Meeting following his election and shall hold office until the end of the Annual General meeting following.

If at any time the position of Auditor becomes vacant, the Management Committee shall at its earliest opportunity appoint as auditor a certified public accountant who is not a member of the AUUG, and he shall hold office until the next annual general meeting of the AUUG.

At least once in each financial year the Auditor shall examine the accounts and financial records of the AUUG. The Auditor shall certify as to the correctness of the accounts of the AUUG and shall report thereon in writing to the Secretary and to the members at the Annual General Meeting.

(20) Vacancies on the Management Committee.

The position of any General Committee Member shall be vacated if the member fails to attend any Management Committee meeting without furnishing a satisfactory explanation as to the cause of his absence, and if the Management Committee resolves that his office be vacated.

If at any time any of the principal Officers (President or Secretary or Treasurer) be unable to continue in office for any reason, then the Management Committee shall appoint one of their number to the vacant office.

Should a vacancy occur among the other Officers but excluding the Auditor, or among the General members of the Management Committee, then the Management Committee shall appoint an Ordinary Member of the AUUG to fill the vacancy.

The Management Committee shall make the approval of such appointments an order of business for the next General Meeting of the AUUG if any such meeting will be held before the next election of Officers and General Committee Members.

(21) Management Committee Meetings.

The Management Committee shall meet formally at least twice per year. Notification of time, place and agenda for each meeting shall be made in writing to each member of the Committee by the Secretary at least four weeks in advance. All members of the AUUG are entitled to be present at such meetings, and may speak when invited by the Chairman, but only members of the Management Committee may vote. The quorum for such meeting shall be four. Resolutions of the committee shall require a simple majority of the members present and voting. The chairman shall have a casting vote in the event of a tie.

(22) Distribution of Income.

The income and property of the AUUG however derived shall be applied solely towards the aims and purposes of the AUUG as set out in these Rules, and no portion thereof shall be paid or transferred directly or indirectly by way of dividend to any member of the AUUG at any time.

The AUUG shall not appoint a person who is a member of the Management Committee to any office in the gift of the association to the holder of which there is payable any remuneration by way of salary, fees or allowances.

Notwithstanding the above the AUUG may compensate the reasonable expenses actually incurred by any member in the conduct of the business of the AUUG under the direction of the Management Committee.

(23) Chapters.

Ten or more members of the AUUG may petition the Management Committee to form a **chapter** of the AUUG.

General rules for the organisation, operation, obligations and privileges of chapters shall be as resolved by the Management Committee or the membership as a whole from time to time.

Each chapter shall appoint a chapter committee consisting of at least a Chapter Chairman and a Secretary/Treasurer.

The chapter committee may convene meetings consistent with the aims of the AUUG, but may not enter into any financial commitments on behalf of or in the name of the AUUG except with the written approval of the Management Committee.

(24) Affiliation or Amalgamation with other organisations.

The Management Committee may at any time seek or discuss the possibility of affiliation or amalgamation with any other organisation whose aims are similar to or compatible with those of the AUUG. No agreement for affiliation or amalgamation may be finalised until the matter has received the assent of two-thirds of the members voting in a postal ballot.

(25) Dissolution of the AUUG.

Upon receipt of a petition requesting the dissolution of the AUUG from twenty or more members, or half the membership, whichever is less, the Secretary shall arrange for the question to be put to the membership by ballot no later than one month after the date that he receives the petition.

If two-thirds of the members voting agree, the AUUG shall be dissolved. If upon the dissolution of the AUUG there remains after satisfaction of all its debts and liabilities any property whatsoever, the same shall not be paid to or distributed among the members or Chapters if any, but shall be given or transferred to some public educational institution, or other institution to be determined at or before the time of dissolution by resolution of the membership.

(26) Changes to the Rules and By-Laws.

Changes to these Rules and By-Laws may be initiated at the request of a General meeting, or by the Management Committee. All proposed changes must be approved by a two-thirds majority of the votes received in a postal ballot of the members before having effect.

(27) Interpretation of the Constitution.

If any doubt arises as to the proper construction or meaning of any clauses in these Rules or By-Laws, the decision of the Management Committee thereon shall be final and conclusive provided such decision be reduced to writing and recorded in the minutes of a meeting of the Management Committee.

Adopted 27/8/84
Rules and By-Laws for
the Australian Unix systems User Group

II. BY-LAWS.

(28) Classes of Membership.

There shall be four classes of members: Ordinary members, Institutional members, Student members and Honorary Life Members.

(29) Ordinary Members.

Any person who is eligible to be a member may become an Ordinary Member.

(30) Institutional Members.

Any person or organisation who is eligible to be a member may become an Institutional Member.

(31) Student Members.

Any full-time student who is eligible to be a member may become a Student Member.

(32) Honorary Life Members.

Any person who is an Ordinary Member of at least five years standing and who has rendered special services to the AUUG may be elected via a ballot of the members as an Honorary Life member.

(33) Rights of Members.

Each member shall be entitled to attend all meetings of the AUUG, including meetings of the Management Committee, provided any prescribed attendance fee is paid.

Each member shall be sent a copy of the association's newsletter.

Each member entitled to vote in a ballot shall be sent notice in writing of all ballots and copies in writing of the annual reports of the Secretary, Treasurer and Auditor.

(34) Obligations of Members.

Each member shall abide by the Rules and By-Laws of the AUUG as they may from time to time appear. Each member shall respect licensing obligations.

Each member shall inform the Secretary of changes to his postal address.

(35) Voting Rights.

All Ordinary, Institutional and Honorary Life Members whose membership is current shall be entitled to cast one vote.

Any voting member may award his proxy to another voting member for the period of a single General meeting providing he so notifies the Secretary in writing at least 24 hours before the appointed time of commencement of the meeting.

(36) Membership Subscriptions and Fees.

