AUUG

Australian UNIX systems User Group Newsletter

AUUGN

Volume 10, Number 6

December 1989

Registered by Australia Post, Publication Number NBG6524

The Australian UNIX* systems User Group Newsletter

Volume 10 Number 6

December 1989

CONTENTS

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

AUUG General Information

Memberships and Subscriptions

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

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

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

General Correspondence

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

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

AUUG Executive

President **Greg Rose**

greg@softway.sw.oz
Softway Pty. Ltd.
New South Wales

Secretary **Tim Roper**

timr@labtam.oz
Labtam Information Systems Pty. Ltd.
Victoria

Treasurer **Michael Tuke**

mjt@anl.oz
ANL Limited
Victoria

Committee Members

Peter Barnes

pdb@uqcspe.cs.uq.oz
Computer Science
University of Queensland

John Carey

john@labtam.oz
Labtam Information Systems Pty. Ltd.
Victoria

Pat Duffy

pat@pta.oz
Pyramid Technology Corporation
New South Wales

Chris Maltby

chris@softway.sw.oz
Softway Pty. Ltd.
New South Wales

Next AUUG Meeting

The AUUG90 Conference and Exhibition will be held
from September 25th to September 28th 1990.
The venue will be the World Congress Centre, Melbourne.

AUUG Newsletter

Editorial

That mad rush at the end of the year is affecting us all, so this will probably arrive too late for me to wish you all a Happy Xmas and Merry New Year (or a Happy Hanukkah for those who prefer). Hope you all had a good and safe time anyway.

Local user groups are starting to emerge from the woodwork and get themselves registered as Chapters of AUUG. WAUG (the West Australian UNIX systems Group) presented a petition to the AUUG management committee at the last committee meeting, and so now it is a Chapter. When I last heard, SWiGS (the Softway Wine Guzzlers Society) was still looking for ten members who could write. (:-)

In other AUUG news, AUUG and Prentice Hall have come to an arrangement that is to be known as the "AUUG Book Club". Under this arrangement, Prentice Hall provide copies of books to be reviewed in AUUGN. When these books are reviewed, there will be an opportunity for AUUG members to purchase them at a 20% discount. You may see me calling for book reviewers here and in the aus.auug newsgroup, so please consider offering your services as a reviewer. The first offer should appear in the next issue of AUUGN (Volume 11, Number 1).

And something else for you to think about. Currently there are two social activities associated with the annual AUUG Winter Conference and Exhibition: the Wednesday evening cocktail party and the Thursday night dinner. The AUUG Management Committee is looking into adding a third social event, and is would appreciate some suggestions. If you have any ideas, please write to myself or any of the committee members.

Finally, I am looking for someone to help me out. As editor of AUUGN, I get about twenty press releases sent to me each month. Mostly they get filed in the little round bucket beside my desk. It would be good if someone were to volunteer to scan these releases for useful information and produce a few pages for AUUGN every two months. That doesn't sound like hard work, does it?

I hope you all attend your local Summer Technical Meeting, and that I will see a good number of you at the Melbourne Meeting.

AUUGN Correspondence

All correspondence reguarding the AUUGN should be addressed to:-

David Purdue
AUUGN Editor
Labtam Information Systems Pty. Ltd.
43 Malcolm Road
Braeside Victoria 3195
AUSTRALIA

ACSnet: *davidp@labtam.oz*

Phone: +61 3 587 1444
Fax: +61 3 580 5581

Contributions

The Newsletter is published approximately every two months. The deadline for contributions for the next issue is Friday the 19th of January 1990.

Contributions should be sent to the Editor at the above address.

I prefer documents to be e-mailed to me, and formatted with troff. I can process mm, me, ms and even man macros, and have tbl, eqn and pic preprocessors, but please note on your submission which macros and preprocessors you are using. If you can't use troff, then just plain text please.

Hardcopy submissions should be on A4 with 30 mm left at the top and bottom so that the AUUGN footers can be pasted on to the page. Small page numbers printed in the footer area would help.

Come on, everyone, contribute! If the muse is upon you, and you have to write, but you can't think of anything to write about, give me a call and I'll throw some ideas at you.

Advertising

Advertisements for the AUUG are welcome. They must be submitted on an A4 page. No partial page advertisements will be accepted. The current rate is AUD$ 200 dollars per inside page. More prominent positions are available but cost more. Contact the editor for details.

Mailing Lists

For the purchase of the AUUGN mailing list, please contact Tim Roper.

Back Issues

Various back issues of the AUUGN are available, details are printed at the end of this issue.

Acknowledgement

This Newsletter was produced with the kind assistance and equipment provided by Labtam Information Systems Pty Ltd.

Disclaimer

Opinions expressed by authors and reviewers are not necessarily those of AUUG Inc, its Newsletter or its editorial committee.

MHS

MHSnet

The Wide Area Network Solution

MHSnet is the recently released commercial version of the software that forms the backbone of ACSnet. It is superior to UUCP in almost every respect.

Whether it be formal business documents such as EDI, multimedia correspondence, file transfers, program to program communication or simple mail messages, MHSnet provides a cost-effective solution for private wide area networking.

☑ Store and forward architecture

☑ Multiple message protocols for handling any binary message

☑ Dynamic routing based on quality of service, throughput and cost

☑ Automatic topology management of any network structure

☑ Broadcast and multicast addressing

☑ Fault-tolerant network reconfiguration after link or machine failure

☑ Throughput that approaches the maximum possible link capacity

☑ Operation over any virtual circuit on any Unix system

☑ Error recovery without loss of transmitted data

☑ Multi-channel, full-duplex transmission

☑ X.400 internal addressing and OSI interfaces

Message Handling Systems Pty Limited
Level 67, MLC Centre, Martin Place Sydney 2000
Telephone: (02) 260 0612 Facsimile: (02) 484 6761 ACSnet: enquiry@mhs.oz.au

Conference Announcement

AUUG 90

Australian UNIX* systems User Group

Conference and Exhibition 1990

September 25-28 1990, Melbourne, Australia

Summary

The 1990 Conference and Exhibition of the Australian UNIX systems User Group will be held at the World Congress Centre, Melbourne, Australia. Tutorial sessions will be held on Tuesday the 25th and the conference proper from Wednesday the 26th to Friday the 28th September 1990.

The conference theme is:

UNIX the Computing Platform for the 90s

Venue

The World Congress Centre is a new purpose built convention and exhibition centre located near the Yarra River. It is within the Central Business District with easy access to transport.

This is a major step up for the AUUG in the quality and size of venue and is in step with the growth of the UNIX operating system.

This Conference and Exhibition is to be held during the week before the VFL Grand Final and gives attendees the chance to attend Melbourne's Premier Sporting Event.

Conference

The Conference, held over three days, will provide UNIX users a chance to hear speakers from Australia and overseas present papers on a wide range of topics including the latest developments and uses of the UNIX operating system.

The conference dinner and the conference itself provide an unique opportunity to meet other people in the UNIX community.

Exhibition

The exhibition will be held in an attractive and well serviced venue, and is supported by major UNIX vendors. It is held in conjunction with the AUUG 90 conference which ensures exhibitors suitable contact will be made with potential buyers of their product.

Interested Exhibitors should contact ACMS promptly to ensure they obtain the optimum location to display their product. The ACMS contact address is given below.

Conference Secretariat

For all enquiries regarding registration, accommodation, and the Exhibition:

AUUG 90 Secretariat
c/o ACMS
26 Hopewell Street
Paddington NSW 2021
AUSTRALIA

Telephone:	International	+61 2 332 4622
	National	(02) 332 4622
Facsimile:	International	+61 2 332 4066
	National	(02) 332 4066

Please Note Change in venue and dates from previous announcements

Call For Papers

AUUG 90

Papers

Papers are invited on topics which will interest an audience of either Research, Technical, Industry, or Commercial UNIX users.

Some suggested topics are:-

- Future Directions
- Standards
- Networking
- Security
- Project Management
- Productivity Tools
- Database
- System Administration
- User Interfaces
- Windowing Systems
- Real Time Systems
- Multiprocessing

Papers that provide broad overviews, technical review, and/or descriptions of new and interesting work in the the subject areas are sought. Papers that describe current Work in Progress, and papers on other topics not listed but relevant to the UNIX user community are also welcome.

The primary author of each paper accepted will receive ONE complementary admission to the conference and the dinner.

AUUG will again hold a competition for the best paper by a full time student at an Australian educational institution. The prize for this competition will be an expense paid return trip from within Australia to the conference to present the winning paper. A cash prize in lieu of this may be made at the discretion of AUUG. Students should indicate with their abstract whether they wish to enter the competition. AUUG reserves the right to not award the prize if no entries of a suitable standard are received.

A special issue of the group's newsletter, AUUGN, containing the conference proceedings will be printed for distribution to the attendees at the conference and mailed to AUUG members who do not attend.

A 1000-2000 word extented abstract is required in early February 1990 which describes the nature of the paper and a summary of conclusions and/or results.

We also require that the author send their full contact details with their extended abstract, plus a photograph and a brief C.V. for publicity purposes. Note that full contact details should include full name and address, telephone number, facsimile number, and e-mail address. The author's picture should be a 7 x 8 or 8 x 10 inch black and white photograph or a color transparency.

Acceptance of papers will be based on an extended abstract and will be subject to receipt of the final paper by the due date. The Programme Committee Chair reserves the right to withhold final acceptance until the final paper is received. Abstracts and final papers should be submitted to:-

John Carey
AUUG 90 Programme Committee Chair
Labtam Information Systems Pty. Ltd.
43 Malcolm Road
Braeside Victoria 3195
AUSTRALIA

Phone:	International	+61 3 587 1444
	National	(03) 587 1444
Fax:	International	+61 3 580 5581
	National	(03) 580 5581
Telex:	LABTAM AA335500	
Internet:	john@labtam.oz.au	
ACSnet:	john@labtam.oz	
UUCP:	uunet!munnari!labtam.oz!john	

Call For Papers

AUUG 90

continued

Final Papers

Final Papers should contain a 100-300 word abstract, 10-20 pages of 10 point single spaced text with illustrative figures or diagrams where appropriate. Papers should reference other related work and contain citations to the relevant literature. They should be submitted in camera ready form in a high quality format, on single sided pages with 25mm margins from papers edge, with small page numbers at the centre-bottom of the page. They must be produced using a high quality printing device (300 dpi or better). The only form that will be acceptable via e-mail is PostScript**. Authors unable to meet these standards are welcome to contact the programme chair to arrange suitable output.

AUUG will require each author to sign a release to AUUG, but the author will retain copyright over their paper.

Timetable

Receipt of Extended Abstracts	Monday 5th February 1990
Letters of Acceptance	Monday 5th March 1990
Receipt of Final Papers	Monday 6th August 1990
Tutorials	Tuesday 25th September 1990
Conference and Exhibition	26th-28th September 1990

Tutorials

People wishing to present tutorials should contact

David Purdue	Phone:	International	+61 3 587 1444
AUUG 90 Tutorials		National	(03) 587 1444
Labtam Information Systems Pty. Ltd.	Fax:	International	+61 3 580 5581
43 Malcolm Road		National	(03) 580 5581
Braeside Victoria 3195	Telex:	LABTAM AA335500	
AUSTRALIA	Internet:	davidp@labtam.oz.au	
	ACSnet:	davidp@labtam.oz	
	UUCP:	uunet!munnari!labtam.oz!davidp	

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

** PostScript is a trademark of Adobe Systems Incorporated.

WAUG

Western Australian UNIX systems Group

PO Box 877, WEST PERTH 6005

Western Australian Unix systems Group

The Western Australian UNIX systems Group (WAUG) was formed in late 1984, but floundered until after the 1986 AUUG meeting in Perth. Spurred on by the AUUG publicity and greater commercial interest and acceptability of UNIX systems, the group reformed and has grown to over 70 members, including 16 corporate members.

A major activity of the group are monthly meetings. Invited speakers address the group on topics including new hardware, software packages and technical dissertations. After the meeting, we gather for refreshments, and an opportunity to informally discuss any points of interest. Formal business is kept to a minimum.

Meetings are held on the third Wednesday of each month, at 6pm. The (nominal) venue is "University House" at the University of Western Australia, although this often varies to take advantage of corporate sponsorship and facilities provided by the speakers.

The group also produces a periodic Newsletter, YAUN (Yet Another UNIX Newsletter), containing members contributions and extracts from various UNIX Newsletters and extensive network news services. YAUN provides members with some of the latest news and information available.

For further information contact the Secretary, Skipton Ryper on (09) 222 1438, or
Glenn Huxtable (glenn@wacsvax.uwa.oz) on (09) 380 2878.

Glenn Huxtable,
Membership Secretary, WAUG

SESSPOOLE

Who is SESSPOOLE?

SESSPOOLE is the South Eastern Suburbs Society for Programmers Or Other Local Enthusiasts. Here we are talking about the South Eastern Suburbs of Melbourne.

What is SESSPOOLE?

SESSPOOLE is a group of programmers and friends who meet every six weeks or so for the purpose of discussing UNIX, drinking wines and ales, and just relaxing and socialising over dinner. Anyone who subscribes to the aims of SESSPOOLE is welcome to attend our meetings, whether they come from the South Eastern Suburbs or not. The aims of SESSPOOLE are:

> To promote knowledge and understanding of the UNIX system, and of similar or related computer systems; and to promote knowledge and understanding of red and white wines, and of similar or related wines.

SESSPOOLE is also the first Chapter of the AUUG to be formed, and its members are involved in the staging of the AUUG Summer'90 Melbourne Meeting.

When is SESSPOOLE?

The next meeting of SESSPOOLE is on Tuesday, the 6th of February 1990, at 6:30pm. This will be just after the AUUG Summer'90 Melbourne Meeting.

Where is SESSPOOLE?

The next meeting of SESSPOOLE will be held in the Beer Garden (or the Bistro if it rains) of the Notting Hill Hotel, 262 Ferntree Gully Road, Clayton. Just a hop, skip and jump from Monash University.

Want to know more?

To find out more about SESSPOOLE and SESSPOOLE activities, contact either David Purdue *<davidp@labtam.oz>* or John Carey *<john@labtam.oz>*. Their phone number is 587-1444 (bh). Or look for announcements in the newsgroup **aus.auug**.

AUUG - Summer '90 (Victoria)

Tuesday 6 February 1990
Monash University, Clayton VIC

Tentative Programme

0830-0930	**Registration**
0930-1000	KEYNOTE ADDRESS *Public Key Privacy and Authentication* Greg Rose (President AUUG) Softway Pty. Ltd.
1000-1030	*Unix Security aspects in the US DoD* Dr Mark Anderson Defence Science & Technology Organisation, South Aust.
1030-1100	**Coffee Break**
1100-1130	*MUSBUS – What as Been Learnt* Dr Ken J. McDonell (Invited) Pyramid Technology Corporation
1130-1200	Allen Nemeth (Guest Speaker) Prime Computers
1200-1230	*Using Unix as a Persistent Programming Environment* Dr John Hurst Computer Science, Monash University
1230-1330	**Lunch**
1330-1400	*Cexp – An expression parser for C* Douglas Ray Melbourne University
1400-1430	*Execution Driven Multiprocessor Simulation* Dr Rhys Francis Concurrency Research Group, La Trobe University
1430-1500	*Shared Data Access Rates as a Metric for Distributed Algorithm Performance.* Arnold N. Pears (Invited) Concurrency Research Group, La Trobe University
1500-1530	*Design and Implementation of a Parallel Unix Kernel* Lim Or Sim Computer Science, Monash University
1530-1600	**Coffee Break**
1600-1630	*Network Options for a Unix Supercomputer* Robert Smart CSIRO Department of Information Technology

1630-1700 *STELNET – Secure Remote Login Frontend and Shell*
Lawrie Brown
Australian Defense Force Academy, Canberra

1700-1730 *Design of the Labtam Xengine*
David Purdue, Tim Roper, Michael Podhorodecki
Labtam Information Systems Pty Ltd

1830- SESSPOOLE Meeting

Registration forms for **AUUG – Summer '90 (Victoria)** have already been distributed. However, if you haven't yet received one, and would like one, please contact me at:

Stephen Prince
AUUG–SUMMER90
Labtam Information Systems Pty. Ltd.
43 Malcolm Road
Braeside, Vic. 3195

Phone:	(03) 587–1444
Fax:	(03) 580–5581
Telex:	LABTAM AA33550
E–mail:	sp@labtam.oz.au

Stephen Prince
AUUG – Summer90 (Vic) Programme Committee Chair

Unix Design And Tuning Issues

Jack Dikian

Q.H. Tours Ltd.
141 Walker St. North Sydney NSW 2060
jack@teti.qhtours.oz.au

ABSTRACT

This paper takes a leisurely look at UNIX* and factors we think important in achieving maximum systems performance. Fine tuning a UNIX system, and indeed any complex operating environment, often requires a deep understanding of the underlying system design; as well as an understanding of the relevant business requirements. While a few sites may operate systems with very specific objectives — order entry and stock control for example — many sites will use the same system for a variety of job mixes. Finite system resources are required to be shared amongst a set of competing processes while at the same time maintaining a high level of resource utilization, providing resource acquisition within reasonable time frames, and minimizing overheads due to mechanisms and policies of resource allocation and scheduling implementations. Memory and disk utilization, load manipulation, and system configuration will be reviewed. We will also spend a little time discussing monitoring tools available under UNIX A great deal of this discussion is based around general operating system concepts. As is often the case, points pertaining to specific system design and implementation differ greatly even across versions and flavors of the same operating system. Much of the discussion here is based on a UNIX SYS V derivation, although the concepts can equally apply to all UNIX environments.

Tuning a system involves tailoring the system in order to meet the requirements of users in the most efficient manner. System requirements differ from one site to another even when compatible platforms are in use. Because each system may have different user requirements, tuning systems is sometimes perceived as a non-exact science. Tuning also does not expand the capabilities of a system beyond it's hardware capacity. Upgrading or migrating to a more powerful platform may be the only answer to poor performance.

Memory ranks high among the important resources UNIX manages. In large systems, the demand for memory at a given time often exceeds the total available. The system must allocate memory to processes waiting to use it [Comer84]. Utilizing memory effectively is one of the most important considerations in ensuring good system performance. Memory allocation can take the form of *swapping*, where entire processes are written (swapped) to disk, or *paging*. A paged system divides the virtual address space into small pieces, *pages*, of equal size. Main memory is similarly divided into page frames of the same size. The page frames are shared between the processes currently in the system, so that at any time a given process will have a small subset of pages resident in main memory (active pages) while the remaining pages are resident on disk [Lister81].

Many UNIX systems today offer the choice of using a swapping kernel and/or a paging kernel. The decision of whether to use a swapping kernel or a paging kernel depends upon the amount of system memory available,

* UNIX is a trademark of AT&T.

and the total size of the working sets for executing processes. Systems configured with little relative memory will always require demand paging kernels. In systems where the load is such that either kernel must swap processes, but not excessively, then it is typical to observe that the paging kernel will be slightly faster than the swapping kernel. However, as the system load increases, the performance of both kernels decreases. The paging kernel can, in such a situation reach a point where performance degrades drastically. The load reaches a point where the entire time slice for a process is spent bringing in its pages. It may be useful to increase the time quantum. Systems where the load is such that neither kernel must swap processes, the paging kernel will deliver faster start-up time in process execution while the swapping kernel will perform slightly better as the overhead of dynamic page management is eliminated.

