AUUGN

Australian Unix systems User Group Newsletter

Volume 6
Number 1

Registered by Australia Post Publication No. NBG6524

The Australian UNIX* systems User Group Newsletter

Volume 6 Number 1

May 1985

CONTENTS

* UNIX is a trademark of AT&T Bell Laboratories.

Editorial

Welcome to volume 6 and my apologies for the lateness of this issue. I thought of making all sorts of excuses, but then I thought "why should I!", so I won't.

This issue should be followed by another next month and then we should be back in sync (small UNIX joke there).

WANTED: AUUGN Columnists

Is there anyone out there that wants to share the fame, wealth and wild living of being a major contributor to this newsletter? Seriously, I am finding the job a little too much for one person and am looking for people, interested in producing a good quality publication, willing to share the load. Perhaps you could take over the preparation of some of the regular sections of the newsletter or just offer general assistance.

If you are interested, contact me as soon as possible. The address and phone number appear on the last page of this issue.

Memberships and Subscriptions

Membership and Subscription forms may be found at the end of this issue and all correspondence should be addressed to

Greg Rose
Honorary Secretary, AUUG
PO Box 366
Kensington NSW 2033
Australia

Next AUUG Meeting

The next AUUG Meeting will be held in Queensland at the University of Queensland on the 26th and 27th of August 1985. Further information appears later in this issue.

Contributions

Come on you people out there in UNIX-land, send me your views, ideas, gripes, likes or whatevers.

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

Advance notice - Winter 1985

AUUG Meeting in Brisbane

The Winter '85 AUUG meeting will be held in sunny, strike-free Brisbane. Your host is the sunny, strike-free Computer Science Department of the University of Queensland. Keynote speaker will be Stu Feldman, author of "make" and the first F77 compiler.

The meeting will be held on Monday 26th and Tuesday 27th August on the University Campus. College accomodation on campus will be available.

The format of the meeting will be similar to previous gatherings, with keynote addresses, papers and "birds of a feather" sessions on Monday, the conference dinner Monday evening, then further papers and tutorial sessions on Tuesday. There will also be an equipment exhibition.

For further information, send electronic mail to auugm@uqcspe.oz, or contact Tim Roper ((07)377-2875) or Peter Barnes ((07)377-4139) at

The Department of Computer Science,
University of Queensland,
St. Lucia QLD 4067.

Start writing your abstracts and papers NOW!

Australian Unix systems User Group
1985 Summer Meeting
University of Wollongong
Programme

Monday, February 11, 1985

09:00-10:30	Registration	
10:00-10:30	Morning Tea	
10:30-10:40	Opening Remarks	Juris Reinfelds *Department of Computing Science, University of Wollongong*
10:40-11:40	Keynote Address "Portability Reconsidered"	Richard Miller *R. Miller Associates*
11:40-12:10	Using Loosely Coupled Processors	Tim Long *Fawnray-Prance*
12:10-13:30	Lunch	
13:30-14:00	Troff Output Previewing	James Ashton *University of Wollongong*
14:00-14:30	Unix - a Tool for Research in the Steel Industry	David Sterling *John Lysaghts Australia*
14:30-15:00	Design of TODAY - a 4GL under Unix	Allan Davies *Product Manager for TODAY, BBJ Computer Services*
15:00-15:30	Afternoon Tea	
15:30-16:10	How to Speed Up File Name Access	Greg Rose, *Fawnray-Prance*
16:10-17:00	Panel Session "User Experiences with Super-Mini Computers"	
19:30-	Dinner	

UNIX is a trademark of AT&T Bell Labs.

Tuesday, February 12, 1985

09:00-09:30	Unix and IBM	Jack Brown, *Manager of Integration, Engineering/Scietific Systems* *IBM System Products Division* *USA*
09:30-10:00	AUUG Business Session	
10:00-10:30	Morning Tea	
10:30-11:00	PORT - A Different Approach	Gary Stafford, *Department of Computing Science,* *University of Wollongong*
11:00-11:15	Virtual Memory Management for System V Unix	Richard Miller, *R. Miller Associates*
11:15-11:30	Unix and the STD Bus	Charles Brady, *Telectronics*
11:30-12:00	CSPELL - a Program to Check and Interactively Correct Spelling	Roy Rankin, *University of Sydney*
12:00-12:15	A Document Processing Package	Terry Reilly, *NCR, Adelaide*
12:15-12:30	FRIEND - a 4GL under Unix	Kim Sadlier, *Sadlier Ltd.*
12:30-13:30	Lunch	
13:30-15:00	Tutorials	
	Tutorial #1 Networking	Piers Lauder, *University of Sydney* Robert Elz, *University of Melbourne*
	Tutorial #2 Assessing Interactive Programs via Batch Processing	John Lions, *University of NSW*
15:00-15:30	Afternoon Tea	
15:30-	Birds of a Feather Sessions	

Using Loosely Coupled Processors

Tim Long
Fawnray-Prance

The ELXSI 6400 is a 4 MIP per CPU loosely coupled multiprocessor to which UNIX was recently ported by a group of people including myself.

This talk is not so much about the port of UNIX to the ELXSI but about some observations on how to and how not to utilise a configuration of loosely coupled processors.

Troff Output Previewing

James Ashton
Department of Computing Science,
University of Wollongong
and
Australian Iron and Steel

A system for previewing output from the *troff* document processing system under Unix is discussed. Due to the high cost of materials used in the preparation of output from a phototypesetter, a low cost, fast method of output drafting is required. The drafting system should reproduce the document such that font size changes and all motions are reproduced exactly, thus indicating exactly how the final output will appear. Font changes may be limited to Regular and Italic, since as a general rule, when drafting for appearance, it is unnecessary to reproduce each character in each font exactly. An approximation is sufficient.

The system as implemented runs under Unix, accepts input from troff, and prepares output for a variety of bit-mapped output devices. Examples of output devices are an ICL Perq and an Apple Imagewriter printer.

The system employs a polygonal representation of characters in a font, and a learning system for reproducing characters as they are set. A scan converter produces a bit-map suitable for display from the polygonal representation in the correct size. Problems in conversion for small sizes are discussed.

The system comes complete with a mouse/puck based editor for creating the character representations for output drafting.

The system has been used successfully to draft a 600 page book, along with a number of smaller papers and documents.

Unix - A Tool for Research in the Steel Industry

David Sterling,
Research and Technology Centre,
John Lysaghts Australia.

The Research and Technology Centre at Lysaghts is carrying out studies as to the effective use of a Unix system as a research and production tool in the steel industry. One of the main themes is the use of the text processing facillities for the development, operation and mainenance of various suites of programs, most of which were written on a previous system at RTC. Examples of software covered by this are systems for CAD/CAM, fluid dynamics, analysis and simulation of hot and cold steel rolling mills, annealing processes, paint line processes and coil winding analysis.

A graphics system based on software provided by the vendor of the system is used to advantage in the display of results.

The system has also been used in the support and development of a furnace automation project, currently being commissioned at Port Kembla. Two other pending applications within RTC may see the introduction of additional Unix systems. These are for signal processing analysis and control of an XPS/SAM surface analysis machine, and a system for software development on Motorola 68000 systems.

Design of TODAY, a 4GL Under Unix

Allan Davies,
'TODAY' Product Manager,
BBJ Computer Services.

The design and development of a fourth generation language system for Unix systems is discussed. A review of the actual project history and the features of 'TODAY' are presented.

How to Speed up File Name Access
or
A Lesson in Programming.

Greg Rose,
Fawnray-Prance

During the course of 1984, a port of Unix System V was carried out to the Elxsi 6400 Multiprocessor.

During the port of the kernel, it was discovered that ~25% of kernel execution time was spent in the routine namei(), responsible for converting a filesystem path name into a pointer to an incore inode structure. The time spent was reduced to ~5% by the application of two methods. The firt method was the circular lookup of filenames within directories. The second method was the hashing of path name segments.

This paper describes the hashing technique used, called "corroboartive hashing", and the resultant performance enhancements. The success of the new hashing scheme is illustrated by collected results.

PORT - A Different Approach

Gary Stafford,
Department of Computing Science,
University of Wollongong.

Port is an Operating System which was designed for use on personal computers. It evolved from a time-sharing system (THOTH) much like Unix and as such kept many of the desirable features in the transformation. Some of the major differences from Unix will be mentioned, notably message-passing, file system structure, and internal structuring of the system based on the concept that processes are inexpensive.

Time permitting, a few features of the programming language in which it is written, (also called Port, to avoid confusion), will be mentioned.

Unix and the STD Bus
or
Squeezing Blood out of a Stone.

Charlie Brady,
Basser Department of Computing Science,
University of Sydney
and
Telectronics Pty Ltd.

The STD bus is a general purpose 8-bit microprocessor bus created to provide a "modular-by-function" approach to control oriented system design. Although successful in industrial control situations, the small card size and simplicity of design of the STD bus have led to considerable success in the general microprocessor market. Naturally, market demands for Unix have been recognised, and answered by many manufacturers.

Some of the techical difficulties of running Unix on a system with an 8-bit data bus and a maximum address space of 128k bytes, with two levels of interrupt and one level of bus access arbitration and a board size of 114*165mm (4.5*6.5") are discussed. Rationale behind the questions why and how are discussed, in view of the availability of other more suitable bus structures.

CSPELL - a Program to Check and Interactively Correct Spelling

Roy Rankin,
School of Electrical Engineering,
University of Sydney

CSPELL is a program to check and interactively corrent spelling. This system, when encountering unknown words searches for possible alternative spellings which can then be interactively selected by the user.

The internal workings of CSPELL are described, including the dictionary lookup algorithm, the unknown word search heuristics, the overall performance of the system, and advantages and disadvantages.

A Tutorial on Networking under Unix

Piers Lauder
Basser Department of Computing Science,
University of Sydney

Robert Elz,
University of Melbourne.

A tutorial on various aspects of networking and Unix is presented. Topics discussed are mail delivery and receipt; storage and management systems for mail; delivery across heterogeneous networks. General principles of networking; store-and-forward systems; routing and switching; name selection and domains. The operation of ACSnet. Aspects of its installation, monitoring, maintenance and use; tailoring to suit individual sites; message handlers; statistics interpretation. Access to the various US networks, and sites within them; interfacing to UUCP; communications media; auto call units; intermittent connections and permanent circuits.

UNIX as a Tool for Research in the Steel Industry

D.A.Stirling : Research & Technology Centre
John Lysaght (Australia) Ltd.

The Research and Technology Centre (RTC) at John Lysaght (Aust.) Ltd. installed a new research computer (Oct. '83) running UNIX. Some observations and comments are made on both UNIX and the computer hardware (HP 9000) as a satisfactory tool for research in the Steel Industry.

Researchers in RTC include chemists, metallurgists, material scientists, mathematicians, physicists, mechanical and electrical engineers. Their research interests cover a wide range of areas such as metallurgy, welding, electrochemistry, surface analysis, corrosion studies, paint aging, hot and cold rolling, annealing, galvanising, roll forming, tension levelling, paint oven modelling and high technology instrumentation.

A high proportion of work in the above areas is connected with mathematical modelling, analysis and simulation which requires locally developed software. The collection, manipulation and analysis of data from plant trials and similar activities is also a significant part of the above areas of research.

The new RTC research computer is a Hewlett Packard HP 9040s with dual 32 bit CPU's, 2 megabytes of main memory, 5 terminals including two graphics (one colour), 1600 bpi 1/2" streaming magnetic tape and 132 megabyte disc. The HP-UX release 2.0 approximates System III UNIX with HP's paging and virtual memory plus a number of Berkeley Enhancements.

In March 1984, release 3.0 was received which gave approximately an overall 20% speed improvement. A further update, release 4.0 added a number of UNIX commands missing in the previous releases. Release 4.0 has also given incremental speed improvements particularly on interactive processes. One important inclusion is a much needed debugger similar to 'sdb' on VAX's. This last update is targeted on AT&T System V release 2.0 with Berkeley and other enhancements.

One immediate benefit, that was recognised and used soon after the system arrived at RTC, was the power and ease of shell programming. Unlike the previous RTC computers' command file syntax, shell programming even looked like a programming language. A simple yet effective shell program was written to automate the file transfers from the old RTC computer system to the HP 9000. This program used 'cmp', 'sed', 'mv' and 'cp' with additional terminal emulation software. The total volume of files transferred was approximately 86 Mb., each being verified with a repeat transmission.

The UNIX file system offered research officers a rational and structural means of organising their software not hitherto experienced in RTC. The addition of the magnetic tape unit made automated incremental daily backup using 'cpio' an added security for software development. Backup on the previous system was typically done on a monthly basis and only on a limited proportion of the disc packs used.

Once all the desired software, being mostly Fortran, was transferred, a range of UNIX utilities was employed to re-install a major proportion of it in a minimal time frame. These utilities basically typically were : shell scripts, 'grep', 'diff', 'sed', 'wc' and 'fc -L'.

As time has progressed various users of the system have adapted the available tools in UNIX for integrating, running and maintaining their software. One in particular, is pre-processing with 'awk' to integrate not readily compatible programs with different format requirements. In addition 'diff', 'sed', 'grep', 'wc' and SCCS are used to assist in the regular house keeping of researchers' software, for example coping with n versions of the same program.

Often new research or development is based on some previous work which has a certain amount of software associated with it. It becomes laborious and almost prohibitive when the previous software has no documentation or even comments in it. To effect in some measure a prevention of such situations, RTC has developed a documentation program 'prod' which will produce useful documentation on Fortran programs. This includes subroutine and common block cross referencing, a subroutine access map and a variable usage table (both type and case). RTC hopes to expand and generalise 'prod' to cope with other languages as the need arises.

Graphical representation of results is an important consideration in any research environment, and often a time consuming one. RTC have developed 'PLOT2' under the HP-UX system to address this need. 'PLOT2' more than covers the range of graphical representations used for the different research areas, with numerous options, combinations and techniques for displaying the data.