The Management Committee shall determine before the commencement of each financial year a scale of fees for entrance to the AUUG, for annual subscriptions and for the attendance at meetings, for each class of members to be applied during that financial year.

(37) Chairman of Meetings.

At all General meetings of the AUUG and at all meetings of the Management Committee except where otherwise provided, the Chair shall be taken by the President, or in his absence, by a member elected by the meeting.

(38) Duties of the Secretary.

The Secretary shall keep or cause to be kept a register of members setting forth the names and addresses in full of all members of the AUUG.

The Secretary shall furnish to the Returning Officer a complete list of all voting members whenever this is required for the conduct of a ballot.

The Secretary shall keep or cause to be kept full and correct minutes of all resolutions and proceedings at General meetings and Management Committee meetings of the AUUG.

The Secretary shall conduct correspondence on behalf of the AUUG.

The Secretary shall, during his last month of office, prepare a written report on the state of the affairs of the AUUG for distribution to the membership.

(39) Duties of the Treasurer.

The Treasurer shall keep or cause to be kept correct accounts and books and records showing the financial affairs of the AUUG.

The Treasurer shall notify the President and Secretary in writing of the usual location of said accounts, books and records whenever this location is changed.

The Treasurer shall receive all fees and subscriptions and all other monies on account of the AUUG and provide receipts for the same. The Treasurer shall deposit all monies received into a bank account maintained by the AUUG.

The Treasurer shall receive accounts for payment for services rendered to the AUUG, and as directed by the Management Committee arrange for payment from the AUUG's account.

The Treasurer shall, during his last month of office, prepare or cause to be prepared a written report on the financial affairs of the AUUG for approval by the Auditor and subsequent distribution to the membership.

(40) Execution of Contracts.

The Management Committee, except as otherwise provided in these Rules and By-Laws, may prospectively or retroactively authorise any Officer or member of the AUUG to enter into any contract or execute and satisfy any instrument, and any such authority may be general or confined to specific instances, except that any contract whose dollar value exceeds an amount predetermined by the Management Committee must be specifically authorised in advance by the Management Committee.

(41) Disbursements.

Signing Officers for the AUUG's accounts shall be the President, the Secretary, the Treasurer and one other General Committee Member chosen by the Management Committee.

All cheques, drafts, and other orders for payment of money out of the funds of the AUUG, if for less than a limit established by the Management Committee, may be signed by only one Signing Officer.

For other amounts, each such instrument must be signed by at least two Signing Officers.

(42) Conduct of General Meetings.

Written notice of the time and place for each meeting and its agenda shall be mailed to each voting member of the AUUG at least four weeks before the date of the meeting.

Business conducted at such meetings shall be confined to matters included in the written agenda, reports from Officers, and resolutions instructing the Management Committee to conduct a formal ballot of the membership on matters of substance.

Such resolutions shall not be binding on the Management Committee unless the meeting was attended by at least twenty voting members, or half the membership, whichever is less, and the resolution was supported by at least two-thirds of the members voting.

(43) Voting.

All voting by the members with respect to the election of Officers and General Committee Members, with respect to the election of Honorary Life Members, with respect to changes to these Rules and By-Laws, and all other substantive matters shall be conducted by postal ballot.

Every voting member of record as of the date of entry of a ballot into the mails shall be entitled to vote in the ballot.

On all questions to be put to a ballot, the Secretary shall designate a date for the ballot to be placed in the mails, and the due date shall be four weeks after that date. The Returning Officer shall nominate the address to which voters shall return completed ballot papers by mail. A ballot will not be counted if it is received after the due date or if the ballot paper does not comply with the instructions printed on it.

The ballots will be received by the Returning Officer, and counted by him and the Assistant Returning Officer. The Returning Officer shall report the result of the ballot in writing to the Secretary no later than two weeks after the due date.

(44) Conduct of Elections.

Elections shall be held annually for all positions of Officer and General Committee Member.

Nominations for each position shall be received by the Secretary up until the first day of May each year. Each nomination must be in writing, must name the position or positions sought, must be signed by at least three voting members, and must be countersigned by the nominated member who must be a financial voting member of the AUUG.

The Management Committee shall ensure that at least one valid nomination is obtained for each position such that if no further nominations are received all offices and positions may be filled. Where only one valid nomination is received for a particular position by the close of nominations, the nominee shall be declared elected forthwith, and no ballot for that position shall be held.

Within first seven days of May, the Secretary shall advise the Returning Officer of all valid nominations received, and if a ballot is required shall advise him of a date no later than the fifteenth day of May for the ballot for all contested positions, and shall provide him with a list of voting members.

While any Ordinary Member may be nominated to more than one office or position, no person shall be elected to more than one position. Ballots shall be determined in the following order: for President, for Secretary, for Treasurer, for General Committee Member, for Returning Officers, and for Auditor.

(45) Election of Honorary Life Members.

If before the first day of May the Secretary receives a petition from at least twenty voting members requesting the election of a member of the AUUG to the position of Honorary Life Member, then he shall arrange a ballot of the membership on this question to be conducted in conjunction with the annual election of Officers and General Committee Members.

November 1986, AUUG Secretary's Report

• Membership (as of 12 Nov 86)

166	<i>Ordinary Members</i>
37	<i>Newsletter Subscribers</i>
42	<i>Unfinancial Members, (still entitled to newsletters)</i>
10	<i>Newsletters issued gratis</i>
255	<i>Newsletters</i>
10	<i>Overseas surface mail</i>
14	<i>Overseas air mail</i>
124	<i>Unfinancial members</i>
441	<i>Mailing List</i>
813	<i>Names</i>

• Correspondence

1. A paper on standardisation from the NZUSUGI was received. A considered reply is probably called for. A brief acknowledgment, and request for permission to publish the paper in AUUG has been sent.
2. Another letter from NZUSUGI seeking our assistance in formulating lists of technical software, literature, etc, for Unix systems was received. This has not been answered.
3. A letter from Ryserson Sch new licensing manager for outside Japan was received. It requests a list of officers, with contact details. No reply sent yet.
4. A second (form) letter from Schwark, giving details of the distribution of the "\$ echo" magazine. He requests distribution to the members, or a list of membership for a direct mailing. No reply.
5. Several requests for membership forms and details were answered.
6. A letter to Peter Ivanov thanking him for his years of service as editor of AUUGN has been sent.