When determining memory requirements, user processes must be considered along with kernel structures, daemons, device drivers, message queues, and even semaphore sets. The extent to which text sharing is carried out is also of important consideration when determining memory requirements. It is possible to determine if a process is using shared text by simply typing ``file *file-name*''. If the file type is ``pure executable'', then the process is using shared text. An ``executable'' file type implies that the process is using non-shared text. Processes using shared text may on one hand utilize memory more economically as only one copy of the text segment exists across many processes using it, however, a demand paging kernel will need to load the entire text. UNIX provides a facility by which the user can force the system to keep a copy of the shared text in a contiguous area of the swap device. This is achieved through the ``sticky'' (t-bit) permission bit. Setting the sticky bit often results in faster start-up times.

It is a relatively simple process to calculate the size of a process if we appreciate that each process is composed of five main sections. The process contains the data segment, holding such things as file information, the text segment, initialized and uninitialized data segments, and the stack. Assuming we are using a 2K block system, multiply the size of the process returned via ps by 2K. The value returned via ps does not include the size of shared text. The UNIX ``size'' utility returns the size of text, data, and bss (uninitilized data) sections of a common object file. The text size returned by the size utility should be added to the number calculated via ps to give the total memory used.

Memory requirements arising from system processes and daemons can be analyzed in a similar fashion. It is important to note however, that at any instant, not all system processes are active. Where the swapper process is always locked in memory and executing, login will be executing only when a terminal is logging in. errdemon, cron, init, lpsched, and pagedaemon are also always executing. Some of these processes are run once per system, while still others are run many times over. For example, at any instant, there will only ever be a single swapper process executing while the system will be supporting as many init processes as there are terminals waiting to be used.

Good systems performance may, to a great extent, be achieved by effective disk utilization. Disk utilization involves organizing the file system as well as reviewing I/O operations on the disk. The UNIX file system [Thompson] is a disk data structure, viewed as a randomly addressable array of 2048-byte blocks. The file system breaks the disk into four main regions. The first block (address 0) is left aside for booting procedures. The second block, the *superblock*, contains housekeeping information such as the size of the disk, boundaries of the other regions, address of the first free storage block list etc. The third main region is the so called *i-list*, a list of file definitions. Each file is defined by a 64-byte structure, called an i-node. The offset of a particular i-node within the i-list is its i-number. There is one i-node for each file on the file system. A file in the UNIX file system is nothing more than a one-dimensional array of bytes. Files are attached onto a hierarchy of directories. Directories are accessed in exactly the same manner as an ordinary file. UNIX supports four types of files. These are ordinary files which are read and written by users. Files are essentially a sequence of unlimited blocks. In practice, UNIX restricts the file size to 1G (now bigger due to larger block sizes) through its file system implementation. Other file types include the *directory* which contain names of files as well as mapping information, *named pipes* which are pipes with a name in a directory, and *special files*. Special files have no storage blocks associated with them. Special files are used to access devices. In the fourth and last region of the disk comes the free storage area, available to hold the contents of the files.

The free space on the disk is maintained by a linked list of available disk blocks. Every block in the list contains the address of the next free block in the chain. The remaining space in the block contains the address of an additional 50 free blocks. A single I/O operation returns 50 free blocks and a pointer to an additional free block.

In time, file systems become more and more fragmented as storage blocks are consumed and or released. As files grow and shrink, the additional blocks may no longer be in the original allocated sequence. The organization of the free list affects the efficiency of file accesses. Files created after the initial system installation, as a general rule, will always tend to be stored less efficiently.

The dynamic parameters [Tunefs] of a file system which affect the layout policies on disk can also be altered. For example, it is possible to define the maximum number of contiguous blocks that are to be laid out before forcing a rotational delay. The amount of rotational spacing between successive blocks (interleave factor) can be adjusted, as well as adjusting the maximum number of blocks any single file can allocate out of a cylinder group before it is forced to begin allocating blocks from another cylinder group. This is typically set to about one quarter of the total blocks in a cylinder group. The intention here, is to prevent any file from using up all the blocks in a single cylinder group, thus degrading access times for all files subsequently allocated in that cylinder group. For file systems with exclusively large files, this parameter can be increased. The minimum free space threshold, or the percentage of space held back from normal users can be changed from the default 10%. It is said that when reducing this value to 0; up to a factor of three in throughput will be lost over the performance obtained at a 10% level.

It is important to maintain a clean directory organization. Directories retain the largest size they have ever achieved. When a file is removed from a directory, its i-node number is simply zeroed out thus leaving an unused slot. During a search of a directory, all blocks of the directory are read until the file is found. A directory containing a large number of deleted files will result in the system reading a large number of useless free blocks before the file entry is found. Consider a directory containing 200 files with the first 170 files deleted, an ls on this directory, looking for a particular file, requires 4 I/O operations (3 for the first 150 free blocks and 1 for the block containing the file looked for).

`fsck -s` can be used to reconstruct the free list by rewriting the superblock. The "`-s`" option allows creation of an optimal free-list organization, taking into account the number of blocks per cylinder and the interleave factor. `dcopy` copies [NCRSuMan] a file system onto a new file system. It can be used to compress directories by removing the vacant entries, and spacing consecutive blocks in a file by the use of the optional rotational gap. Files should also be distributed across multiple disk drives and directories that are frequently modified should not sit on the root file system. Partitioning may be used to increase disk access efficiency. The directories in the `PATH` variable are searched for each command execution. A change in this variable so that large directories are searched last, and the most likely places to find a command appear first, is useful in increasing overall systems performance.

It was mentioned in our abstract that we will briefly review systems monitoring tools. Rather than reproduce much of the text describing monitoring tools, we recommend your UNIX user's manuals for the most definitive statements on availability, usage, and support. Our aim here however, is to draw together the concepts discussed so far and provide general examples of how these tools may be used in practical situations. There are three main classes of monitoring tools available in a UNIX environment. These include the general *Process Accounting* facility which compiles, and reports on such things as the total CPU (system+user) used, average core, and average number of i/o operations clocked up for all executed processes. Another main monitoring facility is the internal system activity reporter (`sar`).

UNIX maintains a number of counters that are incremented as various system actions occur. These include CPU utilization counters, buffer usage counters, disk and tape I/O activity counters, TTY device activity counters, switching and system-call counters, file-access counters, queue activity counters, and counters for interprocess communications.

The `sar` utility can be used to review the above counters. This utility provides various options to allow the user to review subsets of the collected data. For example, specifying "`-u`" flag prompts `sar` to report on only CPU utilization information, "`-d`" produces disk activity information etc. Below is an overview of some of the system activity reporter options and how we may interpert there results.

`sar -u` produces the %usr, %sys, and %idle. These indicate the percentage of time the processor is in user and system mode as well as the percent of time the processor is idle. In cases where the percent idle is consistently less than 10 indicates an under-configured processor for the particular environment.

`sar -b` produces the system buffer activity information. The more the system can avoid physical i/o due to the required blocks being already in the system buffers, the more efficient the system will be. This option displays the average number of physical and logical blocks read and written. The %rcache and %wcache variables indicate the fraction of logical reads and writes carried out through the buffer area. The higher these numbers, the more efficient the buffer area is. A good buffer size configuration should produce %rcache and %wcache of more than 85.

`sar -d` produces block device such as disk and tape activity information. This includes numerous information including percent of time the device was busy servicing a transfer, average number of requests outstanding during busy times, number of physical blocks moved through this device etc. It is important that each disk represents similar disk activity characteristics. A disk busy value of greater than 50% represents bottlenecking.

`sar -w` produces the system swapping and switching activity information. This information includes the average number of physical blocks moved from memory to the swap device, and vice versa. The average number of processes switches per second is also reported. swpot/s should not be too much bigger than 1.

`sar -y` reports on TTY device (terminal) activity.

`sar -c` reports on system call activity.

`sar -q` reports on system queue activity. The run queue lists all processes ready to execute, but waiting for the CPU.

`sar -v` reports on the system table activity. The text, process i-node, record lock, file and file header tables are maintained.

`sar -p` reports on the system paging activity.

`sar -r` reports on the free swap and memory space.

`sar -m` reports on the system message and semaphore activity.

The third main method of monitoring system performance is by the use of program profiling. Profiling can be used to identify those parts of a program that account for significant execution times. The "`-p`" option of `cc` is used to allow the compiler to produce code which counts the number of times each routine is called. The `prof` facility displays profile data for the program in question. For each function, the percentage of time spent executing the function is printed, together with the number of times the function was called and the average number of milliseconds per call. This facility can be used to perhaps identify critical routines that may require optimization.

We have intentionally avoided so far discussing the use of system configuration in fine tuning UNIX Although system configuration, as indeed memory utilization, device utilization, and load manipulation, are all important factors in achieving maximum system performance, configuration often involves modifying parameters that directly affect memory, and device definition as well as their utilization. The configuration tables do not only define the devices a particular system supports, but also allow the user to set the size of various kernel structures. The maximum number of processes per user, the number of entries in the mount table, the number of buffers for block i/o, number of entries in the system open file table are examples of parameters the user can adjust. It is important to note that there are almost as many methods of adjusting kernel parameters, as are vendors supporting UNIX The degree to which a user can tailor the kernel through configuration parameters also varies from one version to another. Once again, your system manager's manual is probably the best guide.

References

[Comer84] Douglas Comer, Operating System Design, The XINU Approach, Prentice-Hall 84.

[Lister81] A M Lister, Fundamentals of Operating Systems, The MACMILLAN PRESS LTD, 81

[Thomson] K. Thompson, Unix Implementation, Bell Laboratories.

[Tunefs] TUNEFS (1M-att) Utility, Pyramid OSx Operating System Manual, 86.

[NCRSuMan] NCR Superuser Reference Manual, Tower 32, July 88.

THE GIGATAPE

HELICAL SCAN 4MM DAT TAPE BACKUP SUBSYSTEM

INTRODUCTION

Data storage requirements are continually on the increase. The amount of data to be stored and saved grows daily. The jobs that computers perform are becoming larger and more complex, while performance and capacity of computers grow with no end in site.

In applications like graphics, CAD, databases or large networks, hundreds of megabytes of data can be accumulated very quickly. Storing and saving this much data used to be a difficult problem in its own right. The expenditures for time, staff, and materials to perform a backup were so costly that backups were either not performed, or performed so infrequently that they lost their real value. With user requirements of backup products including reliability, high storage capacity, unattended use and all at low cost the medium of choice for mid-range users is tape. Rather than reel-to-reel the direction of the future is 4mm cartridge based tape technology. This is due to the reliability, ease of use and storage capacity of these products.

HELICAL SCAN TECHNOLOGY

The breakthrough in storage capacity for tape backup products came with helical scan recording technology. Using helical scan techniques allowed for high recording densities and as a result, high data storage capacity. The original technology gained commercial proliferation in 1978/79 when VHS, BETA and Video 2000 standards were introduced.

The second generation (8 mm technology) has been used to a high degree since about 1985, first in video cameras and later in portable video recorders. The standard was established in 1983. Both methods are still analogue and have a tape wrap angle of 180 degrees or more.

The use of analogue technology in the field of digital data storage has not been very successful despite its use in video for more than 10 years. The missing integrated error correction and the high mechanical stress due to the large tape wrap angle are the limitations of this technology.

4 MM DAT TECHNOLOGY

The third generation of helical scan, R-DAT (Rotating Head, Digital Audio Technology) has now been in use for about 2 years. The basic technology is digital. The cassettes which are the size

of a credit card can be written to repeatedly. For audio use an analogue converter is connected to the drive mechanism. This is not needed however when it is used as a digital storage medium. This is called R-DST (Rotating Head, Digital Storage Tape).
In this third generation of helical scan, the tape wrap angle is reduced to 90 degrees. As a result, up to 4 heads on the drum may be adjusted independently using electronics to separate each short track. With the DST method each head has its own unique polarization to the track azimuth. It is hence possible to overlay the tracks accurately without the danger of "cross-talk", since the magnetic alignment of the adjacent tracks are not picked up by the head. This allows the magnetic surface to be used at maximum density. This was not possible with previous conventional methods.

Another distinct advantage of the 4mm DAT products is due to the low tape wrap angle. With the DST method, tape position information can be read during fast forward and fast rewind movement at 200 times the normal speed. The way this technology operates with a very slowly moving tape passing across a very quickly rotating drum containing the magnetic heads reduces the wear and tear on the tape and reduces the requirement for mechanical parts.

The R-DST technique uses magnetic heads fixed on the sides of a drum. The axis of the drum is inclined from the perpendicular position and the tape passes almost horizontally with very little inclination. This results in the head moving in a spiral form from the bottom to the top of the tape. The resulting track pattern resembles the wrapping pattern found on the grip of a tennis racket. This eliminates all voids on the tape and hence allows for the phenomenally high track density of 73.6 tracks/mm.

Another ingredient of R-DST technology is automatic track finding (ATF). This allows electronic adjustment twice per track hence making this technology dramatically more precise than conventional methods which adjust track position once every several hundred feet. As a result significant mechanical adjustments are not necessary with ATF.

GIGATAPE 4 MM DAT

The Gigatape subsystems employ the R-DST technology in conjunction with specially developed electronic circuitry which optimise the DAT drive in its use in data processing. The R-DST method exceeds the conventional longitudinal or serpentine methods of storage density, in track adjustment and in mechanical reliability. Compared with previous generations of helical scan (VHS and Video-8, respectively), there are fundamental advantages. It is for these reasons that Gigatape has decided to utilise the R-DST method.

Compared with older helical scan methods, R-DST offers these advantages:

- minimal tape wrap
- minimal number of mechanical parts
- fast search with 200 times read/write
- small cassette (3 1/2" form factor is possible)
- precise self adjustment of the track position
- digital recording method
- integrated error correction (error rate < 10^{-15})
- designed for future enhancements.

The cassette used in the Gigatape sub-systems is about the size of a stack of credit cards. It is the DAT standard magnetic tape cassette which has been used in the audio recording field for music storage for several years. Its maximum storage capacity of over 1 gigabyte corresponds to 8 standard 150 Mbyte cartridges.

Functional characteristics available through the use of the Gigatape 4mm DAT technology include:-

* File marks are treated on a logical basis, as opposed to the analogue technique of recording file marks which can result in up to 2 Mbytes of space being consumed per file mark.

* 4mm DAT drives are track addressable allowing a high speed search at 200 times normal recording speed. The average search time for data is 20 seconds, 40 seconds from end to end.

* Capability of handling variable block size data rather than being confined to limited fixed blocks.

* Capability for multiple partioning of the tape, similar to fixed disk storage. This allows for multiple modes or formats to be recorded logically on the same piece of media. As an example, text data in one partition, graphic data in another, video images in a third etc.

* A front panel invocable hardware diagnostics and feature settings capability. Features such as the unit address can be easily set and changed.

* Incorporation of industry standard SCSI, PERTEC and QIC-02 interfaces.

STANDARDS

Standards are necessary with removable media products for data interchange and alternate sources of supply. Data interchange is important for those companies wishing to interchange data between systems of differing architectures.

In recognition of the need to standardise, over 30 major OEMs, including Apple, Cipher Data, Sharp, Hewlett Packard, Wangtek and Gigatape, are working to develop a 4mm DAT standard. From this group at least 4 companies have now committed to supply 4mm DAT drives.

FURTHER INFORMATION

Further information on the Gigatape 4mm DAT product can be obtained from:

LYNX Technologies Pty Limited,
P.O. BOX 567,
Woollahra, NSW,
Telephone (02) 32 5761
Facsimile (02) 327 8357

The Computing Engines of the 1990s

Ross Bott
Pyramid Technology
August 1989

Scenario I: Big Computers

Premise:

An environment for large centralized mainframes servicing"dumb" terminals is sensible, efficient and cost effective.

Players:

IBM, Amdahl, Unisys (2200), Fujitsu, NAS, NEC

Advantages:

1. Centralized. Easy to administer, maintain, and control the data environment.
2. Monolithic databases can be provided to larger user communities.
3. Scenario firmly established and in place. Software applications are in production. Users cannot afford to switch.

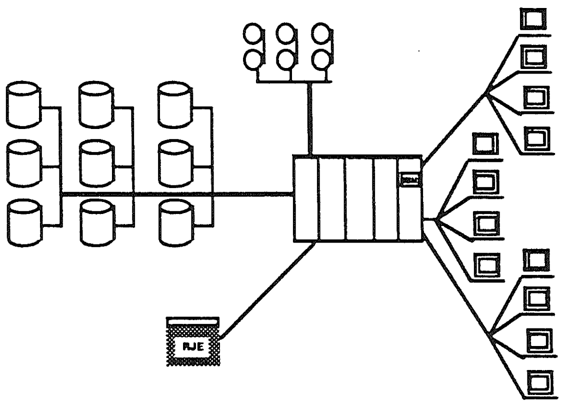
Trends:

1. Bigger and faster (e.g., IBM 3090 S series, Summit)
2. CPU technology will become ever more exotic.
3. Concentration on I/O.
4. Support for existing software is critical.

Disadvantages:

1. Ignores microprocessor revolution. Will never again be price/performance competitive.

Traditional Computing Solution



Scenario II: Hetereogeneous Multi-Level

Premise:

Computing enviornment will consist of specialized computers, each serving a single function and all networked to provide a transparent distributed computing capability.

Players:

Servers: DEC, Pyramid, Convex , etc. Workstations: Sun, Apollo, Apple, etc.

Advantages:

1. Each computer type can be designed for highest performance and most cost effective solution for the role it plays.
2.Burden of complexity is moved to systems and networking software, which needs to be written only once.
3. RISC approach towards systems design.

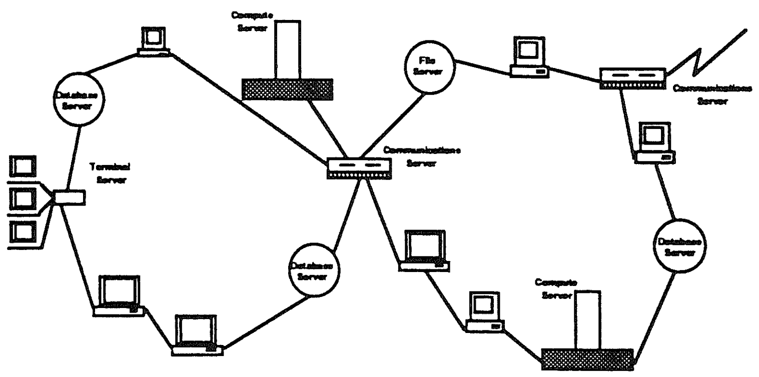
Trends:

1. Specialized, fast compute servers (100+ MIPS) with network interfaces only
2. Fast, high capacity, fault resistant database and file servers.
3. Performance range of workstations: Macintoshes, PCs, Suns

Disadvantages:

1. Networking software issues must still be solved.
2. Customer must be comfortable with the complexities of implementing the network.

Workstation / Server Network



Scenario III: Servers and Intelligent Terminals

Premise:

Computing enviornment will consist of desktop workstations providing only an intelligent, friendly user interface, and high performance servers providing all computing resources for the applications.

Players:

DEC, Pyramid, Convex , etc. Apple, Visual, NCD, etc.

Advantages:

1.Cost effective: workstation expense is concentrated on those aspects which must be located at the user, i.e., just the user interface.