Two specific areas of research and development which have made use of a number of UNIX tools are, firstly, the Tight Coil Annealing Project, and secondly, a CAD/CAM package 'CADROF', used in toolage design for roll formers. Software for the former is being developed on two other computers, neither of which support UNIX. Periodic samples of the code are transferred to the RTC/9000 where they are kept as backups in a SCCS file. This technique also has proved useful in identifying some critical changes that were made between two early versions of the software, by using 'diff'. The partially developed software in 'CADROF' was ported from an HP1000 and further developed under UNIX using a number of tools in UNIX previously mentioned.

Networking

The organisation, gathering and analysis of data is a significant proportion of most activities. In relationship to data trials and tests, RTC expects that a pending local area network (broad band ethernet) will partly address this issue. In addition, two pending systems which would support UNIX within RTC may be networked to the RTC:9000. This would assist in data acquisition and development in general.

Further afield, researchers at RTC would benefit by access to the ACS network and already existing contacts with a number of Universities would be further enhanced.

Conclusions

It has been the general experience in RTC so far, that the various tools in UNIX can be utilised reasonably quickly by non-gurus both to install existing software, and to maintain and develop the same, and or new software for research.

Users within RTC have also adapted the UNIX environment to various degrees to suit their own research activities and temperament. This flexibility is seen as an added benefit of the operating system.

The last and perhaps obvious benefit to RTC in using UNIX is the potential to retain developed skills in the event of another computer system change.

User-Mode Development
Of Hardware and Kernel Software

Robert P. Warnock, III

Fortune Systems Corporation
Redwood City, California 94061

As a general rule, the development of new hardware devices, operating systems drivers for those devices, and other new operating systems functions is considerably more difficult than the development of user-mode functions of similar complexity. Several factors contribute to this: hardware often doesn't work as initially expected (despite documentation); testing drivers and other kernel functions requires a very scarce resource - standalone time on the system; errors often leave the entire system hung or halted with no history trace, making crash analysis a challenge at best; the edit-compile-load cycle tends to be longer and more complex; and a logic analyzer is seldom the most convenient diagnostic tool.

A set of techniques or "tricks" are presented, with examples of their application. While each one may be "obvious" by itself, and not particularly related to the others, together they illustrate a common principle and general method. The principle is that of separation of concerns, together with addressing those concerns in the proper order. "First make it work correctly; then make it work well while remaining correct." The general method is to do the development in user-mode software, using minimal "hooks" to make this possible. Then, after the functionality has been demonstrated and the critical algorithms debugged, the software is "ported" to kernel mode as necessary to attain the required performance goals.

Other authors [Holt] [Wulf] have suggested, in fact, that the "kernel" of an operating system should be quite tiny (a few hundred lines of assembler), and that ALL of what one normally thinks of as the "operating system" should be run in user-mode, including device drivers, file systems, and schedulers. Unfortunately, most of us do not have the freedom to make major modifications to our operating system environment (typically UNIX of some flavor or other). The examples given demonstrate that, at least during initial development, it is possible to obtain the benefits of the "user-mode style" even though the production version may be completely traditional in structure.

The development projects used as examples took place at Fortune Systems between Summer 1982 and Summer 1984, and include:

[Holt] R. C. Holt, Concurrent Euclid, The UNIX System, and Tunis, Addison-Wesley, 1983

[Wulf] William A. Wulf, Roy Levin, and Samuel P. Harbison, HYDRA/C.mmp, McGraw-Hill, 1981

1. A byte-parallel file-transfer link was implemented between a DEC VAX-11/780 and a Fortune Systems 32:16. The VAX driver was developed in user mode using /dev/kUmem to access the hardware. The 32:16 driver was developed in user mode using the "sysphys" feature (UNIX Edition 7 "phys(2)" call) to map the user addresses to the hardware. After the file-transfer application was completely functional, the VAX driver was moved to the kernel, with a 25-fold improvement in performance. (The 32:16 driver was left in user-mode permanently.)

2. A communications co-processor for the 32:16 was debugged using user-mode software (again using "sysphys"). When the UNIX driver was being debugged, host-resident user-mode code was used to mimic the co-processor application on the one hand, while making calls to the driver and comparing the results on the other. A similar procedure was used in developing a bit-mapped graphics controller and a parallel-I/O co-processor.

3. A set of library subroutines was written to allow user-mode emulation of (proposed) new operating system calls. When the "system call" was invoked, instead of entering the normal (kernel-mode) system call handler, a call-request packet was passed through a "pty" to a daemon program which emulated the call and passed a "return value" packet back through the pty. Packet types were provided to allow the daemon to read and write the client process's address space (as the kernel would have been able to do).

   This facility was used to develop a network "socket" mechanism (similar to 4.2bsd sockets). A "network line discipline" was implemented using ordinary terminal ports as network devices. After the internet router and network line discipline were completely functional running in user mode as a system-call emulation daemons (including actually transmitting packets over a multi-host net), they were "ported" straightforwardly into the kernel.

4. In the previous hardware examples, the physical device had its interrupts disabled when driven by the user-mode driver, so as not to crash the unmodified naive kernel with unexpected interrupts. (The user-mode drivers used either busywait-polling or sleep-polling for synchronization.) Similarly, DMA operation was not possible.

   In developing a local-area network interface, it was necessary to utilize both of those features. A slight kernel modification was made to reserve a block of physical memory which the kernel would not use. User-mode library routines were provided that (1) allowed allocation of that memory area to DMA operations (the results of which were then examined with "/dev/mem" or "sysphys"), and (2) allowed run-time installation of minimal interrupt-service routines (using "pre-compiled" templates) which merely stored the device status in a mailbox and cleared the interrupt (the user-mode driver polled the mailbox, rather than the hardware).

   Again, the device driver was not "ported" to kernel mode until the hardware had been completely checked out, the device driver algorithms were debugged, and the sample application programs had demonstrated end-to-end functionality.

Several examples have been given of developing what is normally considered "kernel mode" software in user mode. While these examples are not likely to apply directly to other environments, it is hoped that implementors will be encouraged to consider the "user-mode style" when planning future kernel-mode software development projects.

;login:

The USENIX Association Newsletter

Volume 9 Number 6 **December 1984**

CONTENTS

The deadline for submissions for the next issue of *;login:* is February 15, 1985

USENET in the Sky

Satellite Distribution of Netnews: The Stargate Experiment

Lou Katz

Introduction

Several thousand computer sites in the United States, Canada, Europe and Australia are linked together into a logical "network" which permits the transfer of messages directly from one individual to another (mail) and the posting of messages to be read by anyone who is interested (news). The many sites on this network which are involved with news transfers collectively are called USENET. More specifically, USENET is defined as all sites receiving the newsgroup net.announce. A USENET link between two sites is one that net.announce is sent over. Note that this is different from a *uucp* link, over which mail and file transfers may occur but not necessarily news.

As more computer sites have gained access to this network a number of problems have arisen, in particular with respect to the communications costs incurred in the operation of this net and to difficulty of new sites obtaining access. As usage increases, USENET is faced with the spectre of increased costs possibly forcing curtailment of network activity, an eventuality which is causing great concern in the network community. Furthermore, the magnitude of the load which news places on a site is so large that new sites have great difficulty finding a site willing to feed news to them. Many new sites wishing to get such information are without connections.

At the present time it is estimated that there are about a thousand sites in the network, with that number growing daily! Total network traffic is basically proportional to the number of sites, so that traffic is growing too.

It is vital to realize that network services, to be useful, must connect to the machines a particular individual uses regularly and as a matter of course. For news and especially for mail, it doesn't work for the person to have to make an individual special call to a different machine just to see if there is mail or news for him/her, any more than it makes sense to walk two miles to the post office each day just to see if there is mail, when there often will be none.

Note however, that USENET IS NOT A NETWORK in the formal sense! That is, unlike all other nets (ARPANET, CSNET, BITNET, etc), there is NO administration, no central structure, no joining, and no membership to USENET. The net actually represents the human and professional network of personal, technical and business contacts, and PAIRWISE desires for groups or individuals to communicate and share information easily.

It is just this pairwise organization which gives the network its vitality. Without the burden of administration, all that is required is the telephone switched network, which permits any machine, anywhere to contact any other machine DIRECTLY, subject only to administrative and software agreement between its managers. Some pairs of sites are linked via dedicated high speed circuits, because of the volume of traffic between them. This linkage is not, however, crucial to the operation of USENET.

News forwarding often represents a massive percentage of the overall data traffic flowing through a given USENET site. Some sites have taken this responsibility upon themselves for a variety of reasons, but most sites will only receive news or forward it to very specific recipients. Mail is treated somewhat differently, and many sites will forward mail as a professional courtesy to others, which improves overall mail performance, and helps ensure that others will forward mail to them.

Estimates indicate that MAIL accounts for about 15% of the network "load" and NEWS for about 85%, although at high volume nodes or central sites which forward both news and mail, mail may reach 50%.

For two machines to be networked, they have to be connected in some manner. This connection can be a dedicated link (leased phone line, internal wires within a site, infrared relay, fiber optics) or a shared link such as a dialup line. Dedicated links, except for the trivial case of running a wire between two machines in the same room, almost always involve dealing with large external entities such as local phone company and common carriers to get special dedicated wire services. These links are expensive, are rapidly growing more expensive, and can involve very long (months or longer!) waiting periods for installation in many areas.

The cost of network phone calls is hard to see directly. However, if one conservatively estimates that there is about 1 Mb of news every DAY, and if this is transmitted at 1200 bits per second with a an error-correcting transmission protocol, there are roughly 3 hours of transmission per day. Current phone rates run about $.15-$.30/minute in the dead of the night (the times usually selected for transmission, just to keep the costs down) either interstate or intrastate. If a site getting news initiated the call itself, it would spend about $36/day or a little over $1000/month on such phone calls. Unfortunately, many phone calls wind up with bad circuits, giving numerous retries and aborted messages. This can add up to a factor of two on call costs. If two sites attempt to utilize a single phone line for both a feed in and a feed out they are likely to utilize the entire 11pm-8am nighttime rate slot on one line. Since there is a pyramiding effect with each site servicing several below it, a single site could easily dedicate two or three lines just to network service, and often wind up using more expensive evening and daytime connections.

If only one hundred sites have to make non-local calls for this purpose, the national phone bill attributed to network activity would be over $100,000/month! This amount may very well be much too conservative, as reliable rumors suggest that the phone bill for one particular site is significantly in excess of $20,000/month!

New technology is beginning to provide us with modems capable of working on the nationwide switched network at speeds of 2400 bps, or double the present common speed. The costs of these modems are much higher than the common 1200 bps hardware. Installation of such devices could cut some phone bills considerably (though by less than half, due to various technical factors), but only if both sides of a connection have them. It is unfortunately easier for many system administrators to justify rising phone bills than to receive approval for such a specific purchase, often from a different budget category, so that faster modems may not provide any relief. In fact, even if phone bills COULD be cut in half in this manner, the costs would still remain very high, and would still grow constantly worse as the network grows and news traffic continues to increase.

A Possible Solution

Lauren Weinstein has presented at the Summer 1984 USENIX Conference in Salt Lake City (Cf. Conference Proceedings, p. 18) a promising technological solution to the most pressing part of the problem, the cost of news transmission. The idea is as follows: portions of the video signal on TV transmission are not used for picture information, and can carry other information, in particular, suitably encoded ASCII. The effective bandwidth of this type of transmission could easily exceed 65 Kbps. It should be possible to establish a computer system at the "headend" of a cable or satellite transmission system, and upload such information piggyback on the TV signal. Any site which wished to receive the data would get a decoder and either a cable link or a satellite receiver dish.

The decoder would have sufficient internal memory to store a significant fraction of a day's news transmission (e.g. 500 Kbytes), so that the local computer could buffer and flow control the input and select and extract the information it wanted from the decoder at its own pace. Estimated costs for the decoder are about $1000 each (retail) and about $1300-$1500 for a satellite dish for most locations in the continental U.S. and parts of Canada, if the channel with the information were not also carried by a local cable TV company.

The economics of netnews would then change radically. No longer would a fanout of news have to occur over the dialup network. Rather, each item could be transmitted ONCE to the head end distribution computer, then "broadcast" for all to receive over the satellite system. The TOTAL "national" phone bill for news then decreases to about $1000/month, instead of several hundred thousand dollars.

The cost of the original transmission which occurs when an item is submitted (the phone call from the submitter's computer to the satellite link) is obviously borne by the submitter. The costs of the reception equipment and decoders are either one-time costs to the installation, easily amortized over a few months of phone bills, or else handled as monthly rental fees. This scheme does not, in any way, cut off the current mode of transmission of netnews. However, as more and more sites have to examine their phone budgets, they will generate both dollars and justification for inclusion of more and more newsgroups via satellite transmission.

The Experiment

Lauren Weinstein has secured the cooperation of several corporations and institutions in conducting an experiment into the technical feasibility of this mode of transmission.

The purpose of the experiment is to test the reception quality, error rates, flow control and system reliability and functionality. Reception will be tested both directly from a small reasonably priced microwave dish, and from ordinary cable-TV service in a number of locations.

The USENIX Association is providing support for incoming phone lines at the transmitter site, a small microwave receiver dish to test that mode of reception and travel to the transmission site to set up the system. The Association is also providing coordination of the efforts of Lauren Weinstein and the other participants, as well as dissemination of the results through written articles in *;login:* and, of course, over USENET, and a presentation at the January technical meeting in Dallas. If technical conditions permit, there will also be a live demonstration of the system at that meeting.

SSS (Southern Satellite Systems, Atlanta, Georgia) is supplying the experiment with continuous use of one scan line in their broadcast signal, with an effective baud rate of 1200 baud for a few months. They are also providing access to the uplink encoder which will properly format the input ASCII information and insert it into the TV signal. There transmissions are going out under the TV signal of WTBS, the Atlanta-based "Superstation", which is widely available throughout the United States. They are also providing two sets of tuners and decoders for receiving the signal directly and extracting the ASCII stream from the video.

Bell Communications Research (BCR) is providing modems for the uplink facility and other support.

Fortune Systems Corporation (Redwood City, California) has provided the uplink computer, a Fortune Systems XT30 UNIX system, which will receive netnews articles from dial-in phone lines and format them for insertion into the video signal.