Robert Elz

AUUG Financial Summary

1985 - 1986

Net Worth	1 July 1985	12112
-----------	-------------	-------

Income

Membership & AUUGN subscriptions	8956
UnixWorld	11750
Mailing list sales	250
Bank interest	580
Meeting surplus	1194
	22730

Expenditure

AUUGN	4144
Secretarial	1571
UnixWorld	362
Meeting loss	686
	6763

28079

Notes

1. This summary covers the year 1 July 1985 to 30 June 1986. The financial year of AUUG should be 1 June to 31 May. This will be corrected by shortening the current financial year to 11 months.
2. The income of approximately \$16000 was largely contributed from two sources. The greatest contributor, UnixWorld is not expected to appear again. Then next greatest contributor was the excess of membership over AUUGN costs of AUUGN issues published during the year.

AUUG Budget

Financial year 1986 - 1987

Assumptions

1. AUUGN costs of approximately \$42 per volume per subscription
2. Approximately stable membership numbers
3. One management committee meeting other than in conjunction with an AUUG meeting
4. Some legal fees re incorporation
5. Meetings neither incur losses nor bring income

Net Worth	1 July 1986	28079
-----------	-------------	-------

Income

Memberships & AUUGN Subscriptions	12000
UnixWorld	3900
Mailing list sales	500
Bank interest	3000
	<u>19400</u>

Expenditure

AUUGN	11000
Secretarial	1500
Committee meeting	1200
Legal fees	1500
	<u>15200</u>

Operating profit	4200
------------------	------

32279

Notes

1. While membership is expected to remain stable, revenue from membership is expected to rise from financial members, and to the introduction of Institutional Membership.
2. The UnixWorld income is from the UnixWorld held in 1986. This contributes almost the entire expected operating profit.

**Minutes of the AUUG Management Committee Meeting
November 17, 1986**

1. The meeting opened at 14:10. Present were Chris Campbell (CC), Robert Elz (KRE), John Lions (JL), Chris Maltby (CM), and Ken McDonell (KENJ) in the chair. Apologies were received from Tim Roper (TR) and Lionel Singer (LS).
2. Moved (JL, seconded CC) **That the required notice of the meeting be waived.** Carried (5-0).
3. The minutes of the previous meeting (September 1986) were read.
4. Corrections to the minutes:
 - a. CM seconded the motion by JL at item 10.
 - b. Item 18 is incorrect, the motion at item 14, as modified in items 15, 16, and 17 was carried 6/0.
5. Moved (CC, seconded JL) **That the minutes as amended be accepted.** Carried (5-0)
6. Business arising out of the minutes
 - Item 6* The document on how to hold a meeting was still awaiting action by Peter Wishart.
 - Item 6* Arrangements for handling subscription payment by credit card have not been completed, however CM will continue pursue the matter.
 - Item 9* The balance sheet has been redone, though it is still to be audited.
 - Item 9* No additional funds have yet been moved to a high interest account.

There was some discussion on various high interest bearing account types. It was agreed that a "no risk" type account was required, a bank or building society account would be appropriate.
 - Item 12* Complete.
 - Item 14* Complete.
 - Item 31* KRE indicated that he had obtained some secretarial assistance, but that so far no expense had been incurred.

Discussions of several other matters were deferred to later on the agenda.
7. Revised minutes of the February 1986 committee meeting were presented.
8. It was noted that no action has yet been taken on the decision to provide a prize of a fully paid trip to an AUUG meeting for the best paper submitted by a student.

9. Moved (JL, seconded CM) **That the minutes be accepted.** Carried (5/0).
10. KENJ gave the president's report.
- The president will not be able to attend the next meeting due to overseas commitments.
 - The transition of AUUGN editors seems to have gone well. The most recent edition (Vol 7, number 1) was slightly delayed pending the delivery of papers from the Canberra AUUG meeting. Production delays after the issue was ready had not helped.
 - *There was much discussion on possible methods to get AUUGN back on schedule — to catch up the current 10 month backlog. All methods suggested had as their substance simply dropping the missed AUUGN issues, the differences amounted to how the produced issues should be labelled.*
 - Moved (JL, seconded CM) **That each future AUUGN issue carry 2 consecutive numbers until the issue number agrees with the normal publication schedule.** Carried (4/1, CC dissenting)

11. The secretary's report was tabled.

The paper from NZUSUGI was discussed. It was agreed that the committee should seek expressions of interest from people following the IEEE P1003 effort, to see if the NZUSUGI position is supported in Australia or not.

Moved (JL, seconded CC) **That the paper should be given to the editor for publication in AUUGN.** Carried (5/0).

The secretary was directed to reply to the second letter from NZUSUGI, and suggest that they seek help from /usr/group or from USENET.

A list of office bearers, with postal and electronic mail addresses should be sent to Ryserson Schwark in response to his letter.

The advertisement from AT&T for \$ echo should be given to the editor for publication in AUUGN, without charge to AT&T.

12. KENJ reported that he had arranged a token for the previous newsletter editor, in accordance with the resolution at the previous meeting. However this has not yet been completed, KENJ to follow up.
13. The treasurer presented his report.
- Misleading entries in the balance sheet have been corrected. The result was not affected.
- Forms to alter the signing officers were available.
- Forms to obtain Mastercard and Bankcard were also available. The group's bank does not handle Visa. It was agreed that this was not important, Mastercard and Bankcard would be sufficient.