2.All other computing resources can be shared.

Trends:

1. Cheaper, but more powerful X-display stations and Macintoshes.
2. Standardization of user interface technology and specifications.
3. Concentration of software research in distributed applications.

Disadvantages:

1. Requires very fast networking for some applications; cost of fast network interface conflicts with the low cost of the intelligent terminal.

Scenario IV: All Workstations

Premise:

Each desktop will have sufficient resources to handle all requirements of that user, from PCs and workstations to desktop Crays and desktop database engines. Workstations will be networked primarily for information exchange.

Players:

Stellar, Ardent, Sun

Advantages:

1. Never dependent on availability (reliability or cycles) of resources elsewhere on network.
2. Takes advantage of ever increasing power of microprocessors.
3. Performance and reliability of network is much less of an issue.

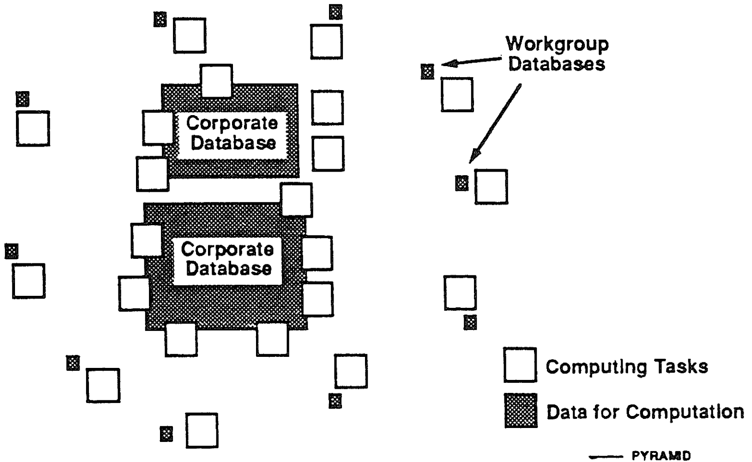
Trends:

1. 50 MIPS, 100 Mflops, 5-10 GB workstations
2. All PC corporate computing environments

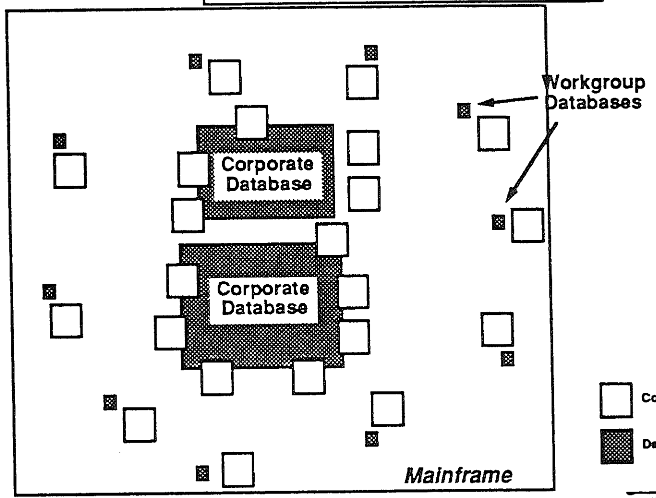
Disadvantages:

1. Database sharing a problem.
2. Poor use of MIPS (humans are slow); CISC approach to network system design.

Computational Structure: Corporate Information Environment



Traditional Mainframe Solution



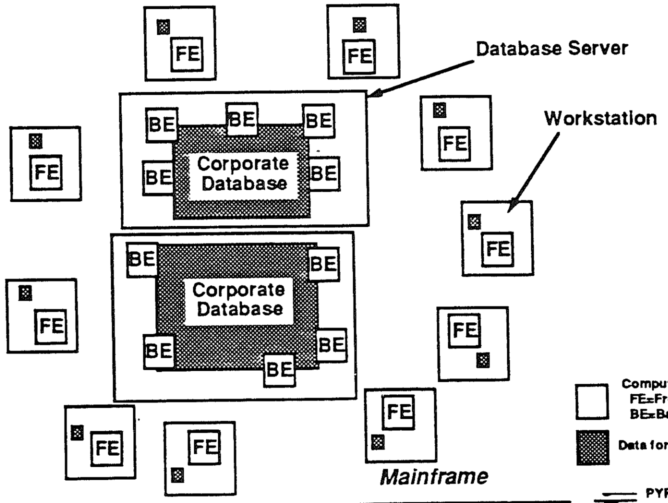
□ Computing Tasks

▨ Data for Computation

PYRAMID TECHNOLOGY ©1989

Page 9

Workstation/Server Solution



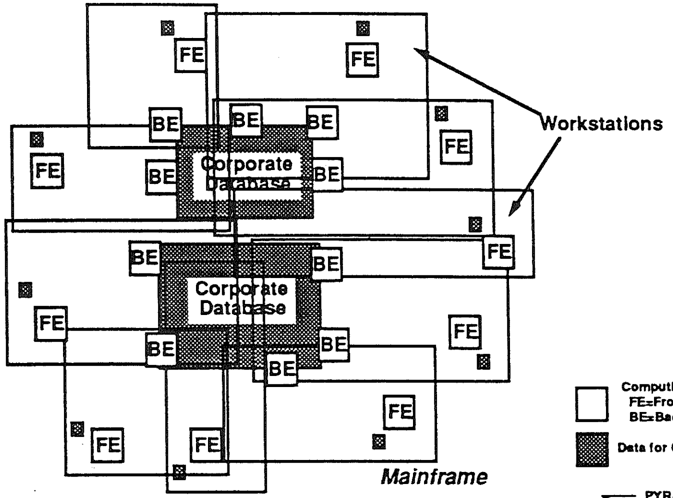
□ Computing Tasks
FE=Front end
BE=Back end

▨ Data for Computation

PYRAMID TECHNOLOGY ©1989

Page 10

All Workstation Solution



□ Computing Tasks
FE=Front end
BE=Back end

▨ Data for Computation

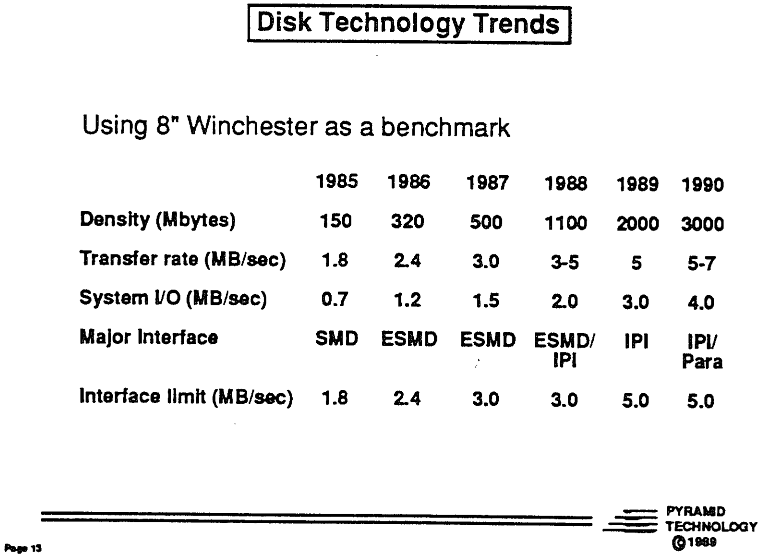
PYRAMID TECHNOLOGY ©1989

Page 11

File Server Evolution

PYRAMID TECHNOLOGY ©1989

Page 12



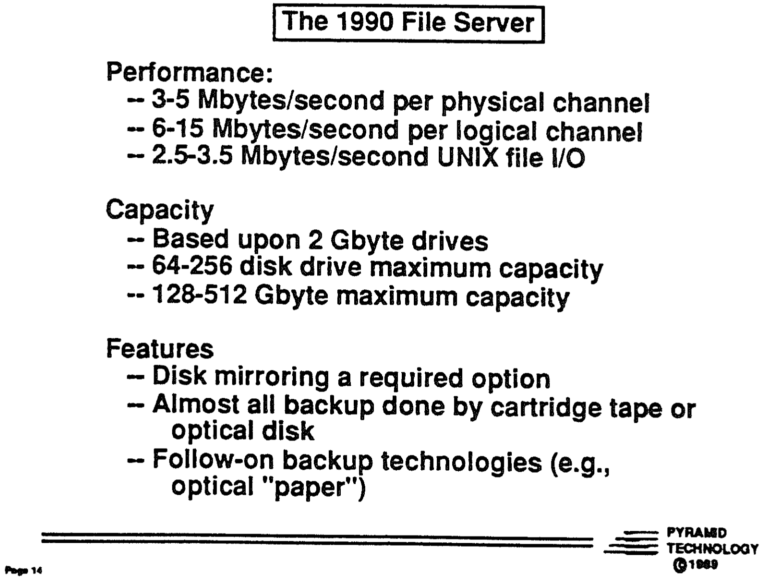
Disk Technology Trends

Using 8" Winchester as a benchmark

	1985	1986	1987	1988	1989	1990
Density (Mbytes)	150	320	500	1100	2000	3000
Transfer rate (MB/sec)	1.8	2.4	3.0	3-5	5	5-7
System I/O (MB/sec)	0.7	1.2	1.5	2.0	3.0	4.0
Major Interface	SMD	ESMD	ESMD	ESMD/ IPI	IPI	IPI/ Para
Interface limit (MB/sec)	1.8	2.4	3.0	3.0	5.0	5.0

PYRAMID TECHNOLOGY ©1989

Page 13



The 1990 File Server

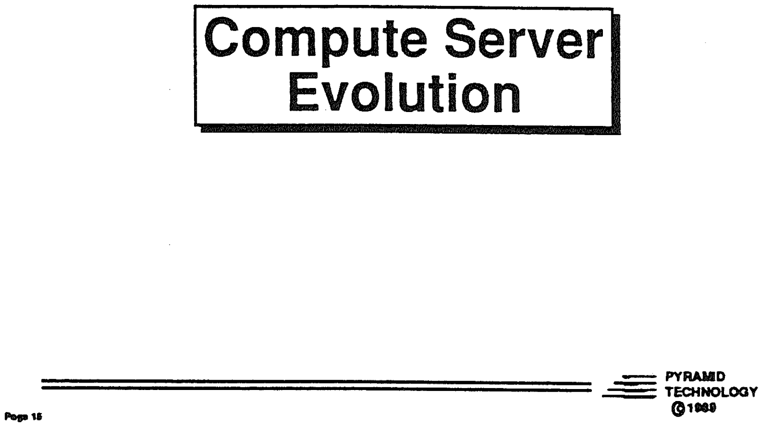
Performance:
- -- 3-5 Mbytes/second per physical channel
- -- 6-15 Mbytes/second per logical channel
- -- 2.5-3.5 Mbytes/second UNIX file I/O

Capacity
- -- Based upon 2 Gbyte drives
- -- 64-256 disk drive maximum capacity
- -- 128-512 Gbyte maximum capacity

Features
- -- Disk mirroring a required option
- -- Almost all backup done by cartridge tape or optical disk
- -- Follow-on backup technologies (e.g., optical "paper")

PYRAMID TECHNOLOGY ©1989

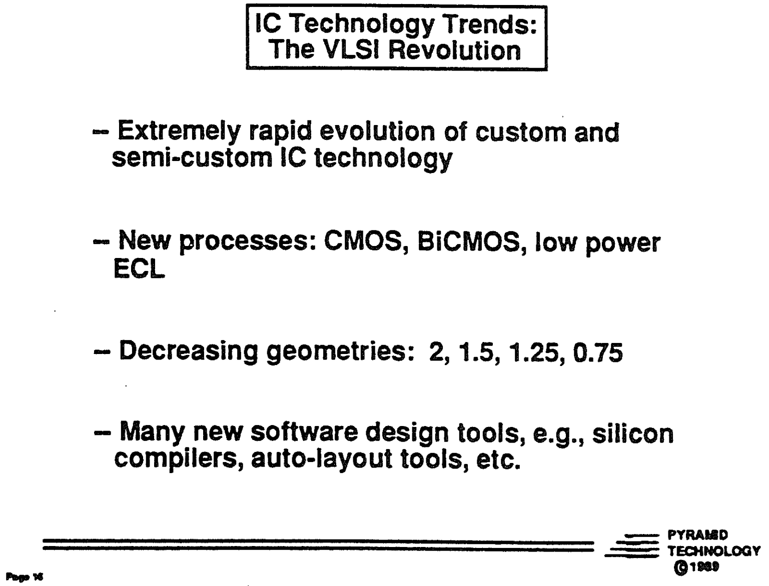
Page 14



Compute Server Evolution

PYRAMID TECHNOLOGY ©1989

Page 15

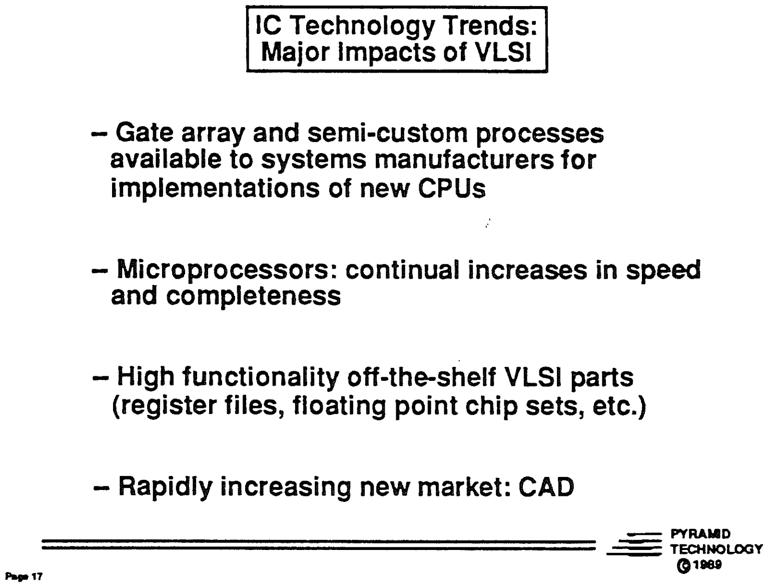


IC Technology Trends: The VLSI Revolution

- -- Extremely rapid evolution of custom and semi-custom IC technology
- -- New processes: CMOS, BiCMOS, low power ECL
- -- Decreasing geometries: 2, 1.5, 1.25, 0.75
- -- Many new software design tools, e.g., silicon compilers, auto-layout tools, etc.

PYRAMID TECHNOLOGY ©1989

Page 16

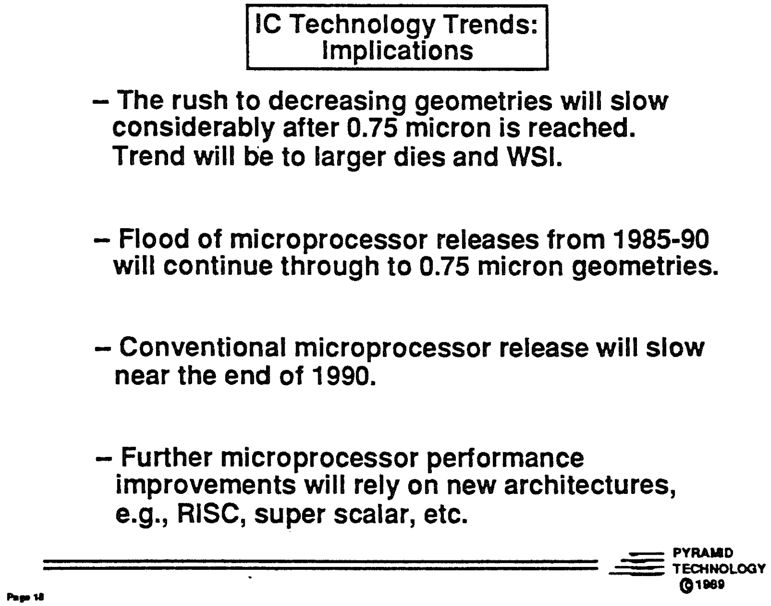


IC Technology Trends: Major Impacts of VLSI

- Gate array and semi-custom processes available to systems manufacturers for implementations of new CPUs
- Microprocessors: continual increases in speed and completeness
- High functionality off-the-shelf VLSI parts (register files, floating point chip sets, etc.)
- Rapidly increasing new market: CAD

PYRAMID TECHNOLOGY ©1989

Page 17

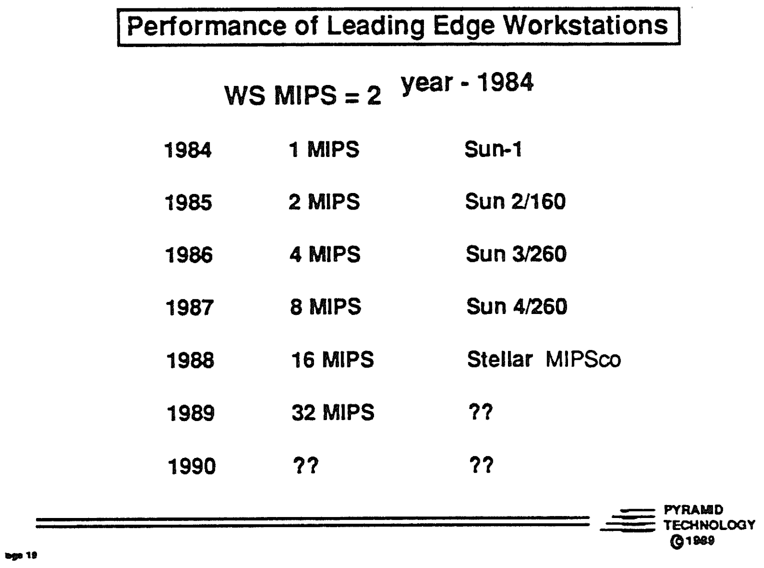


IC Technology Trends: Implications

- The rush to decreasing geometries will slow considerably after 0.75 micron is reached. Trend will be to larger dies and WSI.
- Flood of microprocessor releases from 1985-90 will continue through to 0.75 micron geometries.
- Conventional microprocessor release will slow near the end of 1990.
- Further microprocessor performance improvements will rely on new architectures, e.g., RISC, super scalar, etc.

PYRAMID TECHNOLOGY ©1989

Page 18



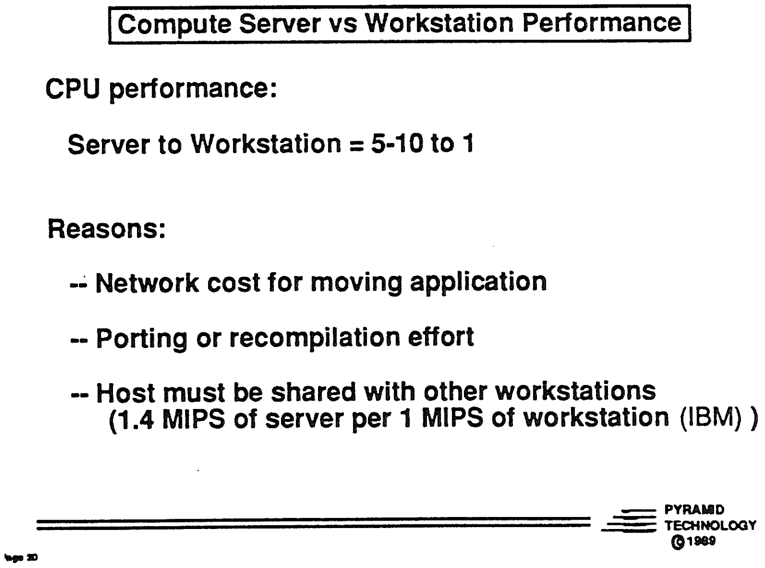
Performance of Leading Edge Workstations

WS MIPS = $2^{\text{year - 1984}}$

1984	1 MIPS	Sun-1
1985	2 MIPS	Sun 2/160
1986	4 MIPS	Sun 3/260
1987	8 MIPS	Sun 4/260
1988	16 MIPS	Stellar MIPSco
1989	32 MIPS	??
1990	??	??