If the experiment shows that we will achieve satisfactory performance from a technical point of view, the UNIX community at large will then be faced with the far more difficult problem: how to make this technology available so that USENET will flourish. The future organization of USENET is a more complex issue. For a stable network capable of functioning over the next few years, a host of legal, financial and organizational issues must be faced. How can even a modest effort be financed? What information or news groups could such a network distribute? Who would be responsible for content? These and other considerations must be worked through if satellite transmission is to become a viable facility.

;login:

The USENIX Association Newsletter

Volume 10, Number 1 **February 1985**

CONTENTS

The closing date for submissions for the next issue of *;login:* is May 1, 1985

A Bit About Eighth Edition UNIX

Harold Cross

Version Eight, Eighth Edition, V8; these names refer to the flavor of UNIX which is currently in use by the Computing Science Research Laboratory of AT&T Bell Laboratories. This article is meant to familiarize the reader with V8. It does not describe it in any great detail. For additional information see the references or send me mail (bellcore!hac).

V8 is based on 4.1BSD. Naturally 4.1 didn't spin around on 1127's disks for long before changes were being made. But it wasn't dubbed Eighth Edition until about two years ago.

> From research!dmr Mon May 30 01:45 EDT 1983
> :tcejbuS v8
>
> The Eighth Edition System is the line discipline stuff, plus PJW's
> 4K file system, plus his remote file system. I.e. we decided
> to give our state a name. Partly this was to disarm complaints
> that we were running 4BSD. Also, Doug is trying to arrange a new
> manual, so besides the considerable system changes there may be an
> actual printed 8th edition manual.

Streams

The line discipline stuff was first described publicly by **dmr** at the Winter 1981 USENIX meeting. Further coverage is found in [1]. Briefly, it is a mechanism providing a full duplex channel through which processes (user level and kernel) communicate. It is also known as a stacked line discipline. Processing modules can be pushed into (and popped) from the channel. Thus, for instance, the *init* program opens a terminal device and pushes a "tty" line discipline into a channel between it and the terminal. Likewise, when switching handlers from the "old" to "new" disciplines using the *stty* program, it first pops the old one from the channel and then pushes in the new one.

The various disciplines are kernel objects (functions). This provides an elegant (clean in design, implementation and use) mechanism that isolates many common character processing functions from device drivers in the kernel. The generality afforded is also exploited to do such things as hardware simulation or, as **rob** has done with the 5620 terminal, to place the terminal handler in another processor.

Fast File System

pjw's 4k file system is a fast file system which coexists with standard 4.1BSD-type file systems[†]. There are two aspects which make this implementation faster than the 4.1 file system (and probably as fast as the 4.2 file system).

The block size is 4K bytes. More interesting is the fact that the "free list" is described by a bitmap. The bitmap resides in core, allowing for quickly locating free blocks and even more quickly adding blocks to the free list. A side effect of this implementation (or perhaps its impetus) is that the search for a free block (given the previous used block) can efficiently locate one on an appropriate cylinder (if there's one available). The latter aspect is probably the most significant factor in overall increased throughput.

I mentioned that the 4k file system coexists with the older type. The file system structure contains a union of the two free list implementations and the appropriate I/O routines check the file system type. Although not as gross as the 4.2 file system, this implementation also trades off conceptual

[†] In fact the superblock structure is rearranged making it necessary to "fix" a 4.1 file system using *fsck*. But it's a simple matter to rearrange it so that this isn't necessary.

simplicity for efficiency.

Network File System

pjw's conceptual pendulum swings the other way with regard to the remote file system. Here entirely new capabilities are added in a straightforward manner. The remote file system uses a hierarchical syntax where remote machines' file systems are mounted on the local file system (the convention is /n/machine/...). It's implemented in the kernel on the local machine and at the user level on the remote. The implementation is transparent to the users' programs. Locally, there is a file system switch (a la *cdevsw*) that causes the appropriate routines to be invoked on the different types of files (local, network, processes (see later), and faces[‡]). Routines that access networked file systems do so by invoking a server on the remote machine.

One of the nicest aspects of this system is its generality. A remote file system is mounted by telling the system the local mount point and giving it a stream connected to the remote file server. Thus the network file system can theoretically run over any communications path (a modem, a tty line, Ethernet, PCL, etc.). Since the server is a program utilizing nothing more system-dependent than *select* (and an understanding files), it can run under any version of UNIX. This means I can share anyone else's file systems but not vise versa.

Another new type of file is implemented in the concept of processes as files [2]. Here the directory */proc* contains files that represent running processes. The standard file access routines in this case interact with the address space of the said processes. This is a nifty way to manipulate them. (By the way, there are 128 file descriptors in the Eighth Edition.)

V8 is more than the key kernel changes described above. Next installment I'll describe some of the wonderful utilities and applications available under V8. But the system is merely a reflection of its designers, contributors and maintainers, most of whom just seem to possess extraordinarily good taste. Remember the Seventh Edition?

References

[1] "A Stream Input-Output System," Dennis Ritchie. B.L.T.J. 63:8, October 1984, pp. 1897-1910.

[2] "Processes as Files," T. J. Killian. Proceedings of the Summer 1984 USENIX Conference, Salt Lake City, Utah.

[‡] When a process opens a file of this type, the appropriate representation of someone's face is retrieved. More on this next time.

The Winter 85 USENIX Conference — First Order Observations

Kip Bore

The 1985 Winter USENIX convention was held January 23-25 in Dallas, Texas. The Fairmont Hotel provided an elegant setting, in keeping with the USENIX tradition of outclassing the attendees. At first we feared we had arrived at the wrong hotel — there was no line of eager wizards waiting to check in, no loiterers in the lobby sharing the finer points of sendmail configuration files. We soon found our way onto a bus (UniForum Route #1) and headed for the AT&T bash at the Anatole, by way of the famed InfoMart (housing the vendor exhibits) and UniForum Bus Route #2. Here we found a few familiar faces, many of whom had new business cards to trade.

We hope that the summer 1984 meeting has established an enduring precedent, as copies of the *Proceedings* were again available at the conference. We'd like to review some highlights that we believe won't otherwise be found in print. The talks got off to a promising start, with an entertaining keynote address by Rob Kolstad ("Whither the Gurus"). This topic seemed especially timely as we searched the audience for seasoned veterans of USENIX meetings, and found them lacking. Lauren brought us up to date on the satellite netnews experiment, complete with slides of rural Georgia (home of WTBS) and a live demonstration. Hotel personnel were puzzled at the throngs gathered around a television set that had been adjusted to split its picture by a snowy horizontal line. (For this, they rented a satellite dish?) Nearby, net.physics scrolled slowly by on a VT100. Susan Nycum gave a lucid presentation on the legal issues that cloud the project. The audience was invited to raise non-technical questions at the open board meeting[‡], effectively staving off the expected controversy.

We learned that the 4.2 BSD XNS tools developed at the University of Maryland were first tested a few days before the conference (and they do work). Ian Darwin enthralled the audience with his talk (it's hard to describe the sound of hundreds gasping ". . . oh no, not *init*"). We found it curious that a speaker had prepared a set of hand-written viewgraphs for his presentation on *troff*. In the final session of the meeting, Peter Honeyman cast a spell of confusion on the dwindling audience when he incanted the words "directed graph." We were amused by the results of Mark Horton's query: "How many of you believe that the present form of mail routing (*i.e., machine!user*) is satisfactory?" (A few hands went up.) "How many of you believe that we need something new (*i.e.,* domains)?" (A few hundred hands went up.) Given these results, we were troubled that most of the addresses we found in the attendee roster were of the form "*machine!user.*" In fact, a rough count showed that these outnumbered domain-style addresses seven to one. From these data we conclude that "domainists" tend to register on-site, and "bangists" prefer to catch early flights home.

One afternoon we ventured to explore the InfoMart (voluntarily), home to vendor exhibits on a daunting scale. Cleverly, we were issued an embossed plastic badge, and each vendor was issued lots of carbon-layered forms and a credit-card machine. Whoosh! With one snap of the wrist, we were assured of finding ourselves on dozens of new mailing lists. We limited our attendance to a handful of exhibits (hot new machines and a few vendors from whom we really *needed* information). On the positive side, we detected a technical presence at several of the exhibits we sampled. We approached one glowing console, and observed 23 lines of failed login attempts (*e.g.,* "guest"). We typed "root" and were promptly rewarded with '#'. Over the years, we have grown weary of "cp /bin/sh /dev/kmem," so we simply cleared the screen and typed '^D'. It was not an outstanding show for collectors of UNIX memorabilia. Although attendance was rumored below expectations, DEC's supply of UNIX licenses was exhausted early. We saw no jugglers or larger-than-life inflatable frogs, but we did notice one "Delilah" in gold lame.

[†] Other articles in this issue have more information on future meetings.

[‡] Covered in the article on the Open Board Meeting elsewhere in this issue.

The social hit of the meeting was a group outing to "Photon," where players donned helmets, battery packs, and LED-studded gear, the better to dart around in the dark zapping one another. The lobby bar was more empty than not, and we concluded that the hospitality suites (and excitement) must abound at the UniForum hotels. This theory died hard when a UNIX luminary appeared late one night, seeking a room at the Fairmont. His UniForum hotel was dead, and he was emigrating to be "where the action was." Blue ribbons adorned more than the usual number of participants, with "Listeners" outnumbering "Speakers," and in turn being surpassed by "Sleepers." We also noticed an occasional "Bored Member" and the prized "Best of Breed." In all, we spotted only one Bill Joy and nary a Rob Pike badge.

The open USENIX Board Meeting was notable for its lack of controversy. The Stargate project was discussed at great length, but we thought nothing new was said and it all came to no particular end. The separation from UniForum was viewed mostly as a good thing; exception taken by those individuals who can attend only one conference per year. The co-occurrence of USENIX and UniForum (in time and space) is not likely to happen in the future, and that presents a dilemma for some who must choose. The "how-many-meetings-should-we-have-each-year" issue was raised again. The answer is still "two," with emphasis on a broad, long, technical conference in the summer and a specialized, short, workshop-oriented winter meeting.

Before long it was time to board our return flight, where we reflected on Dallas in January and began looking forward to Portland in June.

The Winter 85 USENIX Conference — E.U.U.G's Report

Dallas, Texas, 23rd — 25th January

— An Informal Report —

Dominic Dunlop
Sphinx Limited

Dallas, Texas. Where everything is bigger, including UNIX conferences. If you think 2,000 delegates for USENIX is a lot, what about the 20,000 at UniForum, held in the same city at the same time? The division couldn't have worked out better: UniForum attracted all the marketing presentations and put on the biggest trade-show yet, but left USENIX with a solid diet of technical material for UNIX aficionados without three-piece suits.

Dress code for gurus (wear jeans with no more than three holes, but don't bother to buy a tie) was featured in the keynote presentation, "Whither the Gurus?" by Rob Kolstad of Convex Computer Corp. Explaining that gurus are like cabbage-patch dolls — inordinately expensive, hard to find, and all different (though not, on the whole, cuddly) — Rob described how to build a guru trap. You bait it with lots of money, lots of fast hardware (Convex makes super-computers, so no problem there), and a stock-option. And, to keep a guru once you've got one, make sure that your vending machines are restocked with junk food twice a day. Well, this is America.

But how does a mere programmer get to be such a sought-after commodity, and not a mere initiate, wizard or lama (lama?)? Training, that's how. You need to learn left-handed touch-typing (so you can hold your coffee cup in your right hand). You need to learn about sixth-generation computers (AI will *really* work this time around). And you must be aware of the software Peter Principle: any program will ultimately rise to its level of incompetence. Rob's software University offers all these skills and more, and is open to any student with money...

The first paper, by Lauren Weinstein of Vortex Technology, hardly brought us down to earth: it was a discussion of transmitting netnews by satellite. At a previous meeting, Lauren had rashly volunteered to get a demonstration system going, sneaking netnews into unused lines on satellite TV signals (Hmm, how about it, Sky Channel?). Finding a TV company which understood the concept, but wasn't already selling all the available time to Dow Jones for millions of dollars, proved difficult, but WTBS in Atlanta came to the rescue. The initial hook-up followed netmail tradition by beaming a signal 47,000 miles to get it 8 miles from a network gateway to Lauren's experimental receiver. The demonstration at the conference showed a somewhat longer hop, and those of us with badly adjusted TVs in our hotel rooms could actually watch netnews at the top of the screen!

Ninety minutes into the conference and no word yet from a lawyer. Too good to last? Yes. But, surprise, surprise, Susan Nycum hadn't come to tell us about UNIX licensing. She gave an interesting analysis of five good ways of getting sued over the content of your news item. Using a broadcast medium like a satellite might mean the carrier could get sued too. What makes you think the US was founded by lawyers? By the way, you won't find the paper in the proceedings: Susan said it would be impossible to write any sort of legal statement on just a few sheets. Besides, who would put on the ribbons and seals?

Then coffee and time to skip off. A sporadic but free exhibition of American buses of the past fifteen years operated between the conference hotel and Crystal Palace. Sorry, Infomart, a remarkable building modelled after that of the great exhibition in London 120 years ago. UniForum was the first show to be staged there and,. in tribute to the hardware and software on show, construction was almost finished. I dallied several hours among the biggest collection of UNIX hardware and software ever assembled under one roof, and chuckled to myself about the number of thrusting market analysis handouts which AT&T's "kiss and make-up" session with Microsoft, announced two days before, had invalidated.

By the time I'd got back to USENIX, I'd missed all the kernel implementation papers, arriving in the middle of a religious service dedicated to "Modula-2: An Alternative to C for Systems Programming" by Morris Djavaheri and Stan Osborne of San Francisco State University. In America, the state and religion are constitutionally separated so it was only fair that we should next hear about "A UNIX-based Ada Run-time System" from M. D. Scheer and S. Rajeev of AT&T Bell Labs (whose trademark Ada is not). And, answering a similar need by adding new primitives to the only language we can currently rely on, Gehani and Roome (also of Bell Labs) gave an overview of Concurrent C. An unkind suggestion that this exemplified the software Peter Principle at work was adequately refuted in the ensuing panel discussion.