A summary of the accounts was tabled.

Funds still need to be moved to a high interest account.

Moved (KRE, seconded CC) **That the treasurer's report be accepted.** Carried (5-0).

14. KRE reported on a tentative agreement reached with EUUG on forming inter group links. Initial approaches have been made to JUG (Japan) and Usenix (USA) but no replies have been received to date. Letters to NZUSUGI and /usr/group have not yet been sent.

JL and CM have contact details for SINIX (Singapore) and KENJ has similar details for THAINIX (Thailand).

There were suggestions as to how AUUG should handle tape distributions, if at all. It was agreed that this would need to be on a cost recovery basis.

Moved (KENJ, seconded CM)

That the secretary be authorised to proceed and establish formal links with other groups such that

- i. **members of either group qualify for members discounts of the other group**
- ii. **each group make publications available to the other at cost price, and in whatever quantities are reasonable to minimise transport costs**
- iii. **there be a reciprocal exchange of distribution tapes as they become available**
- iv. **there be a reciprocal exchange of newsletters**

Carried (5/0).

15. It was noted that the group's financial year has in practice been from July 1 to June 30, whereas the constitution requires that it be from June 1 to May 31. It was agreed to alter the practice to conform with the constitution.
16. KENJ presented the tentative budget prepared by the budget sub-committee. He noted that the assumptions were:
- i. that the newsletter would cost \$42 per subscription per annum.
 - ii. that there would be no significant net change in membership numbers
 - iii. that there would be one management committee meeting other than in conjunction with an AUUG meeting
 - iv. that legal advice re incorporation would be obtained and that the cost would be as estimated
 - v. that meetings would approximately break even

The budget was tabled.

It was noted that ignoring the income from the May 86 Unixworld meeting, the budget was essentially for no net gain or loss.

CC suggested that the group must continue to grow, which implied that it must continue to spend more. It was noted that there was no requirement that the meetings do no more than break even, this was just an estimate.

17. CC and LS to consider possible expansion of AUUG activities, especially in the commercial area.
18. Moved (KRE, seconded JL) **That the budget be accepted** Carried (5/0).
19. There was some discussion on advertising in AUUGN, and what was easiest.
20. Moved (KENJ, seconded CC) **That the AUUGN editor have discretion to choose material likely to be of interest to AUUGN readers, subject to the following conditions**
 - i. **that only full page, A4, advertisements, received as camera ready monochrome copy be accepted**
 - ii. **that the rate be \$200 per page per issue**

Carried (5/0)

21. Moved (JL, seconded CC) **That legal advice in preparation of a ballot be sought.** Carried (5/0). The president and secretary were delegated to pursue this.
22. There was some discussion of future AUUG meetings. It was agreed that the group needs to plan these rather further in advance than had been past practice.
23. Moved CM (seconded CC)
That the schedule of future meetings be
February 1987, Adelaide (dates not yet available)
27-28 August 1987, Sydney
February 1988, Melbourne, or Geelong
August 1988, Sydney, or Newcastle
February 1989, Hobart, or Brisbane
Carried (5/0).
24. There was some discussion about the forthcoming meeting in Adelaide. It was noted that this had been arranged with very little notice after Telecom in Melbourne had indicated that they were unable to host the meeting. Despite the shortage of time the committee felt that a 2 day meeting was possible, and desirable. No decision on invited speakers was taken, though if someone suitable were available they could be invited.
25. JL indicated that he had received a cheque from Computerworld for AUUG's share of the May 86 Unixworld meeting. This had been passed to the treasurer and banked. There is some question as to the amount of the cheque. JL has

asked Computerworld for a financial statement, but this has not yet been received. LS to follow up.

26. Moved (KENJ, seconded KRE)

That in accordance with section 23 of the AUUG Constitution, it is hereby resolved that the General Rules covering the organisation, operation, obligations, and privileges of Chapters shall be:

- i. A Chapter committee shall conduct its affairs in a manner that is in keeping with the spirit of the AUUG Constitution and By-Laws.**
- ii. A chapter committee shall provide an annual report of its activities and operations to the Management Committee, prior to the AGM of the AUUG.**
- iii. The Management Committee reserves the right to dissolve a Chapter, thereby removing any association between the AUUG and the chapter committee.**

Carried (5/0).

27. The next committee meeting is scheduled to be held in conjunction with the February AUUG meeting, in Adelaide, February, 1987.

28. Meeting closed 16:38.

AUUG

Summer Meeting 1987

Adelaide, February 9 and 10.

The Summer 1987 AUUG meeting will be held in Adelaide, February 9 and 10 (Monday/Tuesday) 1987.

The meeting is being co-hosted by Austek Microsystems, and Vision Systems Ltd.

It will physically take place on the campus of the University of Adelaide. Accommodation will be available in one of the university colleges.

More detailed information about this conference will be posted to the newsgroup *aus.auug* in January 1987.

If you have no access to the electronic news then you should contact Wes Hosking, the conference organisation chairman. He can be reached as wes@vslvax.oz or (08) 349 5988. Paper mail will reach him at

Wes Hosking
Vision Systems Ltd
Technology Park
SA 5095
Australia.

We are now actively seeking papers for this conference. The programme chairman is Ian Richards from CSIRO DIT in Melbourne. Please send abstracts of papers to him at richards@ditmela.oz. The deadline for abstracts is Jan 9 1987. Sorry about the short notice, but the conference organisation has been delayed by a few setbacks (none caused by the organisers in Adelaide). Paper abstracts can be sent (if absolutely necessary) to

Dr Ian Richards,
CSIRO DIT,
55 Barry St,
Carlton,
Vic 3053.
Australia.

If you read this after January 9, then send an indication that you will be sending an abstract, and its title, as soon as possible. Ian can be contacted on (03) 347 8644.