PYRAMID TECHNOLOGY ©1989

Page 19



Compute Server vs Workstation Performance

CPU performance:

Server to Workstation = 5-10 to 1

Reasons:

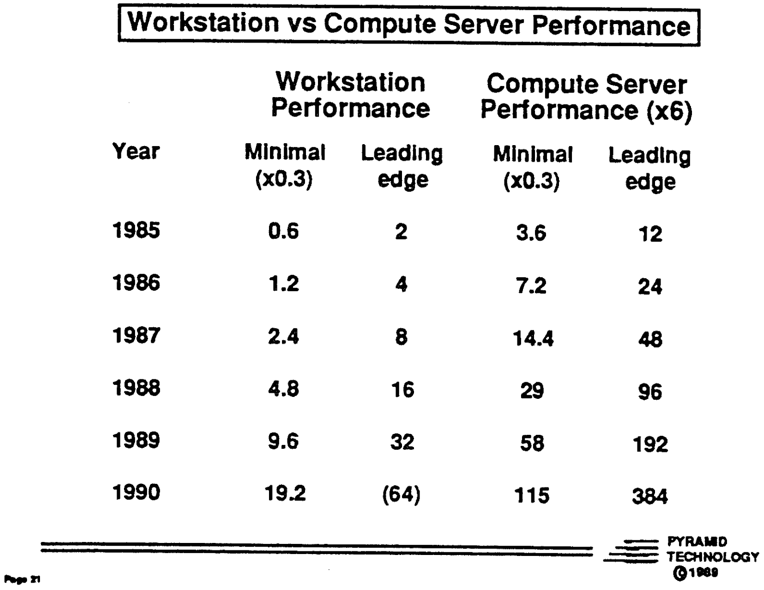
-- Network cost for moving application

-- Porting or recompilation effort

-- Host must be shared with other workstations (1.4 MIPS of server per 1 MIPS of workstation (IBM))

PYRAMID TECHNOLOGY ©1989

Page 20

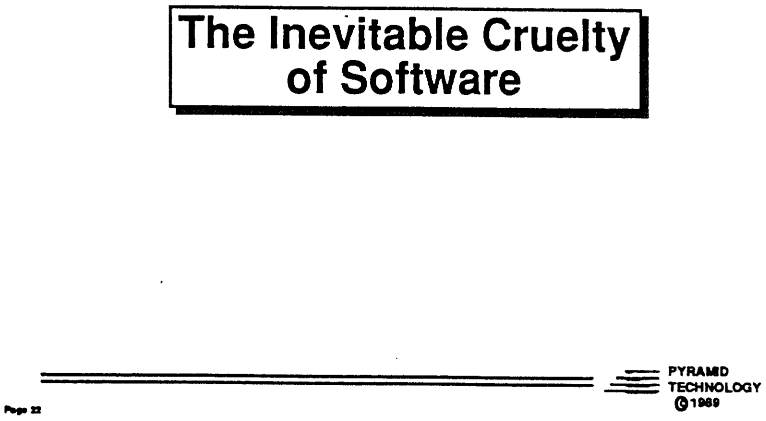


Workstation vs Compute Server Performance

	Workstation Performance		Compute Server Performance (x6)	
Year	Minimal (x0.3)	Leading edge	Minimal (x0.3)	Leading edge
1985	0.6	2	3.6	12
1986	1.2	4	7.2	24
1987	2.4	8	14.4	48
1988	4.8	16	29	96
1989	9.6	32	58	192
1990	19.2	(64)	115	384

PYRAMID TECHNOLOGY ©1989

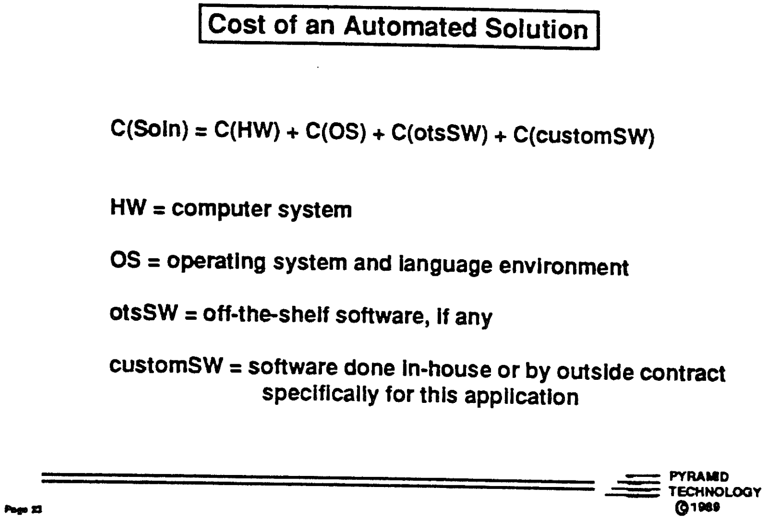
Page 21



The Inevitable Cruelty of Software

PYRAMID TECHNOLOGY ©1989

Page 22



Cost of an Automated Solution

C(Soln) = C(HW) + C(OS) + C(otsSW) + C(customSW)

HW = computer system

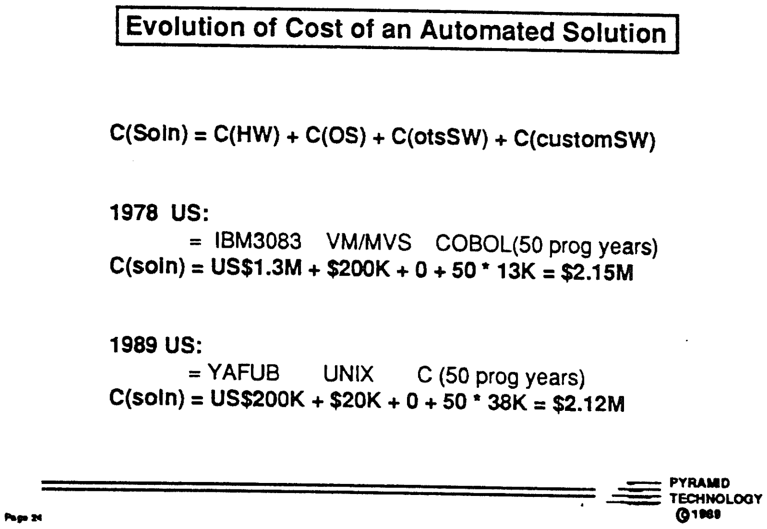
OS = operating system and language environment

otsSW = off-the-shelf software, if any

customSW = software done in-house or by outside contract specifically for this application

PYRAMID TECHNOLOGY ©1989

Page 23



Evolution of Cost of an Automated Solution

C(Soln) = C(HW) + C(OS) + C(otsSW) + C(customSW)

1978 US:
= IBM3083 VM/MVS COBOL(50 prog years)
C(soln) = US$1.3M + $200K + 0 + 50 * 13K = $2.15M

1989 US:
= YAFUB UNIX C (50 prog years)
C(soln) = US$200K + $20K + 0 + 50 * 38K = $2.12M

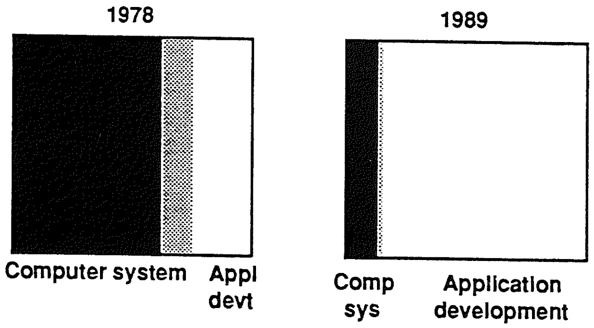
PYRAMID TECHNOLOGY ©1989

Page 24

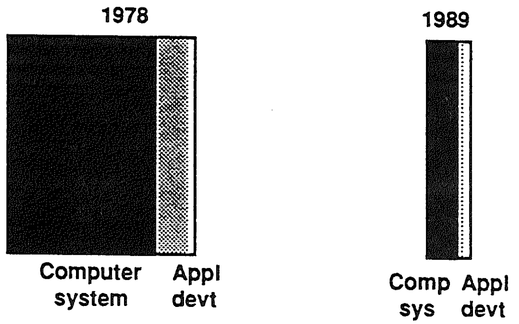
The Cost of a Good Software Engineer

	1978	1989
Silicon Valley	US$18,000	US$60,000
Boston Rt 128	US$17,000	US$50,000
US average	US$13,000	US$38,000
Australia average	US$10,000	US$35,000
India average	US$300	US$1000

Redistribution of Cost of Automation: United States



Redistribution of Cost of Automation: India



Software vs Hardware Complexity

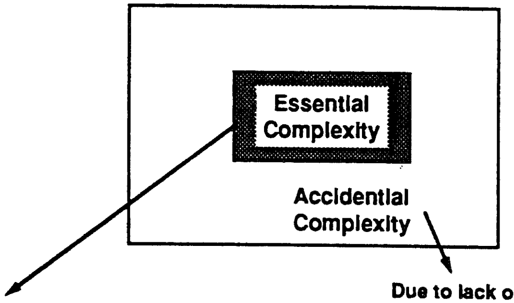
Hardware design time:

Intel 80286	100K transistors	2 years
Intel 80386:	300K transistors	10 months
Intel 80486:	1000K transistors	3 weeks
Intel 80586: ...		

Software design time:

COBOL (1978)	200K lines	50 programmer years
C (1989)	200K lines	50 programmer years

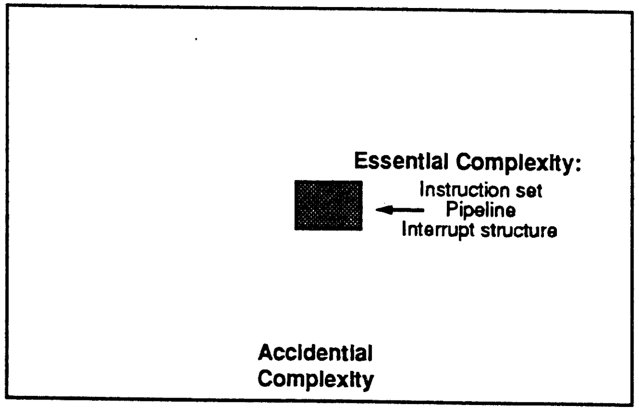
Design Complexity



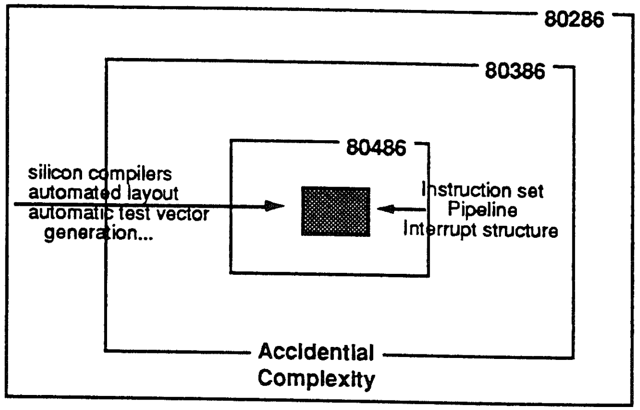
Due to fundamental structure of
- data entities
- relationships between data entities
- control flow
- algorithms
- interface to user

Due to lack of
- specification quality
- expression level of programming tools
- power / intelligence of tools
- generation and coverage of testing tools
- version/configuration control

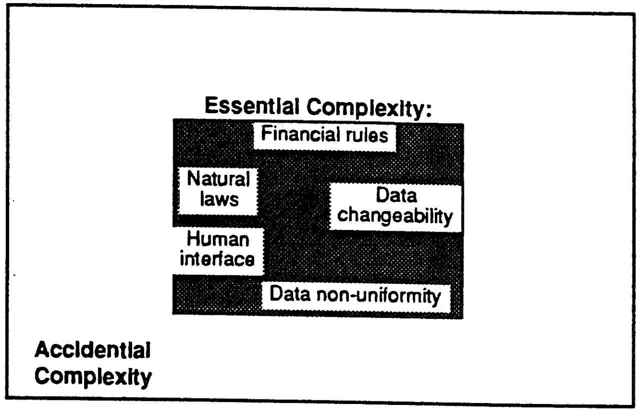
Complexity of a Hardware Design



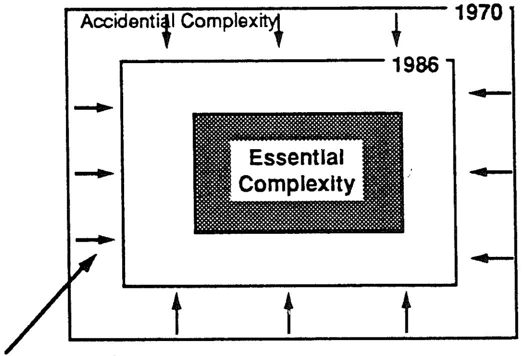
Hardware Design Complexity Evolution



Complexity of a Software Design



Software Complexity Evolution: 1970-1986

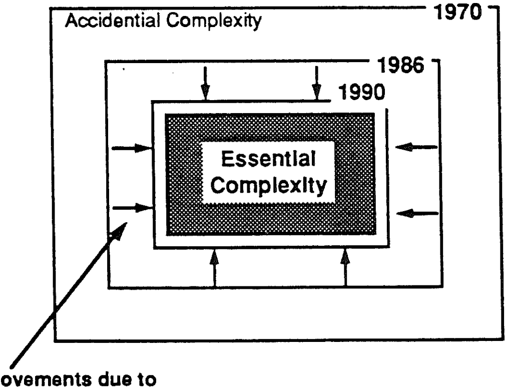


Improvements due to
(1) Machine language --> Assembler --> 3GL
(2) Network / hierarchical DBs (e.g. IMS) --> Relational DBs
(3) Machine level debuggers --> Symbolic debuggers
(4) 3270 --> Bitmapped / windowed interface
(5) Snapshot control --> Version control

PYRAMID TECHNOLOGY ©1989

Page 33

Software Complexity Evolution: 1986-1990



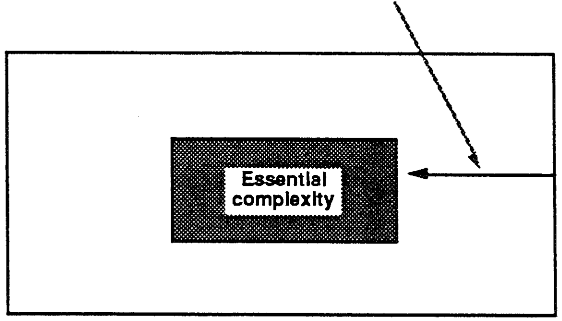
Improvements due to
(1) 3GL --> 4GL
(2) Object-based programming
(3) CASE

PYRAMID TECHNOLOGY ©1989

Page 34

Eliminating Accidential Complexity in Software Development

1989 US:
= YAFUB UNIX C (50 prog years)
C(soln) = US$200K + $20K + 0 + 50 * 38K = $2.12M



PYRAMID TECHNOLOGY ©1989

Page 35

Types of Accidential Complexity Weapons

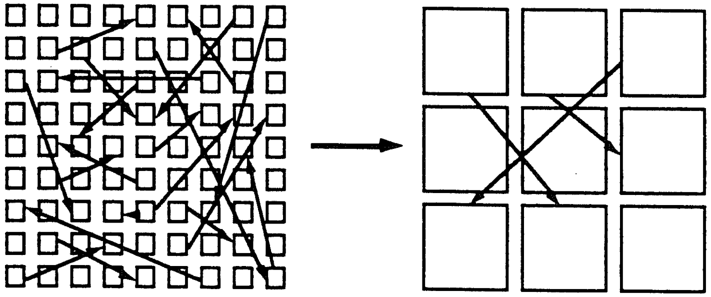
Development process tools:
(1) Graphic / control flow design tools
(2) Compilers
(3) Smart debuggers
(4) Version control and automatic configuration
(5) Automatic test generation

Representational weapons:
(1) High-level languages
(2) Modular programming
(3) Object oriented programming
(4) Interpretive interface

PYRAMID TECHNOLOGY ©1989

Page 36

Reduction of Complexity Through Representational Size

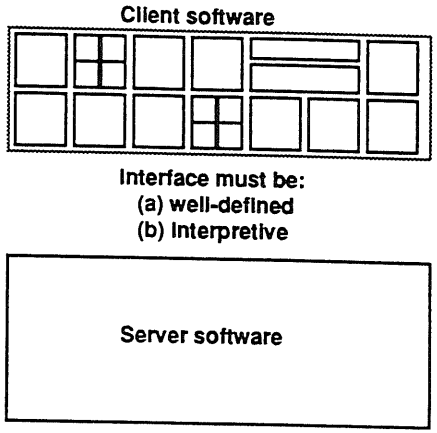


Examples: HLL, OOP, Interpretive Interfaces

PYRAMID TECHNOLOGY ©1989

Page 37

Client-Server Software: Caveats



Client software

Interface must be:
(a) well-defined
(b) interpretive

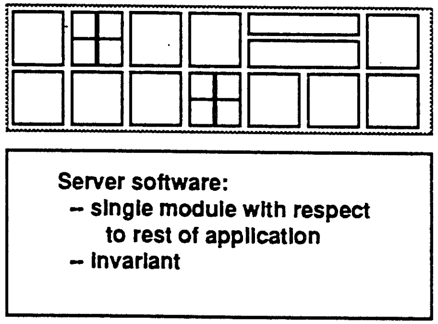
Server software

PYRAMID TECHNOLOGY ©1989

Page 38

Reduction of Complexity: Client-Server Software as an Instance of Representational Size

Client software:
- multiple pieces
- constructed new for each application

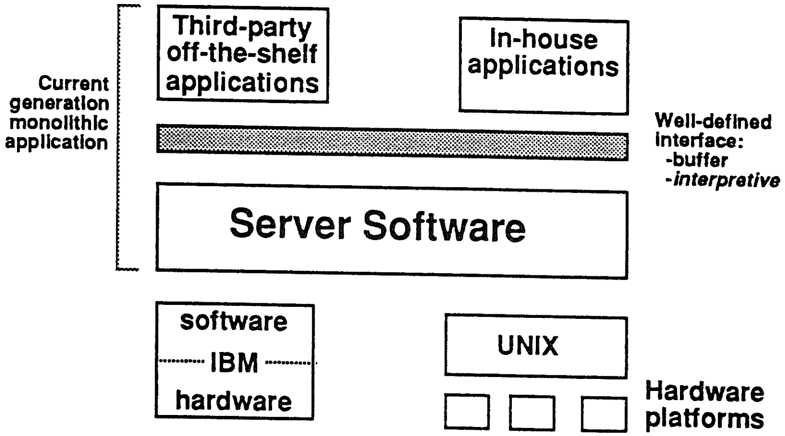


Server software:
- single module with respect to rest of application
- invariant

PYRAMID TECHNOLOGY ©1989

Page 39

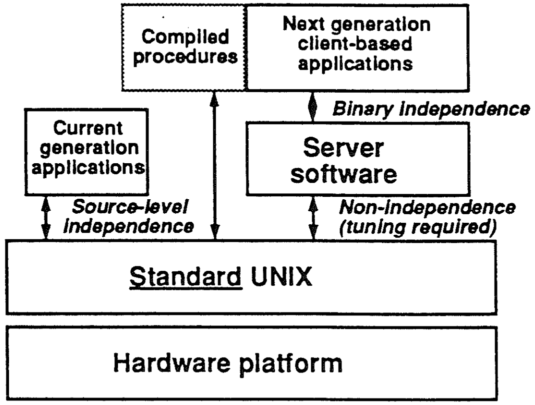
Client - Server Model:
Buffering Applications From Changes