Another break, this time for soft drinks devoid of any unfashionable substance which might make them palatable. The day's final session dealt with performance measurement. Bill Meyer's graphic alternative to *ps* and its relatives looks interesting, and may yet pop-up in net.sources. John Saxter discussed "Interpreting UNIX Benchmarks" in a rather lightweight manner. More interesting was a Birds of a Feather session by Gene Dronek, author of the AIM Benchmarks, a commercial suite. Gene is working on defining a "standard VAX" (750 and 780, System V and 4.2BSD) so that we can know what all these comparisons so beloved by advertisers actually mean (he's an optimist). If you can help, contact Gene. He also has a program which will degrade your disk performance by 3% for each minute it runs... .

During what was left of the evening, we had time to discover two more things that are bigger in Texas: the lack of downtown activity after dark, and the distance to an open restaurant or bar. The USENIX city guide showed where succor could be found.

Friday got straight down to business at 8:30am (the room was $80 a night, couldn't I lie in?) with a well-presented paper from the University of Maryland. A gift of 30 Xerox workstations (why don't I get presents like that?) prompted them to discover that Berkeley's much-vaunted generalised networking kernel isn't really. Making it support Xerox Network Systems protocols as well as TCP/IP across an Ethernet turned out to be quite a job. The code is available free if you're a University Grants Program member. Forget it. You're not.

XNS was a mere hack beside "The Lincs Communications Architecture" described by Joseph Requa of Lawrence Livermore Labs. If you feel like re-writing your kernel from scratch, read his paper. At a much more populist level, Judi Uttal of Locus Computer Corp told us about a Transparent Integration of UNIX and MS-DOS. Neat. Your PC just treats the UNIX system (or one of a choice of networked UNIX systems) as another drive. Finally, somebody from Sun (the paper has nine names on it, and I lost track) gave us an Overview of the Sun Network File System. Not as ambitious as (say) the Newcastle Connection, NFS allows transparent operations on remote disk files (but not device files). There is no remote execution. A display of Ethernet-connected Sun, Pyramid and Gould machines working with NFS at the UniForum show testified to the system's practicality.

After coffee time, comedy time. Ian Darwin and Geoff Collier's paper, titled, among other things, "Real Programs Dump Core", started off by stating that bugs always happen to the other guy. Which would be fine if there were a kernel call to tell you whether you're running in *other guy* mode. Then followed a series of horror stories about some real programs, some subtly changed to protect the innocent (well, AT&T's lawyers would plead that way anyway). More amiable flack for Ma Bell (deceased) came from Motorola's Alan Filipski, describing some fun things they'd found when porting System V to the 68000.

I missed out on the Software Tools and Applications papers, although "Development of a Compiler for the Bourne Shell" by Vincent Kasten and Paul Ruel makes good reading for anybody considering compiling any language designed for interpretation, without a specification, and with lots of weird special cases (example:

```
case i in
        esac|a)
```

is illegal, but

```
case i in
        a|esac)
```

is fine).

A discussion of mail closed the conference, perhaps to remind everybody to keep in touch. Mark Horton et al of AT&T Bell Labs et cetera are struggling valiantly to approximate reality as closely as possible with network maps and (600 Kbyte) databases. Peter Honeyman of Princeton discussed how to parse *seismo!cak%purdue@csnet-relay*, and similar valid but problematic addresses on the various US networks (you knew USENET is not mathematically a network didn't you?). The next speaker, Mike O'Brien of BBN, pointed out that his experience was that some mail bears addresses which owe much more to invention than to logic, and described an inverted index system for generating addresses from names. That USENET is an anarchic tangle was shown by Mark Horton's straw poll: almost all those present voted that a system of domains should be put in place, so that in the absence of an efficient address which is known to work, something like *user!site@europe* would be guaranteed to work. Public domain software to supervise this under 4.2BSD and System V should hit the streets soon. Don't worry – your favourite mile-long '!' addresses will still be supported.

And so it ended, leaving the hotel empty but for the few of us who had elected to leave all of Saturday to get through the labyrinth of the Dallas-Fort Worth airport, apparently a projection into three-space of a perverse higher-dimensional object.

Thanks are due to Charisse Castagnoli of Teknetron Infoswitch for pulling together the programme in record time, and to Rob Kolstad for burning midnight-oil – and the ears of several of the speakers – in order to get the proceedings published before the conference[†].

USENIX Conference Proceedings Available

Proceedings for the following USENIX-sponsored conferences and workshops are available from the organizations listed. Prices and overseas postage charges are per copy. California residents please add applicable sales tax. Payments must be enclosed with the order and must be in US dollars payable on a US bank.

Meeting	Location	Date	Price	Overseas Postage	Source
USENIX	Dallas	Winter '85	$20	$15	USENIX
USENIX	Salt Lake City	Summer '84	$25	$15	USENIX
UniForum	Washington DC	Winter '84	$30	$20	/usr/group
USENIX	Toronto	Summer '83	$30	$15	USENIX
UNICOM	San Diego	Winter '83	$25	$15	STUG

Addresses

USENIX Association	Software Tools Users Group	/usr/group
P.O. Box 7	140 Center Street	4655 Old Ironsides Dr., #200
El Cerrito, CA 94530	El Segundo, CA 90245	Santa Clara, CA 95050

Problems with Dallas Proceedings

If you discover errors such as missing or inverted pages or other problems with your Dallas Proceedings, please return them to the USENIX office and you will receive a replacement at no cost.

Past USENIX Distribution Tapes Available

The USENIX Association Board of Directors recently voted to lower the cost of past distribution tapes from $100 to $75, due to the availability of the VAX 11/730 in our office for copying. Any current Institutional Member may purchase previous distribution tapes for which he has the appropriate licenses by completing a tape release agreement and sending $75 for each tape desired (no purchase orders, please). Only 1600 bpi tapes can be written. All tapes from 1981 and later are in *tar* format. The 1980 tape is *tp* format. For more information, contact the USENIX Office.

A list of distribution tapes currently available follows. Some descriptions are sketchy since little or no information was provided with the submissions. (The 1980 tape list is extremely sketchy since this tape was produced before the USENIX Office was set up and there are no records as to the contents. The listing below was produced from an inventory of the files on the tape.) Remember also that submissions are distributed as received: some may be incomplete or no longer relevant.

1984.1 Tape Contents

Submission	Submittor	License Requirements
Changes to System V kernel, commands, and include files	John Buck Polytechnic Institute of New York	Sys V
New utilities		
Modified version of *make*(1) that understands RCS	Charles LaBrec Purdue University	V7
Enhancements for 4.2BSD Arpanet code	Bill Shannon Sun Microsystems	None

1983.2 Tape Contents

Submission	Submittor	License Requirements
Kernel modification for higher performance raw mode tty input	Wisconsin State Laboratory of Hygiene	PWB, V6
RJE system for UNIX to Univac 1100		PWB, V6
Enhanced spooling system		PWB, V6
Vir: input record entry/retrieval system		PWB, V6
Local mods to many standard UNIX commands		PWB, V6
UTMOST menu-drive office system	Perkin Elmer	None
Zork game	Daniel Strick University of Pittsburgh	32V
Command line argument handling and date handling packages	Solar Physics Group Stanford University	None

1983.1 Tape Contents

Submission	Submittor	License Requirements
LOGO implementation Version 3	Brian Harvey Lincoln-Sudbury Regional High School and Atari	None
A UNIX system performance benchmark suite and related manual pages	Martin Tuori Defense & Civil Institute of Environmental Medicine	None

V7 drivers: Dicomed COM device via DR-11B, modified TM tape driver, and Xylogics disk controller		V7
Bootstrap code for Xylogics controller		V7
Reference information program for scientists	Stephen D. Klyce LSU Eye Center	None
Almost debugged BASIC to C translator		None
Code changes to *ed*, *sort*, etc.		(?)
Primitive graphics routines		None
Miscellaneous routines		None
Routine statistical programs		None
MENUNIX	Gary Perlman University of California at San Diego and Bell Laboratories	None
Data analysis programs		None
Libarg—an argument line cracker	John Quarterman	V7
Line printer spooler	Douglas Gwyn	None
Random utilities	Yoran Shoham Geotronics Corp.	None
Screen editor based on *ed*	J. D. Wise Rice University	V7
Tools for extracting cost information from files and including them in proposals	Geoffrey Kodosky National Instruments	None
Restricted UNIX environment for stand alone utilities and diagnostics		V6
System III *uucp* with "all known bug fixes"	Steven McGeady Tektronics	System III

1982 Tape Contents

Submission	Submittor	License Requirements
Device driver and library for Genisco GCT300 color graphics system on a VAX (source license required for kernel to install driver)	General Instrument Corp.	None
Set of commands to implement functions		None
Line printer programs based on 4.1BSD, including graphics support for the LSY-11 (Printronix 300)		None
Seventh Edition commands for 6th Edition Systems	Geotronics Corp.	None
Terminal ports on and off		None

Interactive programmable form filler		None
Tools for multiplexed binary data files		None
General purpose output spooling system		None
A "lock" call and real time support for 6th Edition kernel		None
Graphics driver for H.I. CPS-15/6 plotter		None
Implementation of LOGO language interpreter	Lincoln-Sudbury Regional High School	None
Creation and typing of form letters		None
Modifications to V7 system programs		V7
Improvements to *nroff*		None
EMACS-like editor called TORES		None
Quite a few games		None

1981 Tape Contents

Submission	Submitor	License Requirements
Various programs	Walter D. Lazear U.S. Air Force	None
TUG	Dennis L. Mumaugh	None
UNIX course	Dept. of Defense	None
coregraph		None
src/GC—programs and library routines for public consumption	Darrell R. Word Geotronics Corp.	None
hlp—primitive help system		None
UNIX utility sources upgraded to V7		V6
UNIX kernel and bootstrap sources plus drivers for CR-11, DZ-11 and XY-11		V6
tig—version of Mike Muuss's terminal independent graphics system		None
Versatec utilities	Michael D. O'Dell Lawrence Berkeley Lab	Phototypsetter
make enhancement for maintaining more complex systems	Robert L. Walton MIT Lincoln Lab	V7
pr—enhancements to make a better file print formatter, used in the lpf shell file		V7
col—ditto for printing manuals sections, modified *nroff* to accept whole shell		V7

cpp—would like to have more advanced *cpp*; included to stimulate interest		V7
a68	Chris J. Terman MIT Computer Science Lab	V7
Device handlers: dx—modification of handler rx—single density floppy handler from scratch xy—incremental plotter interface hm—handler for System Industries Fujitsu 160 Mbyte Winchester disk with 9400 controller mdec—boot loaders for hm and rx cscan—recognizes usual char constant syntax ls—lists directory into specified buffer and sorts match—pattern matches *ed/grep* regular expressions sl echo—modified to use cscan ls—modified to list directory recursively, depth 1st chaos—redirect output to someone else's ADM-3A to cause interesting reactions rtpip—to create, squeeze. etc. an RT file system in a UNIX (special) file f.r—to run RT Fortran, Basic, Macro, link	Geoffery Kodosky National Instruments	V6
nbs/dd m7	Joan S. Bowden National Bureau of Standards	None
dungeon rt11 (?)	Daniel R. Strick University of Pittsburgh	None
PDP11 (?)		None and V6
VAX11 (?)		None and 32V
bin/df—a V7 shell file that inhales the mount table and prints a df for each filesystem and pathname bin/help—shell files similar to *man* and fires up *nroff* to print it src/acct.c—print accounting file src/cookie.c—random print of fortune cookie-type message src/cpm—allows a UNIX system to read and write CP/M format floppy disks; requires a RX-11 driver src/tprintf.c—terminal independent *printf* for optimization	Scott Bertilson Rosemount Inc.	None

src/logger.c—coughs message for log on or off		
src/sim—discrete event simulation package		
src/splot.c—simple screen plotting program		
src/talk.c—like *write*(1), but 1 character at a time		
src/words.c—attempts to find words that sound like thoses entered		
src/xtime.c—tests current time for shell files		
sys/dev/rx.c—RX01 floppy driver, table driven interleaving		
sys/sys/clock.c.diff—small mod to clock.c gives a blinky-light histogram of CPU time usage for a PDP 11/45 or 11/70		
c-fix	George K. Rosenberg	None
compat—PDP-11 compatibility mode for VAX 11/780 modified to support MACRO on V6	Joshua K. Knight, III Stanford University	V6
contents		32V
macro		V6
lpd—line printer spooling system and replacement for V7 *cu* command	Samuel J. Leffler Sytek Inc. Bill Shannon Digital Equipment Corp.	None
man (?)		
tip (?)		
able.ms	Laurence J. Morandi Tektronix	V7
bio.c		
machdep.c		
mch.s		
param.h		
rl.c		
seg.h		
tm.c		
P—prints files on the terminal one full screen at a time	David R. Galloway University of Toronto	None
c(1)—splits a long output up into columns and prints it side by side		
cires_apoc	Ernest W. Harkins	None
crash.doc	University of Colorado	None
leroy		
leroy.ms		
libCf and libCm		
libDV and lib HP		
libplot		
man3		
plot		
unix_at_cires		

1980 Tape Contents

Submission	Submittor	License Requirements
menu	adi	?
tcom		?
concat	New York Blood	?
convert		?
equ		?
news		?
sys1.c		V6-7 (?)
sys2.c		V6-7 (?)
sys3.c		V6-7 (?)
sysent.c		V6-7 (?)
trap.h		V6-7 (?)
dz	Nijmegen	None
fconv		
restor		
rk		
sda		
apl	Purdue	V6
sys		
changes to library routines	Pitt [sic]	V7
assorted stuff	UK	V6
stuff	Boulder	
cf	Ampex	
man/tty.4		V7
stdplt		None
sys		V7
?	Caltech	V6
tools	DPW	None
stuff	Delaware	
campos		
cincinnati		
cwru		V6
cwru.v7/sys		V7
?	Geotronics Corp.	None

wetzel	Pittsburgh	
rosenberg	Pittsburgh	
?	Tektronix	None/V7

A Word Puzzle

Rich Kensicki

Find 16 C Language Keywords

```
N Q S S O D Y S R J D B W T K K
L Q R P X K O E W D M R H A V E
W X O N B K T U S I E Y I L J J
H S I T A U P Z B O T T L Y X S
H R A V R V I N T L T C E T T S
M T S N T J X M A M E C H A R P
O B J H T L U A F E D S R S R A
O G I R O K N Q C P C Z U W N C
N N B T X R L K A N Y R T N E T
U U P G U E T H S Y C I T A T S
T P N V M S I Z E O F X T E B Y
C Z N I Y S J G T N Y A F X S W
U Z V F O H R N J R O V L T I A
R L I R B N B O K L H L B N J E
T L S Z W B L L F W N X E L B K
S X B Q B H K Q D O H Q D D P R
```

Netnews

I have reproduced below some of my network mail and a few "netnews" articles that I thought may be of interest to Australian UNIX users. I have deleted some of the less meaningful data generated by various mailers and news programs. No responsibility is taken for the accuracy (or lack thereof) of anything below.