EUUG

European UNIX[®]† systems User Group
Owles Hall, Buntingford, Herts. SG9 9PL, Great Britain
Tel (+44) 763 73039

Helsinki/Stockholm
Conference
1987

CALL FOR PAPERS
and
PRELIMINARY ANNOUNCEMENT

EUUG SPRING '87 CONFERENCE
UNIX GROWS UP

Helsinki (Finland) and Stockholm (Sweden), 11-14 May, 1987

Conference structure

The next EUUG Spring Conference will be held on board the M/S Mariella leaving from Helsinki on Tuesday 12th May 1987, arriving in Stockholm Wednesday 13th May and finally returning to Helsinki on Thursday 15th May. All sea crossings will be made at night. Tutorials will be held in the Hotel Dipoli in Helsinki on Monday 11th May 1987. The title of the conference is "Unix Grows Up".

Papers are solicited on the following topics.

Topics

- Quality Assurance,
- Real time processing using standard Unix,
- Process control/monitoring using Unix,
- Unix in the National environment,
- Unix on very big and very small machines,
- Unix in Banking,
- Future directions,
- R&D using Unix,
- Etc...

Tutorials

Tutorials will be held in the Hotel Dipoli on Monday 11th. Although the final tutorial programme has yet to be decided it is hoped that, amongst other speakers, Bjarne Stroustrup will be talking on C++

Exhibition

There will be a major exhibition at this conference, potential exhibitors are requested to contact the Secretariat for further information.

† Unix is a registered trademark of AT&T in the USA and other countries

Official Language

The official language of the conference will be English, no interpretation will be provided.

Paper submission

Abstracts should be submitted to the EUUG Secretariat **and** to the Programme Chairman, by ordinary and electronic mail (if possible in troff -ms form). They should include the following information:

- Title of paper,
- Author(s) name(s),
- Speaker name and affiliation (if more than one author),
- Mail address,
- Fax and/or telex number,
- Electronic mail address,
- Special audio-visual requirements if necessary,
- Text of abstract (in English): about 250 words.

Speakers of accepted papers will receive free entrance to the conference, including the conference dinner.

Deadlines

- **December 15th, 1986** : Abstract received by EUUG Secretariat and Programme Chairman.
- **January 1st, 1987** : Notification of acceptance or rejection by Programme Committee.
- **March 1st, 1987** : Final paper received by EUUG Secretariat and Programme Chairman for publication in the conference proceedings.

Programme Committee and Secretariat

Neil Todd (Chairman)
Imperial Software Technology
60 Albert Court
Prince Consort Road
London SW7 2BH
Great Britain
Tel: (+44) 1 581 8155
Fax: (+44) 1 581 5147
Telex: 928476
E-mail: mcvax!list!neil or ukc!list!neil

Jean Wood
DEC
BP29
Sophia Antipolis
06562
Valbonne Cedex
France
Tel: (+33) 93 65 51 11
E-mail: mcvax!unido!decum!jean

Johan Helsingius
Oy Penetron Ab
Box 21
SF-02171 Espoo
FINLAND
Tel: (+358) 0 427632
E-mail: mcvax!penet!julf

VUXPH Hans Albertson, NV 5
ERICSSON Information Systems i Sverige AB
Box 11100
S-161 11 Bromma
SWEDEN
Tel: (+46) 8 764 2616
E-mail: mcvax!erisun!halbirtsson

Mrs Helen Gibbons (Secretariat)
EUUG
Owles Hall
Buntingford, Herts. SG9 9PL
Great Britain.
Tel: (+44) 763 73039



AT&T Unix Pacific Co., Ltd.
No. 1 Nan-oh Bldg., 5th Fl.
2-21-2, Nishi-Shinbashi
Minato-ku, Tokyo 105 Japan
Tel : 03-431-3305
Telefax : 03-431-3680
Telex : J34936 ATTUP

October 13, 1986

Dear Sirs:

Thank you for your continued interest in AT&T.

\$ echo is published by AT&T UNIX* Software Licensing organization for licensees of UNIX Software Products. This Newsletter is designed to establish effective communication lines between AT&T and the UNIX community. It includes information on UNIX Software, new product announcements, policy changes and pricing structures. As you can see from the sample pages enclosed, we have modified some of the contents and added some articles for the Asia/Pacific area. We have recently expanded \$ echo to include more technical and informative papers and are hoping to publish papers submitted by our readers soon.

We distribute a free copy of \$ echo on a quarterly basis to every source code licensee. From the March 1986 issue, we also started subscriptions for non-licensees and provided the ability to purchase additional copies for a reduced rate. Subscription rate is US\$50 (10,000 yen) per year plus US\$20 Air Mail Charge for outside Japan. Additional copy rate is US\$25 (5,000 yen) per year.

If you wish to receive a copy, please fill out the order form and return it with a bank draft payable to AT&T Unix Pacific Co., Ltd. Bank-wire transfer is also available.

If you could pass a copy of this information on to others who you feel might be interested, such as members of your UNIX Users' Group or if you could provide us with a membership list for direct mailing, we would be appreciative.

Sincerely,

Ryerson E. Schwark
Account Executive
Asia/Pacific Region

Tokyo-Japan-RS-ky

Encs.

* UNIX is a registered trademark of AT&T in the USA and other countries.



\$ echo ORDER FORM

AT&T Unix Pacific Co., Ltd.
No. 1 Nan-oh Building, 5th Floor
2-21-2 Nishi-Shinbashi
Minato-ku, Tokyo 105, Japan

Telephone: 81-3-431-3305 (Japanese)
81-3-431-3670 (English)
Facsimile: 81-3-431-3680
Telex: J34936 ATTUP

To order, please send a bank draft with this ORDER FORM to the above address.
Bank wire transfer is also available:

The Fuji Bank, Ltd. Shinbashi Branch
Account of AT&T Unix Pacific Co., Ltd.
A/C No.: 25201 Current Account (Toza)
2-1-3 Shinbashi, Minato-ku
Tokyo 100 Japan

Base subscription fee is US\$50 (10,000 yen) per year.
Additional copy fee is US\$25 (5,000 yen) per year for one subscriber.
For overseas orders, add US\$20 for air mail postage.