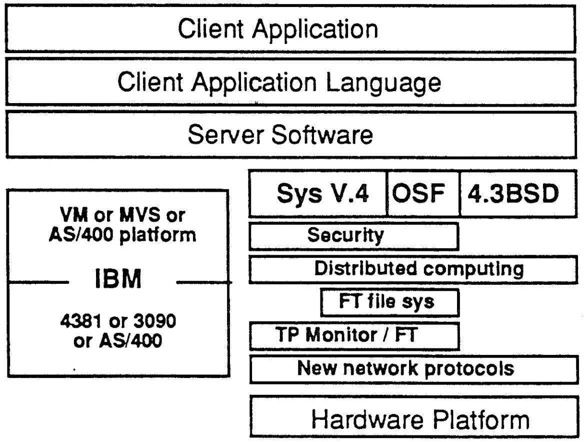
PYRAMID TECHNOLOGY ©1989

Page 40

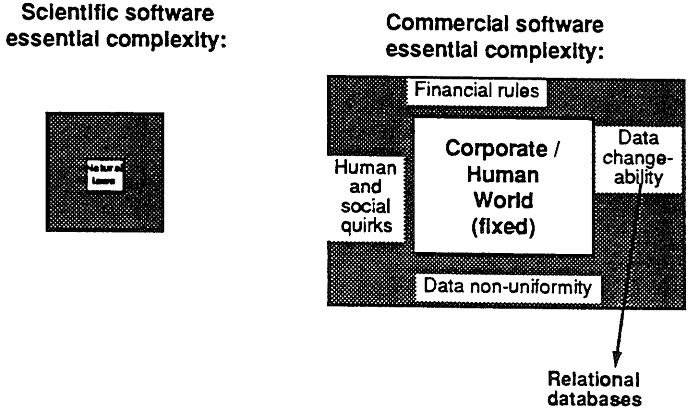
Binary Independence vs Source Code Independence


Compiled procedures
Next generation client-based applications
Binary independence
Current generation applications
Server software
Source-level independence
Non-independence (tuning required)
Standard UNIX
Hardware platform

Client Application: Transparence to Operating System Evolution / Conflicts



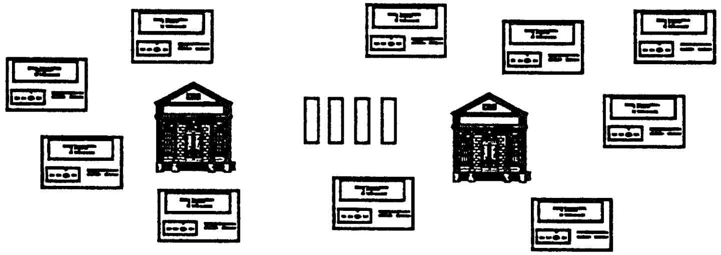
Commercial vs Scientific Design Complexity


Scientific software essential complexity:
Natural laws
Commercial software essential complexity:
Financial rules
Human and social quirks
Corporate / Human World (fixed)
Data change-ability
Data non-uniformity
Relational databases

Database Server Evolution

Typical Customer Requirements

Banking / Automatic Tellers



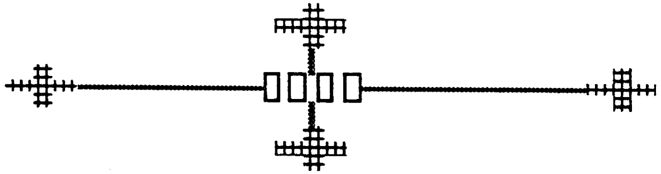
800 ATMs
100 Trans/Hour -> 80,000 Trans/Hour
1,280,000 Trans/16 hour day

Transactions entered in corporate bank data
during 8 hour night shift
-> 180,000 Transactions/Hour -> 44 TPS

Typical Customer Requirements

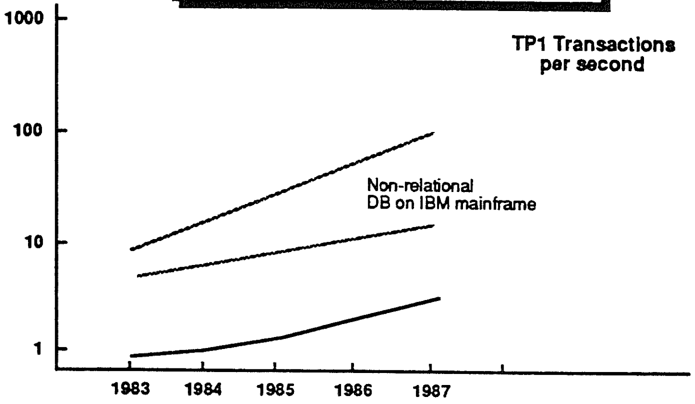
Hospital Patient Database

Four Hospitals Close Enough for Centralized Computer Facility
(20,000,000 patient records)



	Today	Future
Applications	Patient Tracking, Billing	Patient Trends, Disease Trends
Average TPS	2.1 TPS (180,000 TPD)	50 TPS
Peak TPS	63 TPS	500 TPS

Relational Database Performance on Traditional Single Processor Through 1987



1987 Relational Database Performance

Traditional 5 MIPS single microprocessor as RDBMS server

Ideal Conditions

- 1-5 users
- 10-25 TPS
- 5 MIPS CPU + typical transaction -> 20 TPS
  = excellent!

Some Database Contention

- 100+ users
- 3-8 TPS
- 5 MIPS CPU + typical transaction -> 4 TPS
  = excellent?

Where does the time go?

Micro-level answer:

Simple query
1 join + 1 select

Micro-Level Timings

Join

CPU instruction	# Instances	# Cycles
move	4	10
conditional branch	2	5
compare	2	4

Total cycles = 58 per byte

Select

CPU instruction	# Instances	# Cycles
move	2	10
arithmetic	2	5
compare	2	4
branch	3	5

Total cycles = 53 per byte

Typical Simple Query

Assume 5 MIPS traditional processor, 40ns cycle time

Join (58 cycles) + Select (53 cycles) = 111 cycles

At 40ns / cycle, CPU time is 4.44 usec/byte

CPU time to process 8K logical page is 35.5 msec

Typical Simple Query

Conclusions:

At most 28 logical disk block reads/second generated during processing of the transaction

40% system CPU overhead --> 17 blocks/sec maximum

UNIX buffer hit rate --> ~10 blocks/sec maximum

Single traditional CPU is CPU-bound on relational databases

Solving the CPU Bottleneck for Relational Databases

Alternatives:

(1) Use a multiprocessor approach

(2) Use faster single processors

(3) Combination of (1) and (2)

Page 33 PYRAMID TECHNOLOGY ©1989

Solving the CPU Bottleneck for Relational Databases

Issues

(1) Are multiprocessors good for relational databases?

(2) If so, are their any constraints on this approach?

Page 34 PYRAMID TECHNOLOGY ©1989

Are Multiprocessors Necessary For Relational Databases?

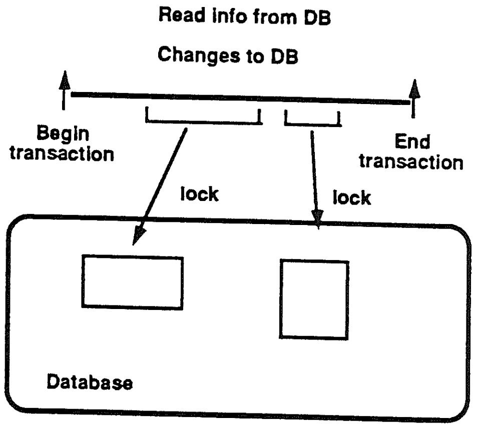
Relational databases are CPU-bound

- one 5 MIPS processor will generate ~10 I/O's a second
- 50+ I/O's possible / disk spindle
- 5 spindle database can consume 125 MIPS
- 20 spindle database can consume 500 MIPS

Design Rule #1: Multiprocessors are necessary for medium to large databases

Page 35 PYRAMID TECHNOLOGY ©1989

Transaction Structure


Read info from DB

Changes to DB

Begin transaction

End transaction

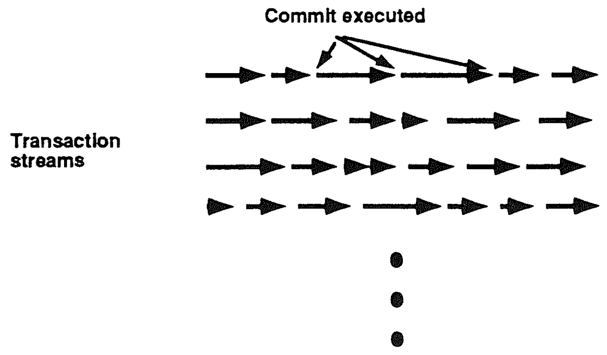
lock

lock

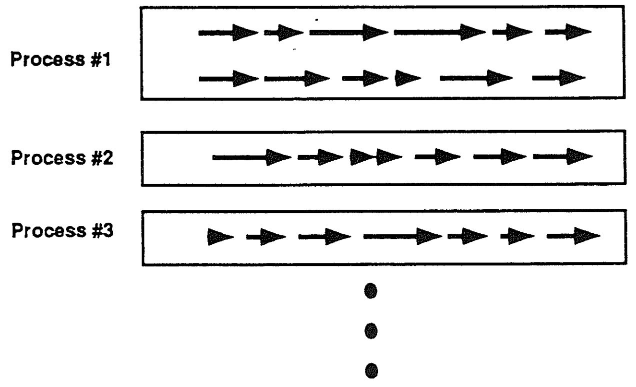
Database

Page 35 PYRAMID TECHNOLOGY ©1989

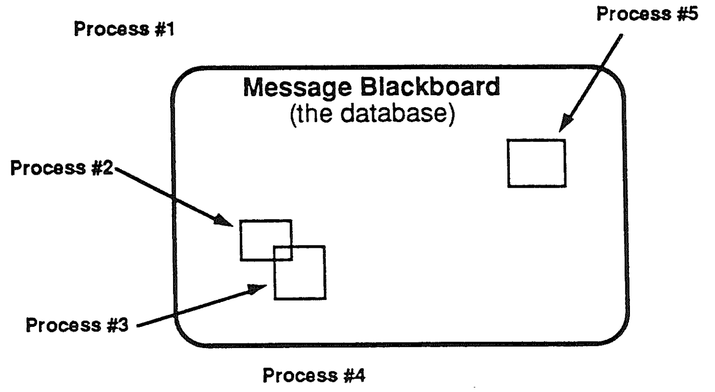
Database Transaction Structure



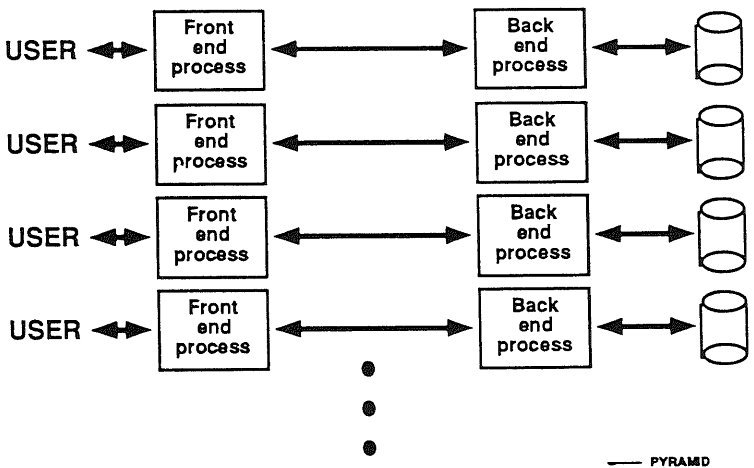
Database *Process* Structure



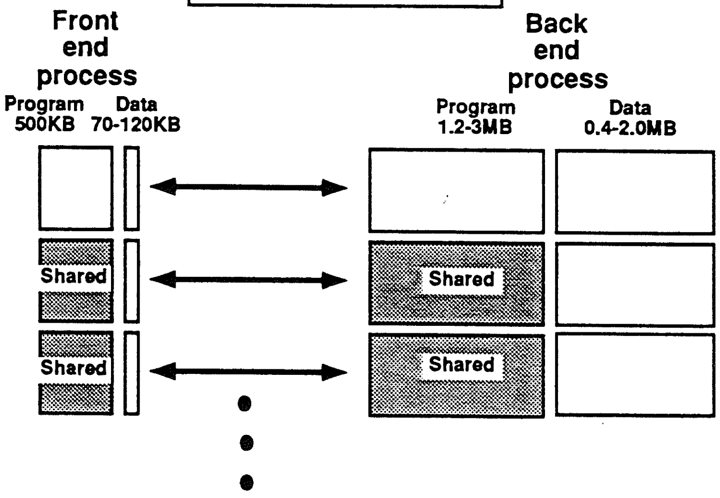
Database Processing As "Blackboard-Based" Parallel Processing



Process Structure for Database System



Process Structure Sizes



PYRAMID TECHNOLOGY ©1989

Design Rule #2: Allow For Very Large Main Memory

Memory needed (MB) = 0.5 + 0.1*N + 2.0 + 0.5*N

Front end   Back end

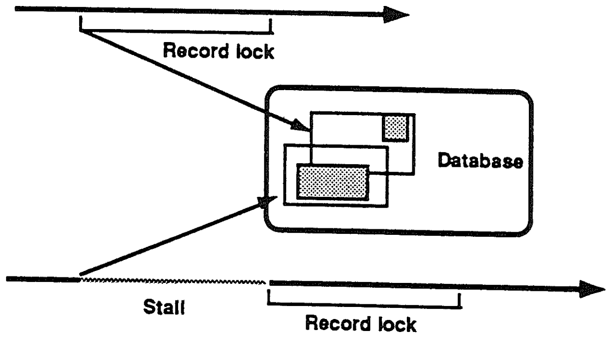
(N = number of simultaneous DB users)

Memory needed (100 users) = 62.5 MB

Memory needed (250 users) = 152.5 MB

PYRAMID TECHNOLOGY ©1989

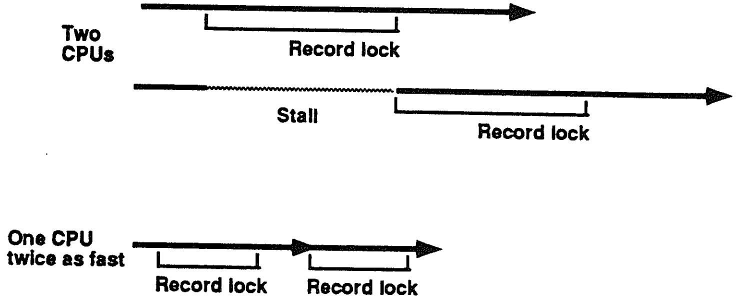
Conflict Between Transactions



Percentage stall time: 2-6%

PYRAMID TECHNOLOGY ©1989

Transaction Conflict: Single vs Multiprocessors



PYRAMID TECHNOLOGY ©1989

Amdahl's Law

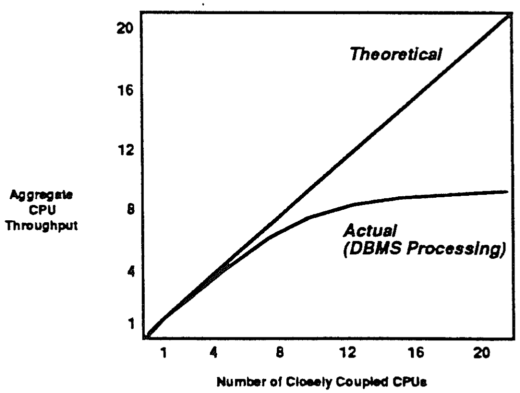
$$\text{Single CPU Multiplier} = \frac{1}{SF + [(1 - SF)] / \#CPUs}$$

SF = Serial Factor; caused by

- Lock Management (stall time)
- Logging
- Operating System

Serial manipulation of key data structures

Amdahl's Law: MP Effectiveness for Databases



Design Rule #3: Use As Few Processors As Possible

SF = 4% ==> very little benefit beyond 16 processors

Design Rule #4: Use Fastest Processors Possible

RISC processors offer most performance for given cycle time

General Commercial Applications: Requirements for Big Systems

Example:

Customer: International construction company

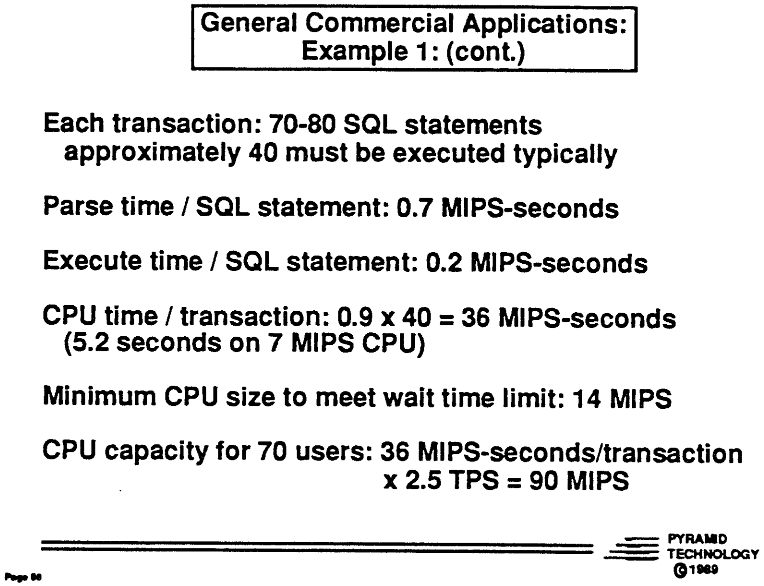
Application: General ledger, accounting

Database: Well-known relational database

Simultaneous users: 60-70 maximum

Transaction rate: 2.5 TPS overall maximum

User delay time: 2.5 seconds maximum



General Commercial Applications: Example 1: (cont.)