From: kre@munnari.OZ (Robert Elz)
Date: 16 Apr 85 16:05:15 GMT
Newsgroups: aus.general
Subject: How to tell people (overseas) your mail address
Organization: Comp Sci, Melbourne Uni, Australia

I have seen far too many poor, & sometimes simply wrong, attempts at indicating an Australian (electronic) mail address to people overseas, that I thought I should try and explain what it is, and why it is nothing like what you might guess...

First - here is your (international) address

```
        mulga!HOST.oz!USER
```

You substitute whatever is appropriate for "HOST" and "USER". HOST can contain sub-domains if that is appropriate for your site.

Notes: The ".oz" is (generally) essential. It is mandatory if your host name contains sub-domains. (Note: even if for some reason your address (to SunIII) doesn't really have ".oz" in it, the ".oz" here is still needed. It is used to signal our mail system that this is a SunIII (ACSnet) mail address, and not uucp. It will be stripped frm the "HOST" part, and whatever is left will be used as the SunIII host address.) It should *always* be used.

There are NO '@' signs, ':' characters, or anything else odd like that. The address looks just like any old ordinary in it. All outgoing mail has the "From" address looking just like this.

Now, there are a few embellisments that you can make to that for various purposes. Eg: you can indicate the path to mulga ..

```
        {decvax,vax135,eagle,pesnta}!mulga!HOST.oz!USER
```

This is usually a good idea, as while there are quite a lot of mail systems around these days that can route to mulga, (or anywhere) some people still have to do it by hand, and mulga isn't exactly on the top ten list of famous hosts.

Next: you can specify an address for people on other networks who may not want to even think about UUCP addresses and routing etc.

The most common is one for people on ARPANET

```
        decvax!mulga!HOST.oz!USER@ucb-vax.arpa
```

("Berkeley" is equivalent to "ucb-vax"). That works for CSNET people as well, as their mailers know how to get to ARPA addresses.

Addresses can be invented for other networks as well, but at the minute I'm not prepared to make any rash statements.
(I don't know if I could manage to get any of them right!)
If and when you get to specify an address that will come across the X.25 links, the same principles hold, just some of the names change.

Now for the why... Mail addressing on various networks developed in a most haphazard fashion. Every network seems to have invented its own syntax, and few of them are really alike. Its for certain that they all have their own particular rules, and foibles.

To manage to survive in this mess, what we have to do is simplify everything to the extreme. Since (in the US) our network looks to be an offshoot of the US uucp network, we do that by pretending that that is exactly what we are (with mulga happening to have links to every host available). Thus, we make our address look just like a uucp address.

That also has the advantage, that few (and perhaps no) networks mangle uucp addresses so badly that there is any real problem.

Addresses that contain ':' or '@' though aren't in that happy position. They are the two most common characters for other networks to mangle. Attempting to specify an address using one of those is doomed ...

Note: mulga will accept a whole variety of wierd address syntaxes, if there's any reasonable way we can parse an address so that it looks to be an Aust mail adress, we will do that, and attempt to deliver the mail. That means that all the old addressing forms that you have been telling people still work (and will continue to).

But this is (really) also one of the sources of the problem. Mulga isn't the only host to do this, many do. With lots of hosts out there all grabbing onto every address as it flies past, and trying hard to interpret as an address that means something special in their particular world, if you try anything at all fancy, one of those mail programs is going to bite on your mail. Goodbye mail.

Now, just like mulga, all of those hosts have valid reasons for doing this, often steeped in eons of history, and it is essentially impossible for any of them to stop what they are doing - having tasted that choice mail morsel, they become adicted.

So, please, try and be careful when telling people what your address is. One day, there might be a simple, universal, form that works everywhere. One day there might be unicorns.

Finally, if in doubt, it is probably better not to specify a return address in your mail at all. That's what the mail header is for. It will usually end up being correct (more often than you might think), its also the address that applies at the point that your message was received. Also, if you specify an address, and it differs at all from the one in the header, people will often reply to both (those that don't most probably just use the header) - if they are both valid, then you end up getting the mail twice, which is annoying, and it gets sent here twice, increasing costs for no benefit.

Robert Elz kre@munnari

From: kre@munnari.OZ (Robert Elz)
Date: Tue, 15-Jan-85 04:15:19 AESST
Newsgroups: aus.general
Subject: Automatic UUCP path insertion at mulga
Organization: Comp Sci, Melbourne Uni, Australia

Mulga will now generate uucp paths to most US and European UUCP sites (plus the few in Korea and 1 in Japan).

You need do nothing, it will happen automatically to all uucp mail. Mulga's uucp mailer will insert a path to the first uucp host in the uucp path that you gave in the mail address you posted to.

If that host happens to be "decvax" or "vax135" then things are pretty simple, there is a direct link, and mulga will just use that.

That means, that any mail that worked before will still work now, just the same way. [You have been required to start uucp paths with one of "decvax" or "vax135" up until now, or your mail would have been returned from mulga ("host unknown").]

However, if you like, you can now omit some or all of the initial path to the final uucp host, mulga will fill it in as best it can. Note: its not possible to omit hosts after the first that you give, from there on the path must be complete (unless you happen to know that another host out there will do similar friendly things to your mail).

Nb: the syntax of the uucp path, either "xxx@host.uucp" or "host!xxx" is irrelevant for these purposes, each is considered to be a request for uucp to "host" and then to "xxx" whatever that might be (more path, or simply a user name).

If no route is known (eg: a new host that isn't in the database yet, or you misspell a host name) then you will get your mail returned, as before. If it was addressed to multiple destinations, then those destinations that were known will be sent - the mail returned should make it clear which ones were rejected (look for "no route known to <xxx>" lines).

The data used in the database is based upon the recent uucp maps, with just a little bit of local knowledge thrown in. (Like, I "know" that "piers" isn't a host connected to basser...)

So, its entirely possible that some of the data might be incorrect, sub-optimal, etc.

If you find evidence of any such cases, please let me know.

Especially, if you get mail returned from somewhere in the US as having been routed incorrectly, and the error is in the part of the path calculated by mulga, then be certain to send me the headers of the returned mail (including the headers of the original mail that was returned).

Note: incoming mail will continue to show the entire path, automatically generated replies that use that path should work without problems. There is certainly no need to deviate in any way from your current US mail practices (except possibly by sending a little less...)

Finally, to satisfy your curiosity, if you send mail to "uucppath@mulga", with the body of the mail containing a list of uucp hosts (and hopefully nothing else), then you should get mail back containing the routes that would be used to get to those hosts.

The "list of uucp hosts" is something like

```
        decvax, cbosgd  ucbvax,
        vortex mcvax!ukc
        sequent!ianj
```

That is, space or comma, or newline separated "words", where each "word" is either the name of a uucp host, or a uucp style path.

In the latter case, the first component of the path will be used (the rest ignored). Note, here the form "xxx@host.uucp" won't work, even though that is fine as a regular address.

Robert Elz   kre@munnari

ps: this applies to all mail arriving at mulga after midnight, Tues 15 Jan. (which time has already past!)

From: avolio@grendel.UUCP (Frederick M. Avolio)
Date: Thu, 10-Jan-85 17:13:24 AESST
Newsgroups: net.announce
Subject: ULTRIX(tm) APPLICATION CENTER OPENS
Organization: Bell Labs, Columbus

DEC Washington, DC Software Services announces the formation of an ULTRIX Applications Center located at their Landover, MD facility. The primary functions of the ULTRIX AC will be to provide ULTRIX software services to customers and to DEC employees and to be a focal point for such services. These services may include -- but are not limited to -- the following:

- offering support for employees and customers seeking information about ULTRIX products and software.
- developing new ULTRIX applications software and supporting good third-party software,
- writing and modifying device drivers,
- providing ULTRIX education services,
- support for the ULTRIX Engineering Group in enhancing the ULTRIX system

Most of this work will be done out of Digital's Landover, Maryland offices, but support will be provided wherever requested.

Kevin M. Lewis, Manager
ULTRIX Applications Center

ULTRIX and DEC are trademarks of Digital Equipment Corporation

--
Fred Avolio
301/731-4100 x4227
UUCP: {seismo,decvax}!grendel!avolio
ARPA: grendel!avolio@seismo.ARPA

[Moderators note: This can probably be considered advertising, but I am posting it because I think it's really of interest to almost the entire Usenet community. - MRH]

From: mark@cbosgd.UUCP (Mark Horton)
Date: Fri, 18-Jan-85 17:16:36 AESST
Newsgroups: net.mail
Subject: UUCP Subdomain Requirements
Organization: Bell Labs, Columbus

UUCP Subdomain Requirements

Mark R. Horton

Bell Laboratories
Columbus, Ohio 43213

Karen Summers-Horton

Usenix UUCP Project
2843 Valcour Ct.; Reynoldsburg, Ohio 43068

ABSTRACT

This document outlines the structure of the UUCP domain, and specifies the requirements for subdomains of UUCP.

1. INTRODUCTION

We are primarily trying to establish some notion of universal service in setting up domains. The current bang code was designed for an environment where every machine has a direct connection to every other machine, phone calls are free or cheap, and local area networks were just around the corner to replace this dialup kind of network. The UUCP network has grown into a huge network, and none of these assumptions are valid anymore. Hence, we are conforming to the only widely used, documented mail standard that has caught on in the electronic mail community, the ARPA domain standard. (The only other documented standard is X.400, which has not yet caught on well enough for us to consider. The old UUCP ! format is undocumented and unmanageable in a network the size of UUCP; we expect this format to continue to work indefinitely, but have chosen to support the ARPA standards at the user level.)

The UUCP community has evolved as an anarchistic, loosely connected network with no central administration and no rules. An anarchy works if it is small, but with thousands of machines already on the network and the number of UNIX machines growing at breakneck speeds, it has already begun to break down. We are establishing a central administration to keep track of who is on the network, and who to contact in case something goes wrong. Nonetheless, we are keeping the established rules to a minimum, to retain the cooperative spirit of the UUCP world.

We have determined that a flat name space is beyond our capability to administer, so we are dividing the UUCP world into subdomains. The current flat name space using the user@host.UUCP syntax will not be supported after a certain date [possibly December 1985] and all hosts will be expected to band together into subdomains. We intend to register UUCP as a top level domain. Direct subdomains of UUCP will therefore be 2nd level domains. (A subdomain is a domain which is beneath another particular domain in the tree. For example, ATT.UUCP is a 2nd level domain which is a subdomain of the first level domain UUCP, CB.ATT.UUCP is a 3rd level domain which is a subdomain of ATT.UUCP, and of UUCP.) 4th level, 5th level, and so on are also possible, but there will be different (presumably less restrictive) requirements for lower level subdomains. We want to keep the number of 2nd level domains manageable, since a complete list of 2nd level domains will be frequently published. We expect a hundred or so 2nd level domains to be a small enough number to be manageable and to allow frequent publishing of the list. These requirements are intended to keep the eventual number of 2nd level domains at around 100.

Rather than having us arbitrarily divide the world into fixed subdomains, we have decided to encourage the world to divide itself up. Any group of machine administrators can join together to become a 2nd level domain, provided the domain meets the requirements stated herein. Groups can decide for themselves the basis for subdivision, although geographic regions are an obvious choice. For example, New England and Northern California would be two obvious choices for 2nd level domains. Very large organizations might also decide to become a 2nd level domain, for example, AT&T is spread out over much of the United States, but accounts for nearly half the UUCP hosts in the world, so will probably have its own 2nd level domain ATT.UUCP. Small and medium sized organizations are encouraged to join up with other nearby organizations to become regional 2nd level domains, in order to keep the total number of subdomains small. An organization with a few machines may wish to become a 3rd or 4th level subdomain, but should not become a 2nd level domain.

Individual machines will not be allowed to be 2nd level domains, hence, the user@host.UUCP syntax will only be supported until we can get the subdomain framework in place. All hosts that want to become part of the UUCP domain will have to become part of some subdomain. It is not necessary that all hosts attach into the domain tree directly off a second level subdomain; further subdivision is allowed if it makes sense locally.

Individual machines may or may not be 3rd level or lower domains, according to the policies of the 2nd level domain. All individual machines are viewed as Nth level domains, for some N. Thus, if OSGD.CB.ATT.UUCP represents a particular machine, it is also viewed as a 4th level domain.

2. REQUIREMENTS

The following is a list of requirements for becoming a 2nd level domain under the top-level domain UUCP. They are grouped as administrative requirements, standards, services, and guidelines.

2.1 Administrative

This section describes the administrative requirements for subdomains of UUCP.

2.1.1 Conformance with RFC920. We are operating under the guidelines established by the ARPANET document RFC920. That document describes the overall domain structure of the domain tree, and sets forth the requirements for domains. Some of these requirements apply only to top level domains, but many of them apply to all levels. Since subdomains of UUCP will be in the ARPA domain tree, they must conform to the rules specified there. Briefly, these rules are that each subdomain must have a responsible contact person, maintain a registry of all their subdomains and machines and the contact persons for them, provide some sort of access to that registry with a domain server, be at least a certain size, and register with their parent domain.