-
- Please send a subscription to the address below.
 - Please send _____ copy(s) of \$ echo to the address below.
 - from March 1986 issue
 - from September 1986 issue
 - from next issue

Name: _____

Company: _____

Address: _____

Tel: _____

Telex: _____ Fax: _____

UUCP: _____

UNIX TRADEMARK REGISTERED

AT&T is pleased to announce the official registration of the UNIX trademark, effective May 6, 1986. Attached is a set of guidelines for the correct usage of the trademark.

USE OF THE TRADEMARK UNIX¹

UNIX is a registered trademark of AT&T used to identify its particular brand of software. The trademark is used in conjunction with several time-sharing operating systems developed at AT&T Bell Laboratories and licensed by AT&T, and might be used in the future on other kinds of software and products.

A trademark identifies the source of a product. Some trademark owners license their trademarks for use by others. A product marked with such a trademark might come from either the trademark owner or from one of its licensees. However, it is AT&T's policy not to license parties outside the company to use the trademark UNIX to identify their products. There are specific provisions in our software agreements for UNIX operating systems on this point.

Notwithstanding this policy, anyone may use the trademark UNIX to refer to the UNIX operating systems developed at AT&T Bell Laboratories. However, to protect AT&T's interest in the trademark, we must ask that others use the trademark correctly. Following are several comments on correct and incorrect use of the trademark. The comments are organized in outline form for convenient reference.

A. Trademark Appearance

1. The trademark UNIX must always appear in a form that is typographically distinct.
2. The trademark UNIX must be clearly and legibly identified as a trademark of AT&T at least once in any article, advertisement or document in which the trademark appears, preferably the first time such trademark is used.
3. The trademark UNIX is a **registered** trademark of AT&T. It is **correct** to use the symbol “®” in connection with the trademark UNIX or to state that UNIX is a registered trademark.

B. Outside Parties

1. Parties outside AT&T may not state or imply that they furnish UNIX operating systems to others and may not use the trademark UNIX in the name of software that they furnish to others. Even if such parties are licensed by AT&T to use UNIX operating systems or to furnish object code derived from such operating systems to others, they are **not** licensed to use the trademark to identify their product.
2. The trademark UNIX may not be used in the name of a publication, business or other organization (such as a user group).

C. Grammatical Usage

1. The trademark UNIX may not be used as a noun, but must always be used as an adjective modifying a common noun as in “UNIX operating system.”
2. The trademark UNIX must always be used to modify a common name for something that is a product with which the trademark is used. For example, it is incorrect to refer to “a UNIX user,” “UNIX terminals” or “UNIX support.” Correct usage is “a user of UNIX operating system,” “terminal on a computer running a UNIX operating system” or “support for UNIX operating system”.

A way to check whether usage of the trademark is correct is to mentally insert the word “Brand” between the trademark and the common name. “UNIX Brand operating system” sounds reasonable but “UNIX Brand user” does not.

3. The trademark UNIX may not be used in a hyphenated expression such as “UNIX-based” or “UNIX-like.”
4. The trademark UNIX may not be combined with the trademark of another party unless the independence of the trademark is clear.

USE OF THE TRADEMARK UNIX¹ - Continued

D. Official Names

1. The official names of the various UNIX operating systems that are currently licensed are:

UNIX Time Sharing Operating System, Sixth Edition, (“V6” or “Sixth Edition”)

UNIX Time-Sharing System, Seventh Edition (“V7” or “Seventh Edition”)

PWB/UNIX Time Sharing Operating System, Edition 1.0 (“PWB”)

UNIX/32V Time-Sharing System, Version 1.0 (“32V”)

Mini-UNIX Time Sharing Operating System, Version 6 (“Mini-UNIX System”)

UNIX System III (“System III”)

UNIX Time-Sharing System for UNIVAC² 1100 Series Systems - Version 7 (“1100 System”)

UNIX System V (“System V”)

UNIX System V, Release 2.0 Version 1

UNIX System V, Release 2.0 VAX³ 11/780 Version 2

UNIX System V, Release 2.0 AT&T 3B20 Version 3

UNIX System V, Release 2.0 AT&T 3B20 Version 4

UNIX System V/M68000

UNIX System V/iAPX 286 Version, Release 2.0

UNIX System V, Release 2.0 NSC32000 Version 1

UNIX System V, Release 2.1

UNIX System V Japanese Application Environment

The abbreviations in parentheses following the official names are suggested for use if a name must be repeated. The trademark UNIX must be used correctly even if it is used in an abbreviation. The terms Mini-UNIX, PWB/UNIX and UNIX/32V are not considered to be separate trademarks from the trademark UNIX, and should therefore be treated in a similar manner to the trademark UNIX.

2. Reference to “the UNIX operating system” is inappropriate. There are several UNIX operating systems (see list in D1.) For a collective term, use “UNIX operating systems,” if that is what is meant.
3. It is inappropriate to use the trademark UNIX in any label (such as file name, subroutine call or the like) in any software.

E. Internal Written Material

Use the “®” superscript the first time UNIX appears in the document.

F. External Written Material

1. United States

Where the context of written material intended for external use in the United States makes clear that the source of the material is AT&T, or an AT&T subsidiary, then use of the “®” superscript after the first appearance of the trademark is sufficient. Where it is not clear that the source is AT&T, or an AT&T subsidiary, then the first appearance of the trademark should be footnoted as follows:

“Registered trademark of AT&T”

2. International (Except Japan)

For written material intended for use abroad (except Japan), the first appearance of the trademark should be footnoted as follows:

“Registered trademark of AT&T in the USA and other countries”

3. Japan

The following notice, rather than a trademark notice, should be used in Japan as a footnote in association with the first appearance of the trademark:

“Developed and Licensed by AT&T”

No material should be used in Japan which carries any of the trademark notices discussed above. Any existing trademark identification must be expunged from all material before being used in Japan.