Each transaction: 70-80 SQL statements
approximately 40 must be executed typically

Parse time / SQL statement: 0.7 MIPS-seconds

Execute time / SQL statement: 0.2 MIPS-seconds

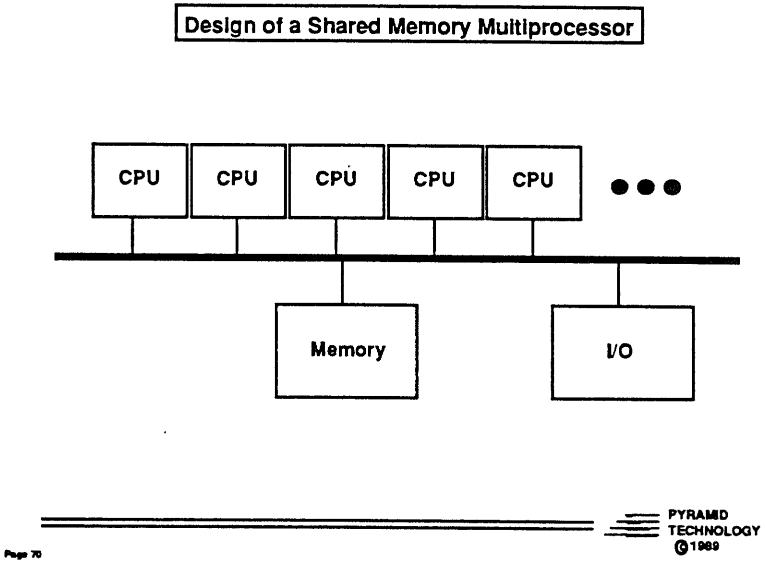
CPU time / transaction: 0.9 x 40 = 36 MIPS-seconds
(5.2 seconds on 7 MIPS CPU)

Minimum CPU size to meet wait time limit: 14 MIPS

CPU capacity for 70 users: 36 MIPS-seconds/transaction
x 2.5 TPS = 90 MIPS

PYRAMID TECHNOLOGY ©1989

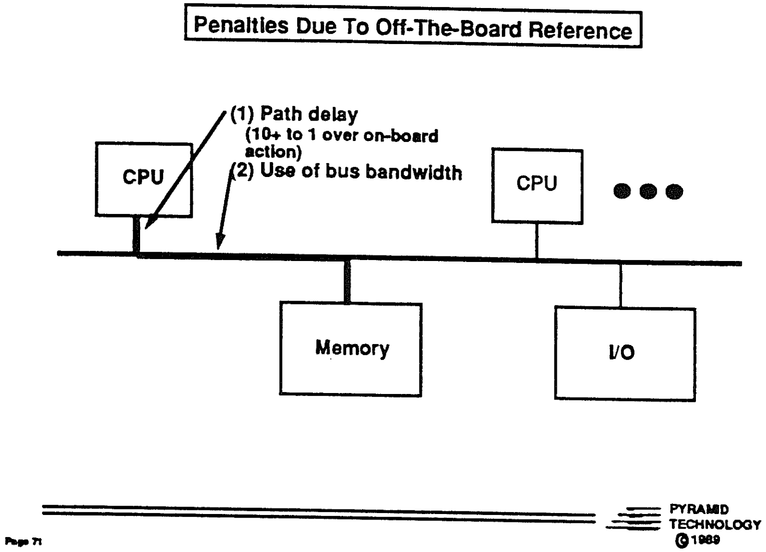
Page 69



Design of a Shared Memory Multiprocessor

PYRAMID TECHNOLOGY ©1989

Page 70

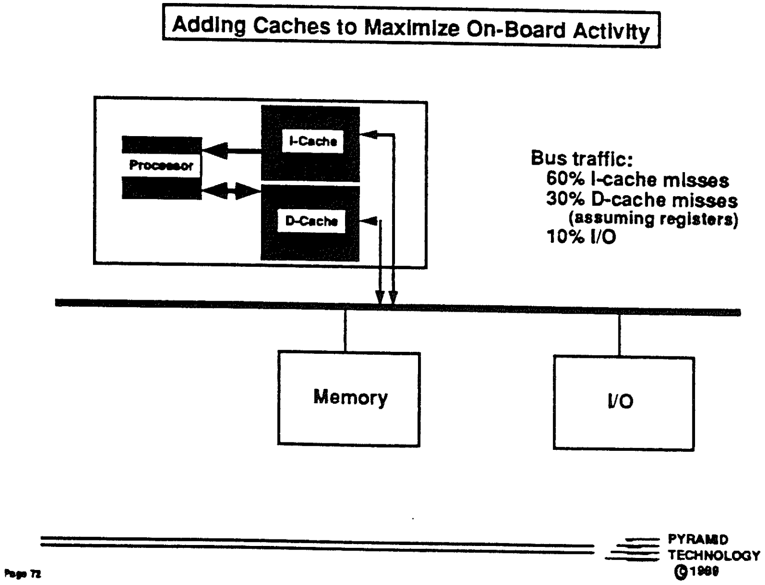


Penalties Due To Off-The-Board Reference

(1) Path delay
(10+ to 1 over on-board action)
(2) Use of bus bandwidth

PYRAMID TECHNOLOGY ©1989

Page 71



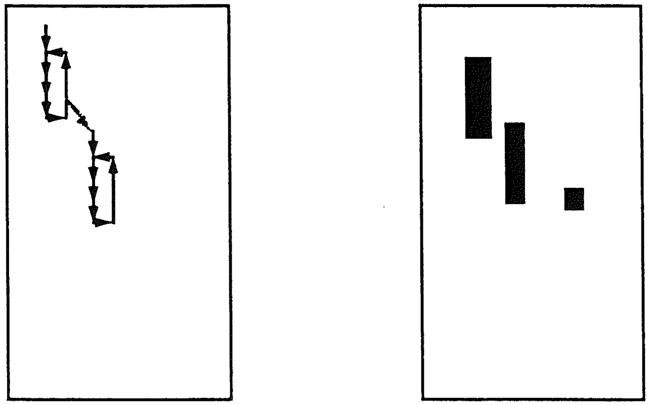
Adding Caches to Maximize On-Board Activity

Bus traffic:
60% I-cache misses
30% D-cache misses
(assuming registers)
10% I/O

PYRAMID TECHNOLOGY ©1989

Page 72

Working Set Size: Scientific Program Behavior



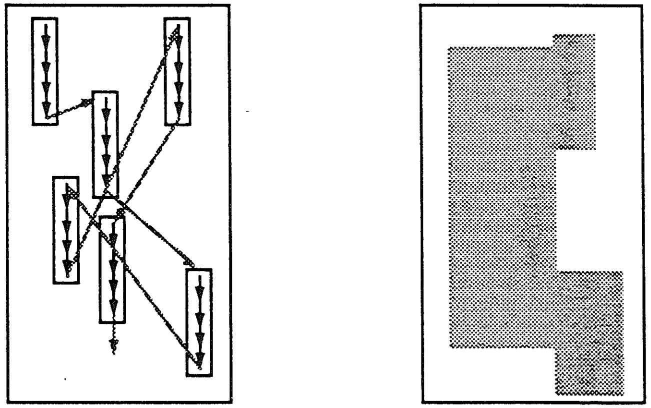
Tight loops; many iterations

10-30 Kbytes / second

PYRAMID TECHNOLOGY ©1989

Page 73

Working Set Size: Database Server Program Behavior



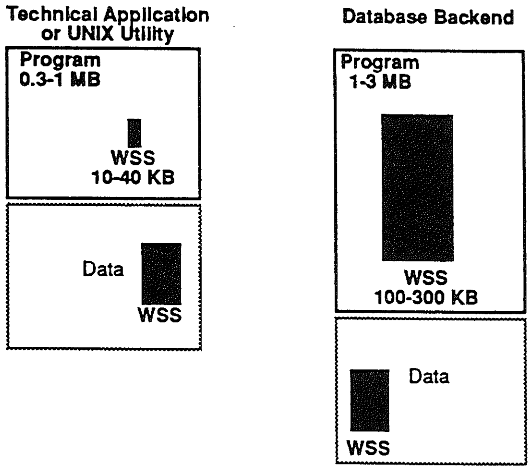
Tight loops; many iterations

100-300 Kbytes / second

PYRAMID TECHNOLOGY ©1989

Page 74

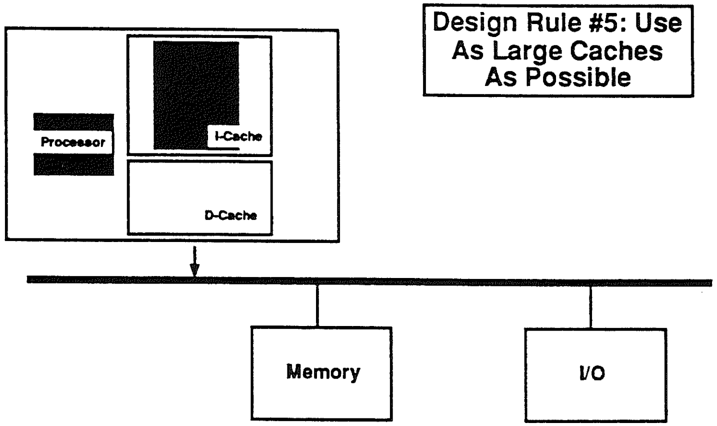
Program Working Set Size (WSS)



PYRAMID TECHNOLOGY ©1989

Page 75

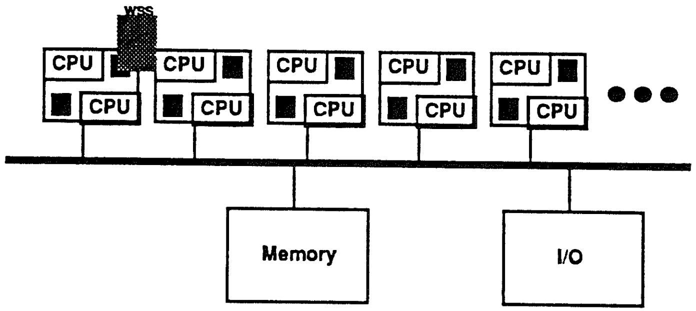
Design Rule #5: Use As Large Caches As Possible



PYRAMID TECHNOLOGY ©1989

Page 76

Design Rule #6: Many Small Inexpensive CPUs Are Inefficient For Databases

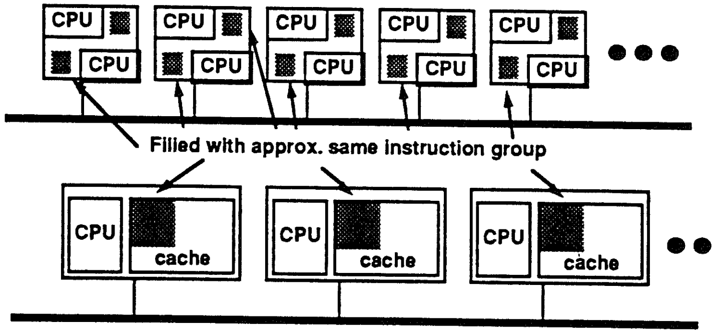


(a) WSS of database can overwhelm cache
(b) Many CPUs increase latching conflicts

Page 77

PYRAMID TECHNOLOGY ©1989

Many Small Caches ==> Inefficient Cache Utilization

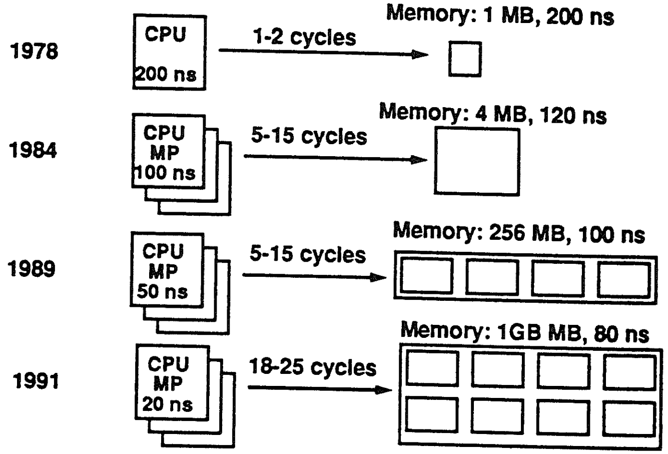


(c) Larger CPU/caches generates more efficient usage of cache chips

Page 78

PYRAMID TECHNOLOGY ©1989

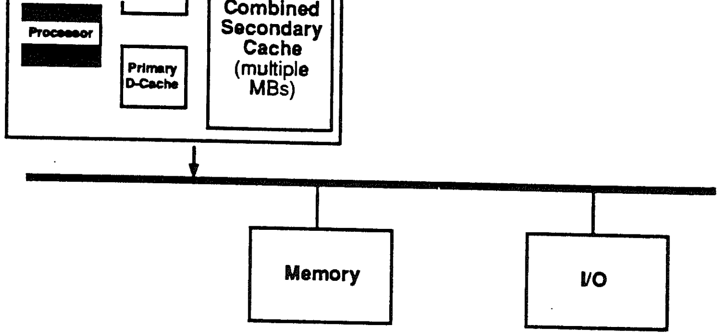
Increasing Distance Between Processor And DRAM Memory Speeds



Page 79

PYRAMID TECHNOLOGY ©1989

Design Rule #7: Use Large Secondary Caches

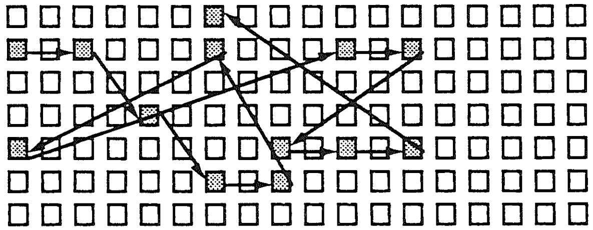


Page 80

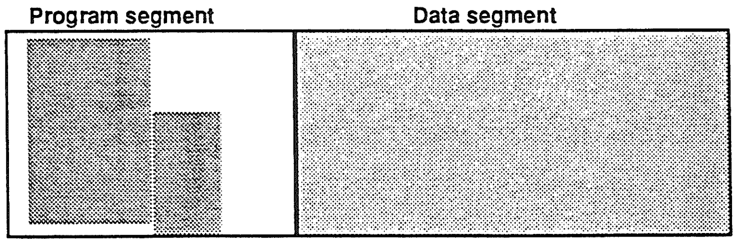
PYRAMID TECHNOLOGY ©1989

Corollary to #7: Databases Are Perverse

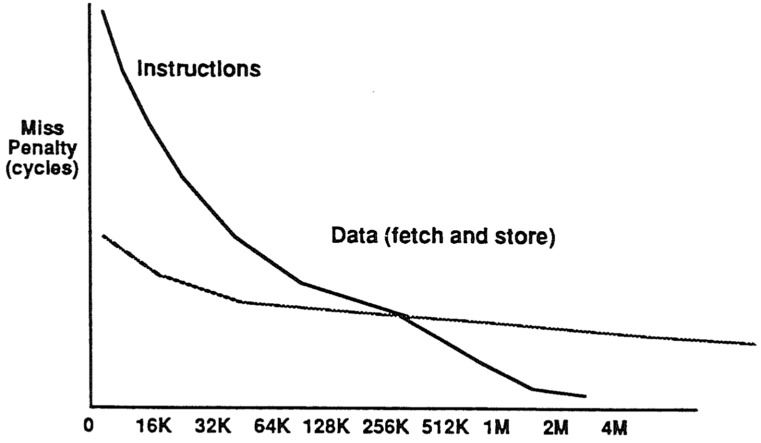
Random record access (a la TP1):



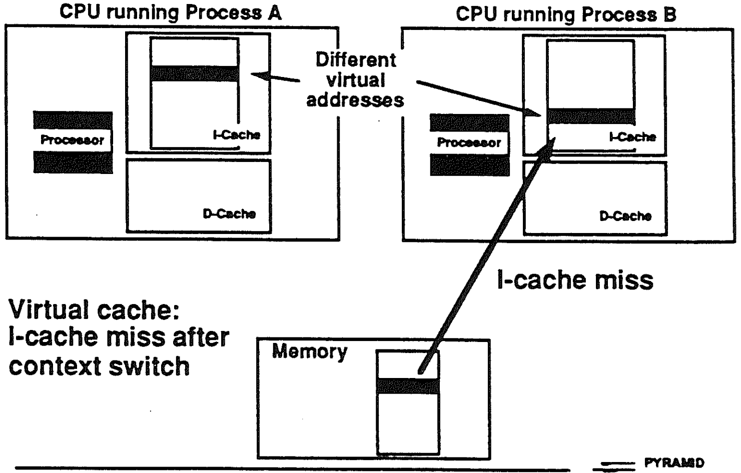
Database Text and Data Working Set



Design Problem: Database Data Access Pattern Is Resistant To Large Caches

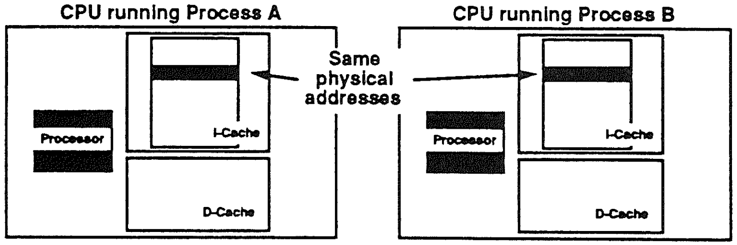


Design Rule #8: Use Physical I-caches

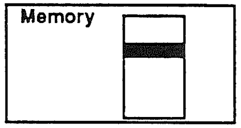


Virtual cache:
I-cache miss after
context switch

Design Rule #8: Use Physical I-caches (cont)



Physical cache:
I-cache hit after
context switch



Summary: Design Rules

(1) Design system for very large memories
0.5-1.0 MB per concurrent user

(2) Multiprocessors are neccessary for most databases

(3) Use as few processors as possible
(a) latching conflicts
(b) database logging
(c) operating system seriality

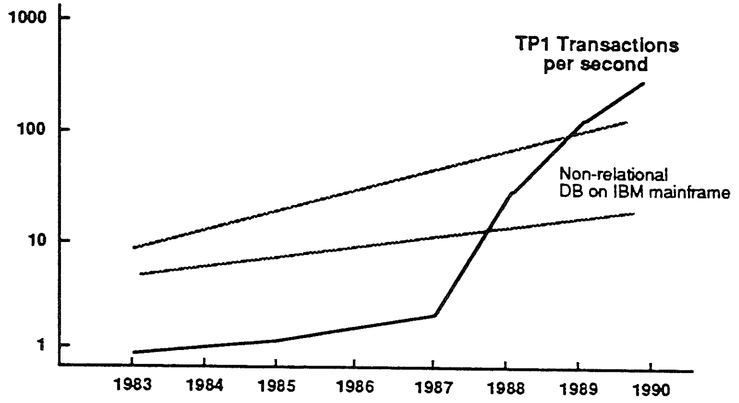
(4) Use fastest processors possible

(5) Use very large caches

(6) Many small caches are less efficient than a few large caches

(7) Secondary caches are necessary but not sufficient.