2.1.2 Responsible Persons. There must be a minimum of two responsible people per subdomain. The main contact should be a technical contact, and the alternate may be either a technical or administrative contact. These people will be responsible to the UUCP Project, and to the UUCP community overall. When a contact person for a subdomain steps down, they must notify their parent domain, and either (a) find a replacement, (b) dismantle the subdomain, or (c) make arrangements with someone to be a temporary replacement.

2.1.3 Size. A subdomain must have a minimum of 100 machines, representing a minimum of 250 users. Exceptions to this rule will be made at the discretion of the UUCP Project. Exceptions are intended for situations where a subdomain is small but isolated from the rest of the community by an expensive bottleneck, for example, Asia and Israel should probably be separate subdomains because of their remote geographic location and the expensive dialup links to them. It is expected that Europe will have one or two subdomains as well.

Note that this requirement is stricter than the 50 machine minimum recommended in RFC920. This is because the UUCP net is larger than the typical network envisioned by the authors of RFC920, is growing faster, and operates using a lower performance transport than the TCP/IP environment assumed in their environment.

Single organizations (such as companies, universities, or government divisions) desiring a 2nd level domain must show that they represent at least 1/100 of the UUCP domain (so that if 100 subdomains are created, such organizational domains will be as large as other subdomains.) A medium sized company that cannot meet this requirement but would like to become a second level domain is encouraged to become a gateway for a larger geographic subdomain in its region, and possibly handle all or part of the domain administration.

Our expectation is that there will initially be 10-20 2nd level domains in the United States, 2-5 in Canada, 2-10 in Europe, one in Australia, and 1-2 in Asia. These numbers are based upon the current distribution of hosts running UUCP, and are subject to revision as needed.

2.1.4 Right of Refusal. We reserve the right to accept or refuse 2nd level subdomain applications. For example, we would not accept two domains with an overlapping general purpose constituency; e.g. two domains that both claim to represent the state of New York.

2.1.5 Application. The responsible person must make application to the UUCP project responsible person (currently Karen Summers-Horton, cbosgd!ksh) outlining who the domain represents, what the name of the domain is, showing how it meets these requirements, the growth plan, and giving the name, postal address, electronic address, and telephone number of both the administrative and technical contacts.

2.1.6 Growth Plan. When a domain grows, it may find that a once workable name space becomes unworkable because of its size, and that it should be subdivided. For example, ``plus5.uucp'' is an accepted convention now, but the size of the domain has grown to the point where subdomains have become essential. As a result, plus5.uucp will probably be renamed plus5.mid-w.uucp, or possibly even plus5.stl.mid-w.uucp. This causes an upward compatibility problem, and the old name must be supported for a reasonable period of time until people are using the new name. This is termed ``growth by lowering'', where hosts become one level (or more) lower in the domain tree.

Growth by lowering is a difficult process, and we would like to avoid it where possible. We therefore ask that all subdomains make estimates of (1) their current size (in number of hosts), (2) size in one year, (3) size in 2 years, and (4) size in 5 years. We request that the subdomain structure of each domain be established so that growth by lowering is not needed for 5 years, if possible, and not for 2 years in any case. This growth plan, including estimates and proposed subdomain structure, should be included when the domain applies for registration in the UUCP domain.

We do ask that an appropriate balance be created between room for growth and length of addresses. If the part of a typical mailing address to the right of an @ sign is longer than 16 characters, chances are that the structure is too bushy. A domain name like osgd.osg.cb.att.uucp might reflect the organizational heirarchy, but is a lot for people to remember and type. Please try to keep the names short and the number of levels small. AT&T is probably the largest subdomain of UUCP, yet addresses no worse than osgd.cb.att.uucp are anticipated.

2.2 Standards

This section describes conventions and standards that all domains are expected to support. Additional standards will be established as necessary by the UUCP project.

2.2.1 Unique Names. Each domain name must be unique within its parent domain. For example, within the UUCP domain, there cannot be two domains named CAL.UUCP and CAL.UUCP. There could, however be two domains called BA.CAL.UUCP and BA.ATL.UUCP, or CAL.UUCP and CAL.CSNET. Note that upper and lower case letters are considered the same in domain names, and that names may cntain the 26 Roman letters A-Z, the digits 0-9, and the hyphen, using periods to separate the levels.

2.2.2 Name Length. No hard limit is placed on the length of names used for addresses in the UUCP domain. However, for human factors reasons, we expect that names chosen will be both representative of the constituency of the subdomain, and short enough that people will not object to typing complete electronic addresses. It is recommended that typical fully qualified domain names be no more than 16 characters long, including periods, and that typical user names on the left of the @ be kept under 16 characters in length as well. For example, the ATT.UUCP subdomain will probably allow electronic addresses in two forms, a machine address form like ``john@ihnp4.ATT.UUCP'' and a person name form like ``John.Smith@ATT.UUCP''. Software should be written to handle addresses of at least 255 characters, since explicit routes can create very long paths.

This is not a hard limit, but rather a guideline. The primary motivation is that short names are easier to type than long names. If there are other overriding considerations, longer names may be necessary. However, in general, we recommend that the number of levels be kept as small as possible, and that names be kept as short as possible, so that addresses are kept easy to type. In many cases, a natural administrative subdomain does not need to be represented in the explicit domain tree, because software is capable of distinguishing a large number of names. For example, given a domain such as ``ernie.cs.berk.uc.cal.uucp'', representing the ``Ernie'' machine of the Computer Science department at the Berkeley campus of the University of California, some of these levels do not really need to be present. The alternative ``ernie.ucb.cal.uucp'' would have done just as well. Two levels of administrative delegation can often be compressed into one level of the domain tree.

Notice also that we avoid longer names such as ``ernie.comp-sci.berkeley.university-of-california.california.uucp''. Try to keep the names as short as possible, while keeping them reasonably readable. Well known (or soon to become well known) two or three letter names for countries, states, provinces, and major geographic regions are accceptable, but don't carry this to an extreme. Three to six letters per domain level is a good typical name length, but there will be many situations where two or eight or ten letter names will be needed.

2.2.3 Software Standards. All subdomains are expected to conform to the appropriate ARPA standards for syntax and semantics of mail and news, including RFC822, RFC920, and RFC850. (News need not be supported, but each contact person is required to have an electronic mail address.)

Mail transferred within a subdomain is an internal matter and can be in any format agreed upon within the subdomain, but all mail leaving the subdomain must appear to external software to have originated on an 822 conforming host, and mail conforming to 822 standards entering the subdomain must be accepted and properly dealt with. It is recommended that internal mail also use the 822/920 syntax, as this makes gateway issues much easier. Two consenting hosts are free to exchange mail or news in any format they mutually agree upon, so long as it does not cause problems for the rest of the network. For example, two hosts may choose to exchange news in notesfile format; there is no problem unless news passing through this link loses information and the resulting news is propagated throughout the rest of the net.

Mail must also conform to the companion document ``UUCP Mail Transmission Format Standard.'' This document summarizes how the 822/920 standards are to be interpreted in the UUCP domain. Subdomains must conform to the UUCP interpretation. In practice, this will mean at least support of one extension, the dom.ain.name!user syntax as being equivalent to user@dom.ain.name.

We expect to provide public domain software that meets these requirements in the next several months, but hosts are free to run any software that conforms to the appropriate standards.

2.3 Services

This section describes ongoing services that each subdomain is expected to provide to their members and to the UUCP community as a whole. Subdomains are encouraged to divide these services up, as much as possible, among the major participants in the domain, in order to share the work load and the traffic load.

2.3.1 Registry. Each subdomain must keep a registry of all machines and subdomains within it. While we do not require the complete registry to be published, it must be possible to determine the organization and contact person for any user, machine, or subdomain within the UUCP domain. For geographic domains, the registry will normally be available to the public, and distributed locally as needed. Private domains may publish their registry or not, as appropriate.

A registry can be through of as a phone book. Top level domains, like country codes, are published everywhere. 2nd level domains, like city codes or area codes, are published in their own country, and available elsewhere. Lower level domains, like local phone numbers, are published locally. PBX extensions within a company, like private domain listings, may be considered private.

There must be only one master copy of the registry for each domain, all others should be copies made from the master copy. (This is to prevent multiple, inconsistent, versions of the registry from appearing, with no final arbiter to determine which is correct, and to make maintenence of the registry practical.) We expect that either a name server will be made available, or else the responsible persons will be able to track down any address within their subdomain and find out who it belongs to. This chain of responsibility is necessary in order to idenfify the source of messages causing problems for other sites, and is a requirement placed on us by the ARPA registry in order to become a top level domain. (These registries are different from the UUCP host name registry, which registers the 6-letter UUCP transport names.) Responses to manual name service queries must include complete information for both contact persons, in order to provide a robust name service.

2.3.2 Domain Server. Once a standard domain server protocol has been documented and public domain software made available to implement it in the UUCP environment, we expect each domain to support such a domain server and allow access to it to anyone. It is not necessary to provide a complete list of all registered hosts, but it is essential that requests of the form ``who is abc.xyz.n-eng.uucp and who is their contact person'' be answered, in order to track down the source of errant messages. It will also probably be necessary to provide a server that maps 6-letter UUCP names into domain names somewhere on the net.

Until a name server has been made available, we adopt the convention that a name service is done by hand. A query such as ``who is the contact person for OSGD.CB.ATT.UUCP'' is resolved by contacting the contact person for the lowest known domain, either by telephone or electronic mail, and asking for the information for the next lower domain. For example, you could phone the ATT.UUCP contact, who would refer you to the CB.ATT.UUCP contact, who would refer you to the OSGD.CB.ATT.UUCP contact. This process will eventually be automated.

2.3.3 Gateway. The subdomain must provide at least one gateway machine for the subdomain. This machine must be able to handle all the traffic between the inside and outside of the subdomain, and must also be willing to forward traffic from outside machine to outside machine. This gateway machine or machines will become part of the UUCP backbone, and complete UUCP connection information for the gateway will be published regularly. Subdomains are encouraged to set up more than one gateway; however, in doing so, they should ensure that all gateways have good solid connections with each other and that all gateways run the same versions of routing tables for the subdomain. External nodes should be free to forward properly addressed mail to any gateway and be sure that the results will be the same as if the mail were forwarded to a different gateway.

2.3.4 Updates. The responsible people will be required to ensure that their parent domain has up-to-date and correct contact and connection information for them. We expect that, unless no information has changed, that gateways will be updated every one week to one month. The contacts for the subdomain will probably want to keep connection information for all internal sites, but are not required to present this information to the UUCP Project.

2.4 Guidelines

This section describes some guidelines for operation of hosts and subdomains. There is more flexibility in conforming to these guidelines than to the requirements above, but reasonable conformance is still expected.

2.4.1 Representative Names. We expect all our subdomains and their subdomains to choose names that are reasonably representative of the constituency of the subdomain. In particular, we discourage subdomain names that are chosen from ``themes'', and subdomain names that are just the name of the gateway. Thus, ``ethel'' (an example from an ``I Love Lucy'' theme) and ``xyzvax'' (a machine name which is also a gateway) should be avoided, in favor of names like ``n-eng'' (New England.) Of course, if the most descriptive name for the subdomain happens to be theme based (e.g. ``homer'' for the machines named ``ulysses'', ``kalypso'', etc, or ``xyz'' if the subdomain is the company named ``xyz'' whose gateway machine is also called ``xyz'') the name will be allowed. In general, a descriptive organizational name or geographic name is preferred, if it is meaningful outside the subdomain.

The intent of this requirement is that it is easier for humans to remember names that are descriptive of the user or the user's organization than ``cute'' names, especially for infrequent users of the system. It is also more helpful when a user receives a message from someone in a domain they don't recognize, if the name is somehow indicative of the location or organization of the sender.

2.4.2 New Machines and Domain Names. If you have a new machine or a new group of machines to register, it is your responsibility to find a domain (at some level) willing to accept your registration. If you belong to an organization that already has a domain registered, you should probably join that domain. If not, there is probably a geographic domain that represents your geographic region, and it should be willing to register you. If your geographic region does not have a domain, you can either find a nearby region and attempt to convince them to expand their borders to include your location; find others in your region and band together with them to create a new domain; or if you are remotely located you can apply for an exception to the minimum 2nd level domain size and create a new 2nd level domain (but this new domain must be willing to register any new machines in your region.)
Choosing a name for a new machine is hard, especially if the owners of the machine are new to the net and unfamiliar with customs and problems that can arise. A good name for the first machine a company gets is the name of the company. It is quite common to name a machine after the type of machine (e.g., ``csvax'' or ``ucbvax'') but this is a bad idea, because if you acquire more machines of the same type the names will be confusing. Plan for the day you have lots of machines, for example, ``framus-a'' or ``a.framus'' if your company name is Framus and your theme is letters of the alphabet, or ``ethel.framus'' if you are naming your machines after an ``I Love Lucy'' theme.

Themes already being used include the Marx Brothers, stars, constellations, Homeric characters, musical tempos, brands of automobile, and the cast of Leave it to Beaver, as well as more mundane themes such as letters of the alphabet, numbers, colors, names of departments, names of the users of personal computers, and so on. Originality is encouraged, as long as the higher level domain name is descriptive of the organization. Bad names include ``unix'', ``bigvax'', ``vax'', ``gateway'', ``sun'', ``framusvax''. Also bear in mind that ``UNIX'', ``VAX'', and similar terms are trademarks of various companies.

2.4.3 Geographic Domains. There are two kinds of domains: geographic and non-geographic. A typical non-geographic domain would represent a particular organization, such as a university, company, government entity, or some other cooperative organization. A geographic domain is one that is intended to register anyone in that geographic region. A geographic domain need not accept top level registrations from sites in the region, but should allow any machine in the region to register somewhere under the domain.

For example, a geographic domain called ``n-eng'' for ``New England'' may subdivide into domains ``boston'', ``nh'', ``mass'', and so on. The ``boston'' domain may in turn have a ``bbn'' subdomain for the BB&N company. A host ``cca'' at the BB&N company should be allowed to join the ``bbn'' domain as ``cca.bbn.boston.n-eng.uucp'', but need not be allowed to join the ``n-eng'' domain directly as ``cca.n-eng.uucp''.