G. Use in a Company Name

The appearance of a trademark in a Company name such as AT&T Unix Europe Limited or AT&T Unix Pacific Co., Ltd., is not a trademark usage as such, and need not be identified by the symbol “®” or a footnote.

H. Other Trademarks

These guidelines are also applicable to identification of other unregistered trademarks, such as “*Writer's Workbench*’.” However, these trademarks use the “™” superscript (internal) or one accompanied by the following footnote:

“Trademark of AT&T”.

The following are trademarks used in this document:

- ¹ UNIX is a registered trademark of AT&T.
- ² UNIVAC is a trademark of Sperry Corporation.
- ³ DEC and VAX and PDP are trademarks of Digital Equipment Corporation.
- ⁴ *Writer's Workbench* is a trademark of AT&T.

AUUG

Australian UNIX* systems User Group.

P.O. Box 366, Kensington NSW 2033, Australia.

auug@munnarl.oz.au {selsmo,hplabs,mcvax,ukc}!munnarl!auug

***UNIX is a registered trademark of AT&T in the USA and other countries.**

Friday 12th December, 1986

John Carey,
AUUGN Editor.

Dear John,

The management committee of AUUG has noted the great delay between the anticipated date of issue of each edition of AUUGN, and its actual appearance.

The committee appreciates that this delay occurred before you were appointed to the editorship, however we feel that something needs to be done to remedy the situation.

We have reached the conclusion that it would be unreasonable to expect that more issues of AUUGN can possibly be produced than the anticipated one every 2 months, so we have, with some regret, decided that we must abandon the missing issues.

You should point out to the readers that the issues being abandoned are ones that they have already failed to receive. In the future we expect that the schedule of one newsletter every 2 months will be maintained.

You should also point out, that in order to maintain this schedule, the committee has requested you to publish each issue at its due date, regardless of whether or not any material has been submitted for publication.

AUUGN is a newsletter of the members, if they fail to contribute, then there is nothing that either you, as editor, or we, as the committee, can do to remedy that.

The committee examined several ways to achieve our aim of bringing the AUUGN volume numbering back into line with the calendar year.

We eventually resolved that each future AUUGN issue should carry two issue numbers until this has been effected.

Issues should continue to appear every 2 months.

We would be grateful if you would put this policy into effect, commencing with the forthcoming issue.

Yours sincerely,

Robert Elz
Honorary Secretary
AUUG

AUUG

Membership Categories

Now that the Australian UNIX systems User's Group has existed a while, its time that all members reviewed their membership types, and even more, checked that they are in fact still members!

There are 4 membership types, plus a newsletter subscription, any of which might be just right for you.

The membership categories are:

- Institutional Member
- Ordinary Member
- Student Member
- Honorary Life Member

Institutional memberships are primarily intended for university departments, companies, etc. This is a voting membership (one vote), which receives two copies of the newsletter. Institutional members can also delegate 2 representatives to attend AUUG meetings at members rates. AUUG is also keeping track of the licence status of institutional members. If, at some future date, we are able to offer a software tape distribution service, this would be available only to institutional members, whose relevant licences can be verified.

If your institution is not an institutional member, isn't it about time it became one?

Ordinary memberships are for individuals. This is also a voting membership (one vote), which receives a single copy of the newsletter. A primary difference from Institutional Membership is that the benefits of Ordinary Membership apply to the named member only. That is, only the member can obtain discounts on attendance at AUUG meetings, etc, sending a representative isn't permitted.

Are you an AUUG member?

Student Memberships are for full time students at recognised academic institutions. This is a non voting membership which receives a single copy of the newsletter. Otherwise the benefits are as for Ordinary Members.

Honorary Life Memberships are a category that isn't relevant yet. This membership you can't apply for, you must be elected to it. What's more, you must have been a member for at least 5 years before being elected. Since AUUG is only just over 2 years old, there is no-one eligible for this membership category yet.

Its also possible to subscribe to the newsletter without being an AUUG member. This saves you nothing financially, that is, the subscription price is the same as the membership dues. However, it might be appropriate for libraries, etc, which simply want copies of AUUGN to help fill their shelves, and have no actual interest in the

contents, or the association.

Subscriptions are also available to members who have a need for more copies of AUUGN than their membership provides.

To find out if you are currently really an AUUG member, examine the mailing label of this AUUGN. In the lower right corner you will find information about your current membership status. The first letter is your membership type code, N for regular members, S for students, and I for institutions. Then follows your membership expiration date, in the format exp=MM/YY. The remaining information is for internal use:

If your membership has already expired, or is about to expire (many expire in January) then now is the time to renew.

If you want to join AUUG, or renew your membership, you will find forms in this issue of AUUGN. Send the appropriate form (with remittance) to the address indicated on it, and your membership will (re-)commence.

Robert Elz

AUUG Secretary.

AUUG

Application for Ordinary, or Student, Membership Australian UNIX* systems Users' Group.

*UNIX is a registered trademark of AT&T in the USA and other countries.

To apply for membership of the AUUG, complete this form, and return it with payment in Australian Dollars to:

AUUG Membership Secretary
PO Box 366
Kensington NSW 2033
Australia

Please don't send purchase orders — perhaps your purchasing department will consider this form an invoice. Foreign applicants please send a bank draft drawn on an Australian bank, and remember to select either surface or air mail.

I, do hereby apply for

- Renewal (indicate which membership type).
- Membership of the AUUG \$ 50.00
- Student Membership of the AUUG \$ 30.00 (note certification on other side)
- International Surface Mail \$ 10.00
- International Air Mail \$ 50.00

Total remitted

AUD\$ _____
(cheque, money order)

I agree that this membership will be subject to the rules and by-laws of the AUUG as in force from time to time, and that this membership will run for 12 consecutive months commencing on the first day of the month following that during which this application is processed.