Relational Database Performance 1988-90



Cost of an Automated Solution: Revisited

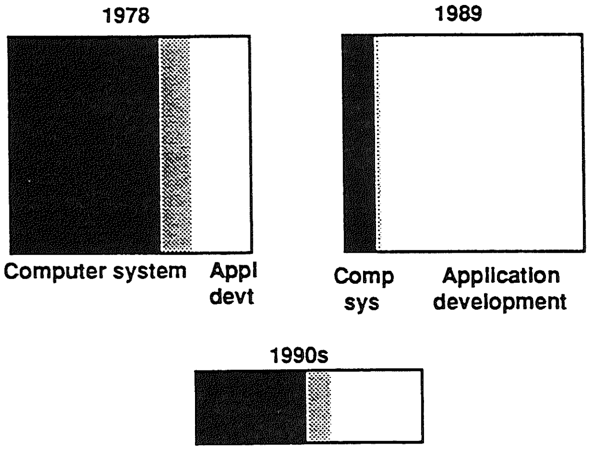
C(Soln) = C(HW) + C(OS) + C(otsSW) + C(customSW)

1978 US:
= IBM3083 VM/MVS COBOL(50 prog years)
C(soln) = US$1.3M + $200K + 0 + 50 * 13K = $2.15M

1989 US:
= YAFUB UNIX C (50 prog years)
C(soln) = US$200K + $20K + 0 + 50 * 38K = $2.12M

1990s:
= Big Server UNIX C (5 prog years)
C(soln) = US$300K + $50K + 0 + 5 * 50K = $600K

Redistribution of Cost of Automation



PYRAMID TECHNOLOGY ©1989

Implications

PCs / workstations: good for interfaces, simple applications

... but representational size weapons are critical to fight increasing inherent complexity of commercial applications and rising cost of programmers

Representational size weapons generate huge demands on CPU power and memory size

Therefore, big machines (100s of MIPS and MBytes) are increasingly necessary

... but building such machines is well within our reach.

PYRAMID TECHNOLOGY ©1989

AUUG Institutional Members

ANL Limited
Adept Business Systems Pty Ltd
Alcatel STC Australia
Alcatel STC Australia
Aldetec Pty Ltd
Alliance Computer Centre Pty. Ltd
Apple Computer Australia
Apscore International Pty Ltd
Australian Electoral Commision
Australian Nuclear Science &;Technology Organisation
Australian Wool Corporation
BHP Melbourne Research Labs
Ballarat Base Hospital
Basser Department of Computer Science
Bond University Library Service
Bond University;School of Information and Computing Science
CADAD Support Section - SECWA
Capricorn Coal Management Pty. Ltd.
Civil Aviation Authority
Colonial Mutual
Commodore Business Machines Pty. Ltd.
Comperex (NSW) Pty Ltd
Computer Power IR+D, NSW Branch
Computer Power Today IR+D
Computer Software Packages
Computerscene International
Corinthian Engineering Pty Ltd
Cybergraphic Systems Pty Ltd
DBA Limited
DMR Group
Data General
Davey Products Pty Ltd
Dept of Agricultural + Rural Affairs
Dept of Industry, Technology and;Resources, Victoria
Digital Equipment Corporation; (Australia) Pty. Limited
Earth Resource Mapping Pty Ltd
Elxsi Australia Ltd
Epson Australia Pty Ltd
Flinders University;Discipline of Computer Science
Genasys II Pty Ltd
Golden Casket Art Union
Gould Electronics Pty Ltd
Harris & Sutherland Pty Ltd
Hewlett Packard Australia Limited
Honeywell Software Centre
IBM Australia Limited
ICL Australia Pty. Ltd.
IPS Radio and Space Services

AUUG Institutional Members

Ipec Transport Group
Labtam Information Systems Pty Ltd
Macquarie Bank Limited
Macquarie University
Main Roads Department, Queensland
Mincom Pty. Ltd.
Motorola Communications Australia
NEC Information Systems Australia Pty Ltd
NSW Parliament
National Engineering Information Services P/L
Nixdorf Computer Pty Limited
Olivetti Australia Pty Ltd
Olympic Amusements P/L
Overseas Telecommunications Corporation
Performance Systems
Prentice Computer Centre
Prime Computer of Australia Ltd
Pyramid Technology Corporation
Q. H. Tours Limited
Queensland Department of Mines
RMIT Computer Centre
Roads and Traffic Authority NSW
SEQEB
Sigma Data Corporation Pty Ltd
Sola International Holdings Ltd
South Australian Institute of Technology
Sphere Systems Pty Ltd
State Bank Victoria
State Bank of NSW
State Library of Tasmania
Sugar Research Institute
Sun Microsystems Australia
Swinburne Institute of Technology
Sybase Australia Pty Ltd
Tattersall Sweep Consultation
Telecom Network Engineering - C.S.S
The Australian Artificial Intelligence Institute
The Australian National University
The Mathematics and Computing Department - BCAE (Kelvin Grove Campus)
The University of Adelaide
The University of New South Wales
The University of Wollongong
University of New England
University of Technology Sydney;Computing Services Division
Vicomp
Wang Australia Pty Ltd
Wyse Technology Pty. Ltd.
Yartout Pty Ltd

;login:

The USENIX Association Newsletter

Volume 14, Number 4 **July/August 1989**

CONTENTS

The closing date for submissions for the next issue of *;login:* is August 25, 1989

THE PROFESSIONAL AND TECHNICAL
UNIX® ASSOCIATION

Call for Papers
Winter 1990 USENIX Conference

January 22-26, 1990
Omni Shoreham
Washington, DC

Papers are sought in all areas of UNIX-related research and development for the technical program of the 1990 Winter USENIX Conference. Papers which are accepted for the conference will be published in the conference proceedings and shall be presented during the three days of technical sessions at the conference.

Appropriate topics for presentation include, but are not limited to:

New Tools and "Little Languages"
UNIX and AI:
  Intelligent Systems
  Neural Nets
Ada and UNIX –
  Real Experience and Future Expectations
File Systems and Servers
  Failsafe and Failsoft File Managers
  Hierarchical File Migration
  Version Control
Architectures and Compilers
Software Release Systems and Servers
Documentation issues
Distributed Systems and Services
Networking and Security
User Interfaces and
  User Interface Management Systems
Experiences and Novel Applications

All submissions will be considered – however, papers detailing new and interesting work will be regarded much more favorably than thinly disguised product announcements or re-runs of previous reports. The Winter 1990 conference is requiring that **extended abstracts** (and *not* full papers, as in previous conferences) be submitted. An extended abstract should describe the nature of the work, summary of results and conclusions, and should be between 1000-2000 words long (two to three typeset pages). Time is scheduled for authors of accepted papers to complete their submissions; therefore, extended abstracts will only be accepted when it is felt that a complete and worthy paper can be produced by the final due date.

The final paper should include a 100-300 word abstract, a discussion of how the paper relates to other work, illustrative figures (where appropriate), and citations to relevant literature. Only previously unpublished submissions will be considered. Final papers should contain on the order of 8-12 pages of single spaced typeset materials. All final papers must be submitted in a camera-ready format or electronic format (*troff -ms* if possible) – typewritten or dot-matrix output is **not** acceptable as final output. For authors without access to a laser printer or typesetter, appropriate facilities will be provided by the program chair.

Please submit abstracts as soon as possible, and mail one hard-copy and one electronic-copy to the addresses below. The final deadline for receipt of submissions is **August 14, 1989**; abstracts received after this deadline will not be considered. Notification of acceptance or rejection will be made by **September 25, 1989**. Final camera-ready papers are due by **November 17, 1989**.

To submit a paper or request additional information, please contact:

Daniel V. Klein
Washington USENIX Technical Program
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

+1 (412) 268-7791 (work)
+1 (412) 422-0285 (home)
+1 (412) 268-5758 (FAX)
wash-usenix@sei.cmu.edu
uunet!sei.cmu.edu!wash-usenix

Large Installation Systems Administration III Workshop and Tutorial Program

September 6-8, 1989, Marriott Hotel, Austin, TX

In light of two successful workshops on Large Installation Systems, there is a demonstrable benefit in bringing together system administrators of sites with 100 or more users (on one or more processors) to compare notes on solutions to a variety of common problems.

Tutorials – Wednesday, September 6

A two-track tutorial program will be offered in conjunction with the workshop. Attendees will be able to change between tracks at each topic change. The tutorial notes will include material from both tracks. The tutorials will be presented by Rob Kolstad of Prisma, Inc., and Evi Nemeth and Trent Hein of the University of Colorado.

Joint discussion of ethics, privacy, and security in centralized and distributed systems

Management policies	Internet networking
Security	NFS/YP
Large scale backups	UUCP
Machine Room Organization	NeWS
Performance	sendmail
User ID management	System upgrades/local documentation

Joint wrap-up discussion: Public Domain Software, Miscellaneous Topics

Tentative Technical Program

Thursday, September 7

Introductory Remarks	Alix Vasilatos, Program Chair
Keynote Address	
Networked Heterogeneous Systems I	Paul Graham, Chair
Accounting	Bjorn Satdeva, Chair
Network Administration	Kevin Smallwood, Chair
Birds of a Feather Sessions	

Friday, September 8

Networked Heterogeneous Systems II	Bjorn Satdeva, Chair
Panel: Distributed Services	
Work in Progress	Paul Graham, Chair
Security	Alix Vasilatos, Chair
Backup	Kevin Smallwood, Chair

The registration fees are $225 for the tutorial and $200 for the technical sessions. You may register for only the tutorial class, only the technical sessions, or both. The registration deadline is **August 30, 1989**. For registration and hotel information, please call the USENIX Conference Office at (714) 588-8649.

Workshop on Experiences with Distributed and Multiprocessor Systems†

October 5-6, 1989, Marriott Hotel, Ft. Lauderdale, FL

The goal of this workshop is to bring together individuals who have built, are building, or will soon build distributed and multiprocessor systems, especially operating systems. The workshop will feature full presentations and work-in-progress presentations on aspects of building and using these systems. The workshop will provide a forum for individuals to exchange information on their experiences, both good and bad, in designing, building, and testing their systems. This includes experiences with coding aids, languages, distributed debugging tools, prototyping, reuse of existing software, performance analysis, and lessons learned from use of such systems.

Tentative Schedule

Thursday, Oct. 5

8:30 **Opening remarks.** George Leach, Workshop Chair

8:45 **Session I: Objects and Virtual Memory**

A Distributed Implementation of the Shared Data-Object Model by Henri E. Bal, M. Frans Kaashoek and Andrew S. Tanenbaum (Vrije Universiteit, Amsterdam)

An Implementation of Distributed Shared Memory by Umakishore Ramachandran and M. Yousef A. Khalidi (Georgia Institute of Technology, Atlanta)

An Object-Oriented Implementation of Distributed Virtual Memory by Gary M. Johnston and R. H. Campbell (University of Illinois at Urbana-Champaign)

10:45 **Session II: Process Control**

Experience with Process Migration in Sprite by Fred Douglis (University of California, Berkeley)

Dynamic Server Squads in Yackos by Debra Hensgen and Raphael Finkel (University of Kentucky, Lexington)

Fine-Grain Scheduling by Henry Massalin and Calton Pu (Columbia University, New York)

1:30 **Session III: Performance Considerations**

The Parallelization of Mach/4.3BSD: Design Philosophy and Performance Analysis by Joseph Boykin and Alan Langerman (Encore Computer Corporation, Marlborough)

Efficient Implementation of Modularity in RAID by Charles Koelbel, Fady Lamaa, and Bharat Bhargava (Purdue University, West Lafayette)

Making libc Suitable for use by Parallel Programs by Julie Kucera (Convex Computer Corporation, Richardson)

3:30 **Session IV: Concepts**

Revolution 89 -or- Distributing UNIX Brings it Back to its Original Virtues by Francois Armand, Michel Gien, Frederic Herrmann, and Marc Rozier (Chorus Systems, En Yvelines)

† Sponsored by the USENIX Association and the Software Engineering Research Center (SERC), in cooperation with ACM SIGOPS and ACM SIGSOFT, and with the IEEE-CS TC on OS and IEEE-CS TC on Distributed Systems.

A Network File System Supporting Stashing by Luis L. Cova, Rafael Alonso, and Daniel Barbara (Princeton University)

4:20 Work-in-Progress presentations.

Friday, Oct. 6

8:30 **Session V: Multiprocessors**

TUMULT-64: a real-time multi-processor system by Pierre G. Jansen and Gerard J. M. Smit (University of Twente, Enschede)

Experiences with a Family of Multiprocessor Real-Time Operating Systems by Prabha Gopinath and Thomas Bihari (Philips Laboratories, Briarcliff Manor)

Implementation Issues for the Psyche Multiprocessor Operating System by Michael L. Scott, Thomas J. LeBlanc, and Brian D. Marsh (University of Rochester)

10:30 **Session VI: Tools**

Experience with P/Mothra: A Tool for Mutation Based Testing on A Hypercube by ByoungJu Choi and Aditya P. Mathur (Purdue University, West Lafayette)

Debugging and Performance Monitoring in HPC/VORX by Howard P. Katseff (AT&T Bell Laboratories, Holmdel)

CAPS – A Coding Aid used with the PASM Parallel Processing System by James E. Lumpp, Jr., Samuel A. Fineberg, Wayne G. Nation, Thomas L. Casavant, Edward C. Bronson, Howard J. Siegel, Perre H. Pero, Thomas Schwederski, and Dan C. Marinescu (Purdue University, West Lafayette)

The Implementation of Aide: A Support Environment for Distributed Object-Oriented Systems by Rodger Lea and Johnathan Walpole (University of Lancaster, Bailrigg)

1:30 **Session VI: Object-oriented Construction**

Experience With Implementing and Using An Object-Oriented, Distributed System by D. Decouchant, M. Riveill, C. Horn, and E. Finn (Bull-IMAG, Gieres)

Prototyping a distributed object-oriented OS on UNIX by Marc Shapiro (INRIA, Le Chesnay)

Clouds: Experiences in Building an Object Based Distributed Operating System by Umakishore Ramachandran, Sathis Menon, Richard J. LeBlanc, M. Yousef A. Khalidi, Phillip W. Hutto, Partha Dasgupta, Jose M. Bernabeu-Auban, William F. Appelbe, and Mustaque Ahamad (Georgia Institute of Technology, Atlanta)

3:30 **Session VII: Communications, Heterogeneous Systems, and the A-word**

Experiences with Efficient Interprocess Communication in Dune by Marc F. Pucci and James Alberi (Bell Communications Research, Morristown)

Using Transputer Networks to Accelerate Communication Protocols by Horst Schaaser (Hewlett-Packard Laboratories, Bristol)

ARCADE: A Platform for Heterogeneous Distributed Operating Systems by David L. Cohn, William P. Delaney, and Karen M. Tracey (University of Notre Dame)

A Decentralized Real-Time Operating System Supporting Distributed Execution of Ada Tasks by Roger K. Shults (Rockwell International-Collins Divisions, Cedar Rapids)

For information on registration, contact the USENIX Conference Office.

5th USENIX Computer Graphics Workshop

November 16-17, 1989, Doubletree Hotel, Monterey, CA

The theme of the 5th USENIX Computer Graphics Workshop is "personal graphics." By this, we mean the use of computer graphics to aid, benefit, or amuse a single person. Generally, personal graphics applications are highly interactive, so that the user has a great deal of control over the result. Furthermore, the graphics is frequently not an end product, but is instead a communication medium between the user and computer. Examples of personal graphics might include desktop publishing, data visualization programs (e.g., MacSpin), windowing systems, micro-world simulations (Kay's vivarium?), and "performance" graphics (e.g., video weirdness). It probably does not include ray-tracing, yet another VLSI graphics chip, or fast rendering algorithms. A distinguishing feature is that the user is included as an integral part of the description of the system. One question that may be addressed by presentations in this workshop is "How are 'ordinary people' going to effectively use computer graphics in their daily lives?"

A program will be available in August. The workshop chair is Spencer W. Thomas, EECS Department, University of Michigan.

For information on registration, contact the USENIX Conference Office.

Preliminary Call for Participation

USENIX C++ '90

Tentatively in late-April 1990 in California

C++ continues to show explosive growth as the object oriented implementation language of choice for production level work. The nearly-annual C++ conference is a haven for those who use the language, those who develop the language, and those who are interested in the language. The conference enables them to take a look at where C++ has been, where is it now, and where future developments should take it.

The conference will consist of a day of tutorials and classes and two days of technical sessions. Papers are invited on all aspects of C++, from the development of compilers and preprocessors to case studies of projects which have used the language. Proposals for tutorials or classes on systems which make use of C++ or on the uses of C++ are also invited.

Paper abstracts and tutorial proposals are due **January 12, 1990**. Abstracts should be no more than two pages, and should describe the work in sufficient detail to allow the referees to judge the merit of the work. Tutorial proposals should be no more than four pages in length, and should describe the content, purpose, and intended audience. Abstracts and tutorial proposals should be submitted either electronically (preferred) or in hardcopy; electronic submissions should be either plain text, *n/troff*, or Postscript. Notification of acceptance will be made by **February 2, 1990**; final papers in camera ready form must be received by **March 9, 1990**. Accepted papers which meet this deadline will be published in a conference proceedings.

Abstracts and proposals should be sent to:

Jim Waldo
Apollo Computer
330 Billerica Road
Chelmsform, MA 01826

waldo@apollo.com
decvax!apollo!waldo

EUUG Autumn '89 Conference and Exhibition

September 18-22 1989
Vienna, Austria

The Autumn 1989 European UNIX systems User Group Technical Conference will include technical tutorials on Monday and Tuesday, followed by a three day conference and exhibition. Subject areas of the technical program will include such topics as security, fault tolerance, transaction processing, RISC archictectures, and user interfaces.

For more information on the conference and tutorial program, please contact:

The Secretariat
EUUG
Owles Hall
Buntingford Herts, SG9 9PL, UK

Tel: +44 763 73039
FAX: +44 763 73255

Email: euug@inset.uucp

Call for Papers
EUUG Spring '90 Conference

April 23-27, 1990
Munich, West Germany

The EUUG invites papers from those wishing to present their work. Full papers or extended abstracts must be submitted. Suggested subject areas include, but are not limited to:

Standards for UNIX Systems
Internationalisation
Object Oriented Development Tools
Object Oriented Graphical Toolkits
Object Oriented Languages
Program Generators for Commercial Applications
Network Administration
Security Issues and Authentication Techniques
Document Context Architectures

Submission deadline:	November 15, 1989
Acceptance notification:	December 15, 1989
Final paper:	February 10, 1990

Full papers or extended abstracts must be submitted by post to the EUUG Secretariat (address above), and, if possible, in electronic form to *euug-munic@cwi.nl*. Notification of acceptance will be acknowledged by return post.

Call for Papers
Convention UNIX 90
March 26-30, 1990, Paris, France

Convention UNIX 90, organized by the French Association of UNIX Users (AFUU) and the Bureau International de Relations Publiques, will have parallel technical, user, and product conferences, and an exhibition and tutorials.

The technical conference will cover a wide range of technical topics. The user conference will stress users experiences, analyses, and strategies in dealing with UNIX. The product conference will have exhibitor presentations concerning existing or future products. Submissions are invited for each of the conferences.

Submissions should include a title, author(s) and affiliation(s), a mention of the conference for which the submission is offered, and a one page abstract. Abstracts must be received by **Oct. 15, 1989**. Notification of papers selected will be made by **Dec. 1, 1989**. Full papers are due by **Jan. 31, 1990**.

Submissions should be made to:

A.F.U.U.
Attn: Secrétariat de la Conférence
Convention UNIX 90
11, Rue Carnot
94270 Le Kremlin-Bicetre
France

(+33) (1) 46.70.95.90
afuuconf@inria.inria.fr

Baltimore Terminal Room

Many of you who were at the Baltimore USENIX Conference used the Terminal Room located in the Hyatt Hotel. The statistics from the Xylogics Terminal Server showed an average of 1,100,000 bytes received and 930,000 bytes sent per hour of operation across the Internet SLIP link. Approximately 400 local and toll free number calls were made from the room during the week, and at least 87 Sun cartridge tapes of GNU software were made.