Non-geographic domains may establish any rules and requirements they wish upon their members. Geographic domains may also establish any rules and requirements, but it is expected that a rule obeying host which pays its own way can register somewhere within the geographic domain within which it is located.

It is recommended that all hosts belong to some geographic domain, in addition to any non-geographic domains it joins. This will enable people to send you mail in terms of the geography. For example, the machine ``osgd'' may belong to the ``cb.att.uucp'' domain, but it should also register with the ``cmh.mid-w.uucp'' domain, since it is located in Columbus (cmh) in the midwest (mid-w.)

2.4.4 Initial Domains. To set the flavor of this structure, our intent is that the initial 2nd level domains under UUCP will be along the lines of Figure 1. This is not a firm requirement, just a guideline. (We are still open to suggestions for restructuring this, changing the spelling conventions, splitting a few of these into two domains, and so on.)

Geographic 2nd Level Domains

WA.UUCP	Washington State
OR.UUCP	Oregon State
N-CA.UUCP	Northern California
S-CA.UUCP	Southern California
MTN.UUCP	Mountain states (AZ, UT, CO, NM, WY, ID, MO)
S-CEN.UUCP	South Central states (TX, OK, LA)
MID-W.UUCP	Midwestern states (ND, SD, NB, KS, MN, IA, MO, WI, IL, IN, MI, OH, KY, WV)
S-EAST.UUCP	Southeastern states (AR, TN, MS, AL, GA, NC, SC, FL)
ATL.UUCP	Atlantic States (VA, DC, MD, PA, NJ, NY)
N-ENG.UUCP	New England (MA, CT, RI, VT, NH, ME)
HI.UUCP	Hawaii
W-CAN.UUCP	Western Canada (BC, AB, SK, MB)
E-CAN.UUCP	Eastern Canada (ON, PQ, etc)
EUR.UUCP	Europe
UK.UUCP	Great Britain, United Kingdom and Ireland
AUS.UUCP	Australia
ASIA.UUCP	East Asia, including Korea and Japan

ISRAEL.UUCP	Israel
	Non-Geographic 2nd Level Domains
ATT.UUCP	the AT&T company
HP.UUCP	the Hewlett Packard company

Figure 1. Sample UUCP 2nd Level Domains

This is just a rough guideline, and the actual domains will determine their exact boundaries. For example, we aren't sure where to put Hawaii, or whether it makes sense to include Australia. There is room for a few additional domains, should some of the above be too big. For example, western New York state and Pennsylvania might wish to form their own domain, or North Carolina might. As new parts of the world join UUCP, such as Alaska or Africa, additional domains will be created, as needed. Finally, the names above are also only suggestions.

2.4.5 Routing. The established convention on UUCP is that if two users on different machines exchange mail often, a direct UUCP link should be set up between the two machines. If this is not possible, a short path should be established between the two, and permission should be obtained from all intermediate hosts (since you are running up their phone bill and using their machine cycles.) For seldom used connections, the convention is that others will forward your mail (at their expense) if you will forward their mail (at your expense.)

Domains are not the same as routes. Mail from cbosgd.att.uucp to seismo.arpa does not necessarily travel to machines att.uucp, uucp, and arpa. Direct links and known short paths should be used whenever possible. Routes that go up the tree and back down should be viewed as fallback routes, used only when no better route is known.

3. CONCLUSION

This document is a draft. It does not represent final requirements. Comments and suggestions on these requirements are encouraged. Please send them to cbosgd!mark. cbosgd can be reached via seismo, ucbvax, ihnp4, allegra, decvax, and many other well known hosts. Discussion of the plan on Usenet newsgroup net.mail is also encouraged.

4. FURTHER READING

For additional reading, see:

1. RFC822. Standard for the Format of ARPA Internet Text Messages, August, 1982.

2. RFC882. Domain Names - Concepts and Facilities, November 1983.

3. RFC883 Domain Names - Implementation and Specification, November 1983.

4. RFC920. Domain Requirements, October 1984.

5. RFC921. Domain Name System Implementation Schedule, October 1984.

6. UUCP Mail Transmission Format Standard, UUCP Project, in preparation.

January 17, 1985
DRAFT

From: mark@cbosgd.UUCP (Mark Horton)
Date: Fri, 18-Jan-85 17:16:24 AESST
Newsgroups: net.mail
Subject: UUCP Mail Transmission Formouatt Standard
Organization: Bell Labs, Columbus

UUCP Mail Transmission Format Standard

Mark R. Horton

Bell Laboratories
Columbus, Ohio 43213

ABSTRACT

This document defines the standard format for the transmission of mail messages between machines. It does not address the format for storage of messages on one machine, nor the lower level transport mechanisms used to get the data from one machine to the next. It represents a standard for conformance by hosts in the UUCP domain. We assume remote execution of the rmail command (or equivalent) as the UUCP network primitive operation.

1. Introduction

Our general philosophy is that, if we were to invent a new standard, we would make ourselves incompatible with existing systems. There are already too many (incompatible) standards in the world, resulting in ambiguities such as a!b@c.d which is parsed a!(b@c.d) in the old UUCP world, and (a!b)@c.d in the Internet world. (Neither standard allows parentheses, and in adding them we would be compatible with neither. There would also be serious problems with the shell and with the UUCP transport mechanism.)

Having an established, well documented, and extensible family of standards already defined by the ARPA Internet, we choose to adopt these standards for the UUCP domain as well. While the actual transport mechanism is up to the two hosts to arrange, and might include UUCP, SMTP, MMDF, or some other facility, we adopt RFC920 (domains) and RFC822 (mail format) as UUCP domain standards. All mail transmitted between systems should conform to those two standards. In addition, should the ARPA community change these standards at a later time, our standards will change to remain compatible with theirs, given a reasonable time to upgrade software.

This document specifies an interpretation of RFC822 and

RFC920 in the UUCP world. It shows how the envelope should be encoded, and how UUCP routing is accomplished in an environment of mixed implementations.

2. Basics

Messages can be divided into two parts: the envelope and the body. The envelope contains information needed by the mail transport services, and the body contains information useful to the sender and receiver. Sometimes an intermediate host will add to the body (e.g. a Received line) but, except in the case of a gateway which must translate formats, it is not expected that intermediate hosts will change the body. In the UUCP world, the envelope consists of the ``destination addresses'' (normally represented as the argument or arguments to the rmail command) and the ``source path'' (normally represented in one or more lines at the beginning of the message beginning either ``From '' or ``>From '', sometimes called ``From<space> lines''.) The RFC822 header lines (including ``From:'' and ``To:'') are part of the body, as is the text of the message itself.

2.1 Hybrid Addresses

The UUCP domain, in explicitly conforming to the ARPA Internet standards, adopts the convention that (a) addresses containing ``!'' to the left of ``@'', e.g. hosta!user@hostb.UUCP, may cause unpredictable behavior on other hosts, and production of so-called ``hybrid addresses'' is strongly discouraged; (b) all systems implementing such extensions are strongly urged to use the Internet interpretation, where the ``@'' has priority over the ``!'', that is, the above address would be interpreted as (hosta!user)@hostb.UUCP. For reasons of upward compatibility, however, we recommend that implementations support hybrid addresses. Eventually, it may be possible to phase out the ! syntax, but this is not possible in the near future.

2.2 Transport

Since SMTP is not available to much of the UUCP domain, we define the method to be used for ``remote execution'' based transport mechanisms. The command to be ``remotely executed'' should read

```
     rmail user@domain ...
```

with the message on the standard input of the command. The ``user@domain'' argument must conform to RFC920 and RFC822. More than one address argument is allowed, in order to save transmission costs for multiple recipients of the same message.

An alternative form that may be used is

```
rmail domain!user
```

where ``domain'' contains at least one period and no !'s. This is to be interpreted exactly the same as user@domain, and can be used to transport a message across old UUCP hosts without fear that they might change the address. The ``user'' string can contain any characters except ``@''. This character is forbidden because it is unknown what an intermediate host might do to it. (It is also recommended that the ``%'' character be avoided, since some hosts treat ``%'' as a synonym for ``@''.) However, to route across hosts that don't understand domains, the following is possible

```
rmail a!b!c!domain!user
```

A ``domain'' can be distinguished from a 6 letter UUCP site name because a domain will contain at least one period. (In the case of single level domains with no periods, a period should be added to the end, e.g. Mark.Horton@att becomes ``att.!Mark.Horton''. A translator from ! to @ format should remove a trailing dot at the end of the domain, if one is present.)

2.3 Envelope

The standard input of the command should begin with a single line

```
From domain!user date remote from system
```

followed immediately by the RFC822 format headers and body of the message. It is possible that there will be additional From<space> lines preceding this line - these lines may be added, one line for each system the message passes through. It is also possible that the ``system'' fields will be stacked into a single line, with many !'s in the ``user'' string. The ``>'' character may precede the ``From''. In general, this is the ``envelope'' information, and should follow the same conventions that previous UUCP mail has followed. The primary difference is that, when the system names are stacked up, if previously the result would have been a!b!c!mysys!me, the new result will be a!b!c!mysys!domain!me, where domain will contain at least one period, and ``mysys'' is often the 6 letter UUCP name for the same system named by ``domain''.

The receiving system may discard extra ``From<space>'' lines if it folds the information into a a single From<space> line. It passes the user@domain along as the ``envelope'' information containing the address of the sender of the message, and possibly preserves the date and system in a newly generated header line, such as Received or Sent-By. If the receiving system passes the message along to another system, it will add a ``From<space>'' line to the front, giving the same user@domain address for the sender, and its own name for the system. If the receiving system stores the message in a local mailbox, it is recommended that a single ``From<space>'' line be generated at the front of the message, keeping the date (in the same format, since certain mail reading programs are sensitive to this format), and not using the ``remote from system'' syntax.

It is possible to distinguish UUCP domain generated mail from mail generated by non-UUCP domain hosts, using the ``From<space>'' line. If a host name contains a ``.'', it is in Internet format. If it does not, it should be assumed to be in the old UUCP ``!'' format. (Note - if an intermediate system adds text such as ``system!'' to the front of a ``user@domain'' syntax address, either in the envelope or the body, this is a violation of the standard.)

The ``envelope sender'' information (the From<space> line or lines) are the same as always, except that when the ! address is reconstructed from them, one of the hosts will contain one or more periods, representing the domain address. That is, foo!bar!cbosgd.uucp!cbscc!rlp can be turned into a domain address by stripping everything to the left of the dotted domain (cbosgd.uucp!cbscc!rlp), and placing the dotted domain on the right hand side of the @ (cbscc!rlp@cbosgd.uucp). Note that intermediate systems passing mail from one system to the next should not do this reconstruction, but should keep the envelope in the ! form.

2.4 Routing

In order to properly route mail, it is sometimes necessary to know what software a destination or intermediate machine is running, or what conventions it follows. We have tried to minimize the amount of this information that is necessary, but the support of subdomains requires that different methods are used in different situations. For purposes of predicting the behavior of other hosts, we divide hosts into three classes. These classes are:

Class 1 old-style UUCP ! routing only. We assume that the host understands local user names:

```
rmail user
```

and bang paths

```
rmail host1!host2!user
```

but we assume nothing more about the host. If we have no information about a host, we can treat it as class 1 with no problems, since we make no assumptions about how it will handle hybrid addresses.

Class 2 Old style UUCP ! routing, and 4.2BSD style domain parsing. We assume the capabilities of class 2, plus the ability to understand

```
rmail user@domain
```

if the ``domain'' is one outside the UUCP domain which the host knows about. Class 2 hosts do not necessarily understand domain!user or have routers, but do understand

```
rmail user@host.UUCP
```

if the name ``host'' appears in the L.sys file (e.g. is directly connected.) Some class 2 hosts may serve as gateways for RFC920 subdomains. Hosts in non-UUCP RFC920 domains are considered class 2, even though they may not understand host!user.

Class 3 All class 1 and 2 features are present. In addition, class 3 hosts must be able to handle UUCP mail for hosts that are not immediately adjacent (that is, can respond to

```
rmail user@domain.UUCP
```

even if ``domain'' is not a directly adjacent UUCP host) and also understands the syntax

```
rmail domain!user
```

as described above.

This document describes what class 3 hosts must be able to process. Classes 1 and 2 already exist, and will continue to exist for a long time, but are viewed as ``older systems'' that may eventually be upgraded to class 3 status.

3. Algorithm

The algorithm for delivering a message to an address ``user@domain'' over UUCP links can be summarized as follows:

a. If the address is actually of the form @domain1:user@domain2, the ``domain'' used for the remainder should be ``domain1'' instead of ``domain2'', and the bang form reads domain1!domain2!user.

b. Determine d: the most specific part of ``domain'' that is recognized locally. This part will be a suffix of ``domain''. This can be done by scanning through a table with entries that go from specific to general, comparing entries with ``domain'' to see if the entries are at the tail of ``domain''. For example, with the address ``mark@osgd.cb.att.uucp'', if the local host recognizes ``uucp'' and ``att.uucp'', d would be ``att.uucp''. The final entry in the table will be the null string, matching any completely unrecognized domain.

c. Look in the found table entry for g: the name of the ``gateway'', and for r: a UUCP !-style route to reach g. G is not necessarily directly connected to the local host, but should be viewed as a gateway into the d domain. (The values of g and r for a given d may be different on different hosts, although g will often be the same.)

d. Look at the beginning of r to find the ``next hop'' host n. N will always be directly connected to the local host.

e. Determine, if possible, the class of g and n.

f. Create an appropriate destination string s to be interpreted by n. (See below.)

g. Pass the message off to n with destination information s.

In an environment with other types of networks that do not use UUCP ! parsing, the table will probably contain additional information, such as which type of link to use. The path information may be replaced in other environments by information specific to the network.