I understand that membership includes a subscription to the AUUG newsletter, and that I will be entitled to attend AUUG sponsored functions at member rates for the duration of my membership.

Date: ___/___/___

Signed: _____

- Tick this box if you wish your name & address withheld from mailing lists made available to vendors.

For our mailing database - please type or print clearly:

Name:

Phone: (bh)

Address:

..... (ah)

.....
.....
.....
.....

Net Address:

Write "Unchanged" if details have not altered and this is a renewal.

Student Member Certification *(to be completed by a member of the academic staff)*

I, certify that
..... *(name)*
is a full time student at *(institution)*
and is expected to graduate approximately ____/____/____.

Title:

Signature:

Office use only:

Chq: bank _____ bsb _____ - _____ a/c _____ # _____

Date: ____/____/____ \$

Who: _____

Memb# _____

AUUG

Application for Institutional Membership Australian UNIX* systems Users' Group.

*UNIX is a registered trademark of AT&T in the USA and other countries.

To apply for institutional membership of the AUUG, complete this form, and return it with payment in Australian Dollars to:

AUUG Membership Secretary
PO Box 366
Kensington NSW 2033
Australia

Foreign applicants please send a bank draft drawn on an Australian bank, and remember to select either surface or air mail.

..... does hereby apply for

- Renewal of existing Institutional Membership \$250.00
- New Institutional Membership of the AUUG \$250.00
- International Surface Mail \$ 20.00
- International Air Mail \$100.00

Total remitted

AUD\$ _____
(cheque, money order)

I agree that this membership will be subject to the rules and by-laws of the AUUG as in force from time to time, and that this membership will run for 12 consecutive months commencing on the first day of the month following that during which this application is processed.

I understand that I will receive two copies of the AUUG newsletter, and may send 2 representatives to AUUG sponsored events at member rates, though I will have only one vote in AUUG elections, and other ballots as required.

Date: ___/___/___

Signed: _____

Title: _____

Tick this box if you wish your name & address withheld from mailing lists made available to vendors.

For our mailing database - please type or print clearly:

Administrative contact, and formal representative:

Name: Phone: (bh)

Address: (ah)

..... Net Address:

..... Write "Unchanged" if details have not altered and this is a renewal.

Please complete the other side.

Please send newsletters to the following addresses:

Name:
Address:
.....
.....
.....
.....

Name:
Address:
.....
.....
.....
.....

Write "unchanged" if this is a renewal, and details are not to be altered.

Please indicate which Unix licences you hold, and include copies of the title and signature pages of each, if these have not been sent previously.

Note: Recent licences usually revoke earlier ones, please indicate only licences which are current, and indicate any which have been revoked since your last membership form was submitted.

Note: Most binary licensees will have a System III or System V (of one variant or another) binary licence, even if the system supplied by your vendor is based upon V7 or 4BSD. There is no such thing as a BSD binary licence, and V7 binary licences were very rare, and expensive.

- | | |
|---|--|
| <input type="checkbox"/> System V.3 source | <input type="checkbox"/> System V.3 binary |
| <input type="checkbox"/> System V.2 source | <input type="checkbox"/> System V.2 binary |
| <input type="checkbox"/> System V source | <input type="checkbox"/> System V binary |
| <input type="checkbox"/> System III source | <input type="checkbox"/> System III binary |
| <input type="checkbox"/> 4.2 or 4.3 BSD source | |
| <input type="checkbox"/> 4.1 BSD source | |
| <input type="checkbox"/> V7 source | |
| <input type="checkbox"/> Other (Indicate which) | |

Office use only:

Chq: bank _____ bsb _____ - _____ a/c _____ # _____

Date: ___/___/___ \$

Who: _____

Mem# _____

AUUG

Application for Newsletter Subscription Australian UNIX* systems User Group.

*UNIX is a registered trademark of AT&T in the USA and other countries.

Non members who wish to apply for a subscription to the Australian UNIX systems User Group Newsletter, or members who desire additional subscriptions, should complete this form and return it to:

AUUG Membership Secretary
PO Box 366
Kensington NSW 2033
Australia

Please don't send purchase orders — perhaps your purchasing department will consider this form an invoice. Foreign applicants please send a bank draft drawn on an Australian bank, and remember to select either surface or air mail.

Please *enter / renew* my subscription for the Australian UNIX systems User Group Newsletter, as follows:

Name: Phone: (bh)

Address: (ah)

.....

..... Net Address:

.....

.....

.....

Write "Unchanged" if address has not altered and this is a renewal.

For each copy requested, I enclose:

- Subscription to AUUGN \$ 50.00
- International Surface Mail \$ 10.00
- International Air Mail \$ 50.00

Copies requested _____

Total remitted _____

AUD\$ _____

(cheque, money order)

Tick this box if you wish your name & address withheld from mailing lists made available to vendors.

Office use only:

Chq: bank _____ bsb _____ - _____ a/c _____ # _____

Date: ___ / ___ / ___ \$

Who: _____

Subscr# _____

AUUG

Notification of Change of Address Australian UNIX* systems Users' Group.

*UNIX is a registered trademark of AT&T in the USA and other countries.

If you have changed your mailing address, please complete this form, and return it to:

AUUG Membership Secretary
PO Box 366
Kensington NSW 2033
Australia

Please allow at least 4 weeks for the change of address to take effect.

Old address (or attach a mailing label)

Name: Phone: (bh)
Address: (ah)
.....
Net Address:
.....
.....
.....

New address (leave unaltered details blank)

Name: Phone: (bh)
Address: (ah)
.....
Net Address:
.....
.....
.....

Office use only:

Date: ___ / ___ / ___

Who: _____

Mem# _____