The first entries in the table mentioned in part (b) are normally very specific, and allow well known routes to be constructed directly instead of routing through thhe domain tree. The domain tree should be reserved for cases where no better information is available, or where traffic is very light, or where the default route is the best available. If a better route is available, that information can be put in the table. If a host has any significant amount of traffic sent to a second host, it is normally expected that the two hosts will set up a direct UUCP link and make an entry in their tables to send mail directly, even if they are in separate domains. Routing tables should be constructed to try to keep paths short and inexpensive for as much traffic as possible.

Here are some hints for the construction of the destination string n (step f above.) The ``envelope recipient'' information (the argument(s) to rmail) may be in either domain ! form (host.uucp!user) or domain @ form (user@host.uucp) as long as the sending site is sure the next hop is class 3. If the next hop is not class 3, or the sending site is not sure, the ! form should be used, if possible, since it is hard to predict what the next hop would do with a hybrid address.

If the gateway is known to be class 3, domain ! form may be used, but if the sending site is not sure, and the destination is of the form user@host.UUCP, the 6 letter ! form should be used: r!user, for example: dumbsite!host!user. If the gateway appears to actually be a gateway for a subdomain, e.g. by the presence of an address containing multiple dots, such as user@host.gateway.uucp, it can be assumed to be at least class 2, and it is probably class 3. This allows routes such as dumbhost!domain!host.domain.uucp!user to be used with a reasonable degree of safety. (If ``domain'' is a class 2 host, the above may not work, but if there is no direct link to ``domain'', the only alternative is to pass the message off to another class 2 or 3 host with the expectation that it will eventually arrive at the destination.) If a direct link exists to the destination host, the domain @ syntax should be used.

All hosts conforming to this standard are class 3, and all subdomains of the UUCP domain must be class 3 hosts. There may exist a few hosts which are gateways between UUCP and other domains that are class 2, while these gateways are considered in violation of this standard (since there may be no way to reach them) they will probably continue to exist for some time, and we should try to make things work with them where possible.

4. Example

Suppose host A.D.UUCP sends mail to host C.D.UUCP via intermediate host B.D.UUCP. We know that A and C are class 3, but we don't know about B.

The user on A types

```
mail user@c.d.uucp
```

The user interface creates a file such as

```
Date:  9 Jan 1985   8:39 EST
From: myname@A.D.UUCP (My Name)
Subject: sample message
To: user@c.d.uucp

This is a sample message
```

and passes it to the transport mechanism with a command such as

```
sendmail user@c.d.uucp < file
```

The transport mechanism looks up a route to c.d.uucp and finds that the path is bname!cname!%s, and that c.d.uucp is a class 3 host. It prepends a From<space> line and passes it to uux:

```
uux - bname!rmail cname!c.d.uucp!user < file2
```

where file2 contains

```
From A.D.UUCP!user Wed Jan  9 12:43:35 1985 remote from aname
Date:  9 Jan 1985   8:39 EST
From: myname@A.D.UUCP (My Name)
Subject: sample message
To: user@c.d.uucp

This is a sample message

```

(Note the blank line at the end of the message - at least one blank line is required.) This results in the command

```
rmail cname!c.d.uucp!user
```

running on B. B prepends its own from line and passes the mail along:

```
uux - cname!rmail c.d.uucp!user < file3
```

where file3 contains

```
From nuucp Wed Jan  9 12:43:35 1985 remote from bname
>From A.D.UUCP!user Wed Jan  9 11:21:48 1985 remote from aname
Date:  9 Jan 1985   8:39 EST
From: myname@A.D.UUCP (My Name)
Subject: sample message
To: user@c.d.uucp

This is a sample message

```

The command

```
rmail c.d.uucp!user
```

is run on C, which stacks the From<space> lines

```
From bname!aname!A.D.UUCP!user Wed Jan  9 12:43:35 1985
Date:  9 Jan 1985   8:39 EST
From: myname@A.D.UUCP (My Name)
Subject: sample message
To: user@c.d.uucp

This is a sample message

```

and stores the message locally, probably in this same format.

5. Summary

Hosts conforming to this standard should accept all of the following forms:

```
     rmail localuser                (no !@% in user)
     rmail hosta!hostb!user         (no !@% in user)
     rmail user@domain              (only . in domain)
     rmail domain!user              (at least 1 . in domain)
     rmail domain.!user             (in case domain has no dots)
```

The ``envelope'' portion of the message (``From<space>'' lines) should conform to existing conventions, using ! routing. The ``heading'' portion of the message (the Word: lines such as Date:, From:, To:, and Subject:) must conform to RFC822. All addresses must be in the @ form. The originating site should ensure that the addresses conform to 822, since no requirement is placed on forwarding sites or gateways to transform addresses into legal 822 format.

From: jas@mungunni.OZ (John Shepherd)
Date: Thu, 24-Jan-85 10:33:24 AESST
Newsgroups: aus.jokes,aus.mail
Subject: Domain/Network Names
Organization: Machine Intelligence Proj, CompSci, Melbourne Uni

While the controversy rages in aus.mail over the name (case insignificant or otherwise) of this particular Australian Computer Network, these names might provide food for thought, and perhaps even some food for voting at Wollongong:

Network Name	Domain Name*	Why we need this name
GOANNet	GOANNA	my personal obssession
MOZZYNet	FLY	in honour of some local fauna
SIGNet	SWAN	to keep west australians happy (after all, they do have The Cup)
FISHNet	STOCKINGS	to honour Australian women
FISHNet	BARRAMUNDI	to honour Australian fish
GUMNet	EUCALYPT	for Snugglepot and Cuddlepie
PRACTICENet	CRICKET	one of our sporting obssessions
DRAGNet	CONVICT	to honour our beginnings
Cabinet	HAWKE	to honour our future President
BEERNet	XXXX	the domain name is so cute (and it is a popular pastime)
Bassinet	SYDNEY	to keep Basser people happy (impossible! noone here would agree)
Planet	EARTH	*must* be the name for the very top level of the domain hierachy (at least for the next few years)

* I am using upper-case domain names merely by convention.

[Mild Flame]
While I'm on the subject, I find the American obsession with two or three character domain identifiers to be particularly ugly and reminiscent of ancient operating system JCL's.
For example, how would you like to be known as:
bert@ernie.cs.berk.uc.cal.edu
As our expreiences with nroff macro names have shown, there is only *so much* information which can be instilled in one or two characters, and I think that such names belong way down in the mail forwarding system, not in the header of my mail.
I would much rather say:
Bert, c/o Ernie, Comp Sci Dept, University of California, Berkeley, USA
or even
bert@ernie.CompSci.UCBerkeley.USA
and it doesn't look too hard to convert that address into the two char domain name format (in fact, stripping of the first letter from each word nearly dows the trick).

jas@mulga.CompSci.MelbUni.Australia

```
From mark Mon Mar  4 12:20:29 1985
Forwarded to: peteri
>From dpb@aaec  Thu Feb 28 10:09:38 1985
Date: Thu, 28 Feb 85 10:09:38 edt
From: dpb@aaec (Phil "I never learn" Belbin)
To: mark@elecvax
Subject: UNIX
```

UNIX for large system/370 users.

IBM has taken a step into the UNIX world with the announcement of IX/370.

IX/370 is a multiuser,multitasking system with the ability to run a number of independent or interrelated tasks simultaneously.

IX/370 is based on UNIX System V release 2 and provides consistent function across IBM's System 370 processor range. It offers an excellent growth path as well as the ability to match system capacity to customer requirements.

System 370 architecture,operating with VM,provides a proven and reliable base for UNIX.Existing knowledge of both System/370 and VM can be used to support the installation and operation of UNIX functions.

IX/370 can be used for the consolidation of multiple small UNIX systems onto a larger mainframe and offers many advantages, such as data sharing and simpler operation.There will also be occasions when customers find existing UNIX-based applications which can be transfered to a single IBM system to create a unified information system.

Full-duplex ASCII devices such as the IBM 3101 or the IBM PC are supported through a channel-attached Series/1 and the IX/370 ASCII Control feature.And because IX370 runs under VM,other VM functions such as CMS can also be run on the same IBM processor

Certain additional functions have been added to the UNIX System V base by IBM.They help make IX/370 an excellent IBM alternative for customers with growing UNIX systems..

QUOTE..GASP...

RUSS

Australian Unix systems User Group
Financial statement for 1984/85 to 20th February 1985

Credits

Date	Amount	Comments
01/07/84	$3144.51	Value of Assets (Opening Bank Balance)
27/09/84	$670.00	Memberships/Subscriptions/Backissues
08/10/84	$110.00	Memberships/Subscriptions/Backissues
28/12/84	$3466.29	Melbourne Aug 84 Meeting Net Income
07/01/85	$548.00	Memberships/Subscriptions/Backissues
08/01/85	$80.00	Memberships/Subscriptions/Backissues
04/02/85	$528.00	Memberships/Subscriptions/Backissues
08/02/85	$304.00	Memberships/Subscriptions/Backissues
13/02/85	$1315.00	Memberships/Subscriptions/Backissues
18/02/85	$519.00	Memberships/Subscriptions/Backissues
	$10684.80	Total

Debits

Date	Amount	Comments
08/10/84	$9.90	Secretary (Minute book etc.)
08/10/84	$665.40	Special Executive meeting (UNIXWORLD).
26/11/84	$10.40	Piers Lauder taxis to 17,23/10 meetings
11/12/84	$1155.00	AUUGN Vol 5 #6
21/01/85	$12.12	Secretary (Post Box fee)
04/02/85	$120.00	Melbourne Uni - refund of overpayment
	$1972.92	Total

Balance

	$1972.92	Total debits
18/02/85	$8711.88	Value of Assets - Account #906-419
	$10684.80	

Membership report at 20th February 1985

Membership Type	Financial	Unfinancial	Total
Founding	28	45	73
Life	0	-	0
Ordinary	44	0	44
Student	0	0	0
Newsletter Sub.	18	108 *	126
Total	90	153	243

* The 108 unfinancial subscribers are those who subscribed to volume 5 of the newsletter but have neither converted to founding membership nor resubscribed to volume 6.

Financial Statement for AUUG Newsletter Volume 5

Credit	Debit	Comments
0.00		Opening Bank Balance - New Account
700.00		Initial funding from AUUG
4681.87		Volume 5,6 subs and backissues
150.00		Mailing labels
	20.00	Registration of AUUGN as postal category B
	51.55	Rubber stamps for AUUGN envelopes
	90.00	Reimbursement of UNSW for survey mailing
717.00		Further newsletter funds from AUUG
	914.08	Printing of volume 5 Number 1 (106 printed pages)
	114.94	Mailing of volume 5 Number 1 (issue cost $1029.02)
	870.00	Printing of volume 5 Number 2 (90 printed pages)
	76.80	Mailing of volume 5 Number 2 (issue cost $946.80)
	1088.00	Printing of volume 5 Number 3 (122 printed pages)
	96.75	Mailing of volume 5 Number 3 (issue cost $1184.75)
1000.00		Further newsletter funds from AUUG
	1100.00	Printing of volume 5 Number 4
	101.80	Mailing of volume 5 Number 4 (issue cost $1201.80)
	1088.08	Printing of volume 5 number 5
	66.43	Mailing volume 5 number 5 (issue cost $1154.51)
1155.00		AUUG reimbursment for volume 5 number 5
	986.00	Printing of volume 5 number 6 (88 printed pages)
	114.95	Mailing of volume 5 number 6 (issue cost $1100.95)
	350.00	Data preparation and entry for Software/Hardware database
	365.94	EBCO Microfilming - Production of back issue microfiche
	30.00	Registration of AUUGN for 1985
	105.00	Lunch for AUUGN volume 5 volunteers
	24.14	Urgent delivery of mailing labels to W'gong U
	6.61	Bank and Government Charges
31.48		Interest

	7661.07	
	774.28	Current balance of AUUGN account
--------	--------	
8435.35	8435.35	

Total spent on AUUGN Vol 5 - $6617.83 (approx $5 per copy)

Australian UNIX* systems User Group
(AUUG)

Membership Application

I, ____ do hereby apply for ordinary($50)/student($30)** membership of the Australian UNIX systems User Group and do agree to abide by the rules of the association especially with respect to non-disclosure of confidential and restricted licensed information. I understand that the membership fee entitles me to receive the Australian UNIX systems User Group Newsletter and I enclose payment of $_______ herewith.

Signed ________________________________ Date ______________________

===

Name __________________________________

Mailing address for AUUG information _______________________________

____

____

Telephone number (including area code) _____________________________

UNIX Network address ____

	YES	NO
I agree to my name and address being made available to software/hardware vendors	□	□

===

<u>Student Member Certification</u>

I certify that ____ is a full-time

student at ____

Expected date of graduation ____

Faculty signature ______________________________ Date _______________

===

Office use only

10/84

* UNIX is a trademark of AT&T Bell Laboratories

** delete one

Australian UNIX* systems User Group Newsletter
(AUUGN)

Subscription Application

I wish to subscribe to the Australian UNIX systems User Group Newsletter and enclose payment of $__________ herewith for the items indicated below.

Signed ______________________________________ Date __________________________

==

□	One years subscription (6 issues) available on microfiche or paper	$30.00
□	Back issues of Volume 1 (6 issues) available only on microfiche	$24.00
□	Back issues of Volume 2 (6 issues) available only on microfiche	$24.00
□	Back issues of Volume 3 (6 issues) available only on microfiche	$24.00
□	Back issues of Volume 4 (6 issues) available on microfiche, some paper copies	$24.00
□	Back issues of Volume 5 (6 issues) available on microfiche or paper	$24.00
□	Subscribers outside Australia must add an extra to cover surface mail costs	$10.00
□	Subscribers outside Australia must add an extra to cover air mail costs	$30.00

Name ____

Mailing address ____

____

____

Telephone number (including area code) ____________________________________

UNIX Network address ____

I agree to my name and address being made available to software/hardware vendors

YES □ NO □

10/84

* UNIX is a trademark of AT&T Bell Laboratories

Peter Ivanov
AUUGN Editor
School of EE and CS
University of New South Wales
PO Box 1
Kensington NSW 2033
AUSTRALIA

+61 2 697 4042