I would like to thank Telebit, Xylogics, AT&T, and OSF for providing equipment and funds; Judy DesHarnais, Mike Ballard, Cerafin Castillo, John Loverso, Van Jacobson, Len Tower, Edgar Merke, Evi Nemeth, Ed Gould, and Trent Hein for their assistance in getting the room organized; and all the volunteers who worked long hours so that the conference attendees would not miss any urgent information at their home sites.

Sonya Neufer
Terminal Room Coordinator

And the winners are!

Attendees at the USENIX Baltimore Exhibition who registered at the Apple Computer and/or IBM Corporation booths were eligible to win a computer.

The winner of the Apple Macintosh IIxc system and four application software packages was Peter Lega from Digital Equipment Corporation. Richard Karpinski from the University of California, San Francisco, won the fully configured IBM/RT Model 135.

Informal Programming Chair

The Association would like to thank Alix Vasilatos for being the Informal Programming Chair at Baltimore. For the Winter 1990 Conference, Sonya Neufer has been appointed. In addition to her responsibliities as terminal room coordinator, Sonya will coordinate such activities as the opening night reception, orientation session, some BOFs, and overseeing the many other functions that are not part of the technical program.

Long-Term Calendar of UNIX Events†

1989 Sep 6-8	*Large Systems Admin. Workshop	Austin Marriott, Austin, TX
1989 Sep 12-13	MALNIX	Kuala Lampur, Malaysia
1989 Sep 18-22	EUUG	Vienna, Austria
1989 Sep 19-22	ACM SIGCOMM	Austin, TX
1989 Sep 25-29	GUUG	Wiesbaden, Germany
1989 Sep 27-29	Workstation Operating Systems	Pacific Grove, CA
1989 Oct 5-6	*Distributed Systems Workshop	Marriott Marina, Ft. Lauderdale, FL
1989 Oct 16-20	IEEE 1003	Brussels, Belgium
1989 Nov 1-3	UNIX Expo	Javits Conv. Ctr., New York, NY
1989 Nov 6-10	DECUS	Anaheim, CA
1989 Nov 9	NLUUG	The Netherlands
1989 Nov 9-10	14th JUS UNIX Symposium	Osaka, Japan
1989 Nov 16-17	*Graphics Workshop V	DoubleTree Hotel, Monterey, CA
1989 Nov 24	AFUU	Paris, France
1989 Dec 5-6	JUS UNIX Fair 89	Tokyo, Japan
1989 Dec 8-9	Sinix	Singapore
1990 Jan	UNIX in Government	Ottawa, Ont.
1990 Jan 22-26	USENIX	Omni Shoreham, Washington, DC
1990 Jan 23-26	UniForum	Washington Hilton, Washington, DC
1990 Jan 29	IEEE 1003	New Orleans, LA
1990 Mar 26-30	AFUU	Paris, France
1990 Spring	*C++ Conference	California
1990 Apr	IEEE 1003	Montreal, Que.
1990 Apr 23-27	EUUG	Munich, Germany
1990 May	UNIX 8x/etc	/usr/group/cdn; Toronto, Ont.
1990 May 7-11	DECUS	New Orleans, LA
1990 Jun 11-15	USENIX	Marriott Hotel, Anaheim, CA
1990 Sep 11-14	AUUG Conference	Southern Cross, Melbourne, Australia
1990 Oct 22-26	EUUG	Nice, France
1991 Jan 21-25	USENIX	Grand Kempinski, Dallas, TX
1991 Jan 22-25	UniForum	Infomart, Dallas, TX
1991 Feb	UNIX in Government	Ottawa, Ont.
1991 May	UNIX 8x/etc	/usr/group/cdn; Toronto, Ont.
1991 May 20-24	EUUG	Tromso, Norway
1991 Jun 10-14	USENIX	Opryland, Nashville, TN
1991 Sep 16-20	EUUG	Hungary
1992 Jan 20-24	USENIX	Hilton Square, San Francisco, CA
1992 Jan 21-24	UniForum	Moscone Center, San Francisco, CA
1992 Spring	EUUG	Jersey, UK
1992 Jun 8-12	USENIX	Marriott, San Antonio, TX
1993 Jan	USENIX	Town & Country, San Diego, CA
1993 Mar 2-4	UniForum	Washington, DC
1993 Jun 21-25	USENIX	Cincinnati, OH

† Partly plagiarized from John S. Quarterman of TIC and Alain Williams of EUUG by EY.
* USENIX Workshops

New Conference Sessions

USENIX gatherings have grown from a small group of people exchanging tricks of the trade to conferences of over 2000 attending two days of tutorials and three days of refereed papers. The interests of attendees have expanded to encompass networking, system administration, programming languages and development environments, text processing, windowing systems, user interfaces, and turnkey applications. Over the years, USENIX has adapted to this expansion by adding BOFs, tutorials, a vendor exhibit, workshops, published proceedings, and the journal. What's next?

During the Women's BOF at the San Diego USENIX conference, a suggestion was made to augment the existing conference format to help bring people of matching skills and interests together. Beginning with its Winter '90 conference, USENIX will introduce experimental parallel sessions to provide new and complementary forums of technical excellence. Attendees will be free to migrate between these new sessions and the traditional sessions at will. Suggestions for new sessions include:

- the informal exchange of information on current practical problems, resulting battle scars and solutions
- "short courses" about specific tools and tricks
- panel sessions providing experienced volunteers to answer questions
- survey talks to broaden the expertise of members

Lori Grob and Eric Allman have volunteered to do the initial organization of the new sessions at Washington and at the following Summer conference in Anaheim. The early plans are modest: five free "short courses" spanning the three conference days, including:

- Andrew Hume on make and regular expressions.
- Eric Allman on C Style/Portability.
- John Quarterman will discuss survival in a global network.
- The traditional meta-talk on submitting and presenting papers will be moved into this series.

Other possibilities include talks on introduction to parallel programming, on submitting and diagnosing problem reports, and on fundamental principles in UNIX.

These new sessions are planned for before and after lunch on Wednesday and Thursday, and before lunch on Friday. The final schedule will be included in the regular conference mailing.

We fully expect this new session to evolve and hope to be guided by discussions at the upcoming sessions and continual feedback from members. If the new sessions succeed, we may continue them as an annual event. Please send any and all comments and contributions regarding possible speakers,* topics, and formats to

```
newsession@usenix.org
```

Please don't bother us with offers of vendor presentations.

Sharon Murrel
Eric Allman
Lori Grob
Ellie Young

* Like all speakers in the technical program, these will get free registration only.

;login: 14:4

The FaceSaver Project

The USENIX FaceSaver Project reappeared at the Summer Conference in Baltimore. This time it was set up in the vendor exhibit area, which gave us the room, power and cooling necessary to run a comfortable operation. In just three days we collected over 900 portraits, which have been sent to *uunet*. These have been added to, or replace some of the 2222 portraits already on file, bringing the total of "unique" faces to 2957. They are available via automatic mail request or via ftp. In addition to names and E-mail addresses, most of these portraits may contain phone numbers, companies and street addresses as supplied by the attendees.

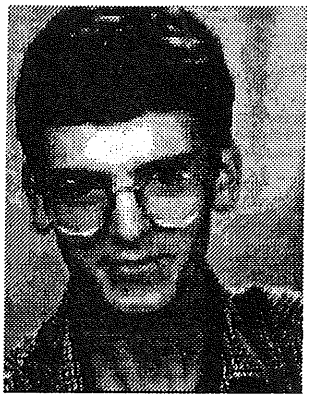
Kathryn Johnson & Craig Schwartz

I want to thank the USENIX Association for sponsoring and funding this project, Rick Adams and UUNET Communications Services, Inc. for providing over 82 Mb of on-line storage for the pictures as a service to the community, the QMS Corporation of Mobile, AL for providing a laser printer, and Bell Technologies (now a division of the Intel Corporation) of Fremont, CA for providing their 80386 System V Release 3 system. The latter two enabled us to run two portrait stations in parallel.

We were all very pleased that Craig Schwartz was our photographer again; he was assisted by Kathryn Johnson. I am grateful to Vincent Cawley and Mary Salus for cheerfully staffing the data terminals and dealing with the release forms, and to B. Edward R., Sir Peter Langston, Dan Klein, Ken Arnold, Paul Kooros and Reidar Bornholdt for their generous help. I also want to thank John Donnelly and Judy DesHarnais for taking care of the arrangements with the convention center and hotel.

As in the past, attendees could have their pictures taken and/or retaken at no charge. The care taken by Craig and Kathryn in getting attractive portraits clearly justified the extra time they spent with each person. Some of the faces were transferred via cartridge tape to the show network file server during the conference and were brought up on workstations at the Sun® booth; the serial interface board in the network file server wedged every time we tried to *uucp* the pictures over, preventing full database transfer during the show.

Mary W. Salus


Vincent Cawley

I have deposited a revised C program for printing individual pictures, and a C program with an associated PostScript program for generating a page of labels from the portrait data with *uunet*, for all to use and enjoy.

Lou Katz

Saver of Lost Faces

Information on accessing the faces from *uunet*:

Welcome to faceserver, a system for distribution of faces by electronic mail and other means. This text is the reply you'll get to:

```
        mail uunet!faceserver <EOF
        help
        EOF
```

To examine the full index for all the faces send a request of the form:

```
        send full-index
```

You may include several requests in a single piece of mail, but put each on a separate line. Faces are usually stored by their electronic mail addresses: e.g. *rick@uunet.uu.net* would be stored as *uunet.uu.net/rick*. To get it, you would send the command:

```
        send uunet.uu.net/rick
```

uucp sites are stored in a domainish format: e.g. *usenix!lou* would be stored as *usenix.uucp/lou*. So, to get it, you would send the command:

```
        send usenix.uucp/lou
```

Those faces that do not have an email address associated with them are stored in the directory *no-email-address* by their first_lastname. e.g.:

```
        send no-email-address/p_s_langston
```

(see the full-index for the actual names). The format of the files is described in the file format. To get it send:

```
        send format
```

To request programs send the command:

```
        echo send index from programs | mail uunet!faceserver
```

To get a specific program:

```
        echo send WHATEVER from programs | mail uunet!faceserver
```

Send the requests to *"uunet!faceserver"* even though replies appear to be coming from *"uunet!faceserverd"*. You'll be talking to a program, so don't expect it to understand much English.

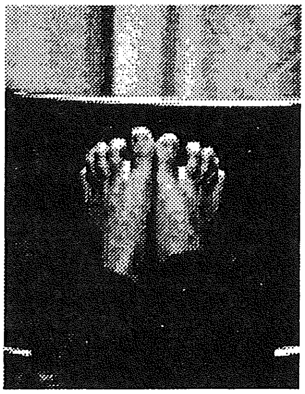
A picture file contains some or all of these information lines:

```
FirstName: Random J.
LastName: Attendee
E-mail: rja@nullsys.uucp
Telephone:1 800 555 1212
Company: Computers R Us
Address1: 1234 Fifth Street
Address2: MS 275W-137N
CityStateZip: Gotham, UX 99999-0000
Date: Jun 12, 1989
PicData: (Actual data) width - height - bits/pixel
Image: (Should be transformed to) width - height - bits/pixel
(Blank line)
Hexified picture in scanline order.
```

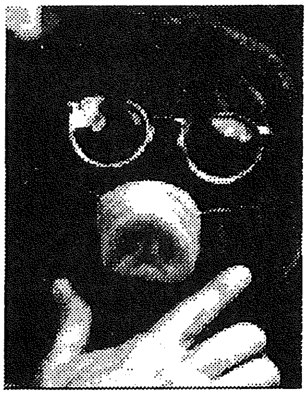
Separated at Birth?

Selected "Faces" from Baltimore

USENIX Weanie

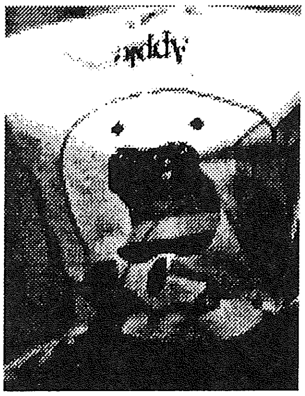


So Tired

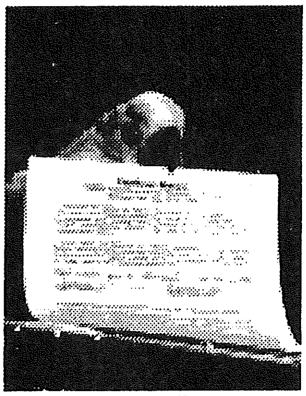


Abominable Hackerman

Dogs of USENIX

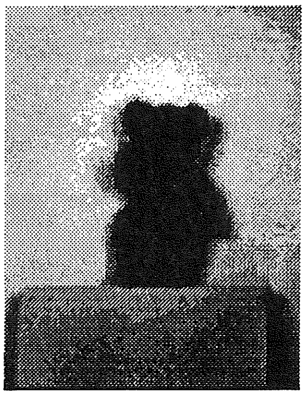


Baghwan Bagman



CyberPink Barry

Fido



Henry

Beanheads of USENIX

;login: 14:4

USENIX Online Index

What Is It:

The USENIX online index is an electronically available list of papers published by the USENIX Association and related groups. The index is kept as a simple ASCII file, in refer/bib format, sorted by author. It contains information about papers published in USENIX and UNIX-related conference and workshop proceedings, newsletters, journals, and the like.

In some cases, electronically readable versions of full papers or abstracts are also available. If a paper is available online, this is indicated in its index entry.

The index is updated approximately monthly.

How to Get the Index:

The index is available online from uunet, either via a mail server or anonymous ftp. The index is about 200K, and available only in its entirety. To get it via electronic mail:

```
echo send bibliography | \
mail uunet!library
```

A (non-human) server will automatically break the index up into mailable chunks (if necessary), and return it to the sender of the mail.

Or, the index can be retrieved via anonymous ftp to uunet.uu.net:

```
ftp> get library/bibliography
```

To get a help file:

```
echo help | mail uunet!library
```

To pick up the date the index was last updated:

```
echo send date | mail uunet!library
```

(Note - There is no person associated with "library-request" and it will never be read by human eyes.)

Online Papers:

We are actively soliciting the donation of electronic versions of papers to include in the library. If you have a paper you would like to donate, either with or without releasing copyright, contact the office for specific details. When the paper retrieval capability is fully functional, we will announce the procedures.

Publications Indexed:

Currently we have indexed all available issues of the following:

USENIX:
  Conference proceedings
  Workshop proceedings
  Computing Systems Journal
  Newsletters (*;login:*)

European UNIX User Group:
  Conference proceedings
  Newsletters

Software Tools User Group:
  Conference proceedings

Australian UNIX User Group:
  Newsletters

The *UNIX Review* periodical is currently being indexed and will soon be available. Other sources (AFUU, GUUG, NZUSUGI, etc.) are being continually evaluated and will be included as deemed suitable.

More Information:

For additional information about the online index and library, and/or instructions for donating papers, contact:

```
usenix!index    (index@usenix.org)
```

Or write to the USENIX Association.

Book Review

Xlib Programming Manual (Volume One)

by Adrian Nye

(O'Reilly and Associates, Inc.) 611 pages

Xlib Reference Manual (Volume Two)

(O'Reilly and Associates, Inc.) 700 pages

Reviewed by Marc Donner
donner@ibm.com

According to all the X gurus I know, one should never need to read or use books like these two. One is supposed to write X applications using a toolkit (any of a number, including Xt from MIT, Xaw from CMU, and others) and never descend to the level of the X libraries. If this is the case, who might be customers for these two books? Toolkit writers are the first who come to mind. Book reviewers come second. In a more serious vein, I suspect that the X application programmer will not feel secure unless he (or she) has a copy of the contents of these two books at hand.

The material in these books should be substantially the same as that in the Xlib documentation that comes with X11R3. I looked at a printed copy of the release documentation to see what might make me want to spend money for thirteen hundred pages of documentation, when I could print fewer than three hundred on my local printer.

The book is up-to-date on several minor matters of detail. For example, all atom names are now prefixed with `XA_` and the books reflect this, though the free documentation does not. I looked in the *Xatoms.h* file to see which was correct and it was the book. The man-page-like entries were adapted almost verbatim from the distribution, up to and including sentences ending with prepositions. On the other hand, the description included with each man page is much more detailed in the book and shows evidence of having been written by someone with a good understanding of English. Several improvements to the free man pages include reference to appropriate sections of volume one in the description, itemization of the error returns, and a listing of related functions.

Volume one is full of friendly descriptive text about how things work and some things to watch out for, but it seems to be pitched a little too low. The cross referencing is rich, though it isn't always clear that it is relevant. The exposition in volume one is based about some examples. The one pain is that the code is not easily available in machine-readable form. Perhaps it will make it to some future release of X11, but for now it isn't possible to play with the code without typing it in or sending $10 for a diskette.

The great strength of volume two is the large quantity of cross referencing provided by the various appendices. Each appendix tries to organize some part of the vast X name space according to one principle. The first appendix provides groups of related functions. The third appendix does the same thing with macros. The fifth appendix provides a man page for each event, while the sixth appendix details all the data structures. The only extra thing I'd have asked for here is a cross reference to the appropriate header file for most of the things named here (and in the man pages as well) Finding things in the X directories is always an adventure, even when you know their names.

All in all, the books are well executed and moderately well written. The content seems complete and about as free of obvious errors as X11 itself. The books are primarily aimed at application programmers, even though the X community is urging everyone to use toolkits instead of writing on top of the library directly. I suspect that until a lot more work is finished it will be necessary for application programmers to use the library interface from time to time, so these books will be useful to them.

White Paper on System Administration for IEEE 1003.7

Susanne W. Smith
Windsound Consulting

John S. Quarterman
USENIX Association

ABSTRACT: The new POSIX committee on System Administration, IEEE 1003.7, is attempting to standardize an area in which there is little prior art, and no generally accepted solutions to many of the known problems. It is a large area, and one that intersects with other areas such as networking (IEEE 1003.8) and application programming (IEEE 1003.2). Some of the most applicable prior art was not designed for operating system administration, but for network administration. Perhaps most importantly, there are two basic models for system administration. One must be chosen from the outset, and the choice will affect everything the committee does.

The USENIX Association has coordinated the production of this White Paper to set forth the basic issues the committee must address, to recommend certain choices it will have to make, and to outline some of the existing solutions that must be considered.

1. Introduction

The role of the system administrator has evolved over the years. Where once an administrator was responsible for a single machine or machines from a single vendor there is now often a network of machines from different vendors. Both the homogeneous single machine case and the heterogeneous networked case must be addressed by the system administration committee in producing a standard. This paper offers a description of system administration, its component tasks, and its scope; it recommends a model upon which to build the standard; it presents an overview of some current system administration practices; and it provides a reference list.

2. The Basic Model

The most basic choice for a system administration standard is between a single machine model and a model based on a network of machines.

2.1 A Single Machine

The results of 1003.7 will be applied to many machines that are not connected to any other machines, except perhaps by some indirect technique such as UUCP. The standard must be applicable to such machines. For this purpose, it need only specify a command interface and detail what the commands are supposed to accomplish.

However, there is a problem with basing the standard on a single machine as a model, because such a standard will not adapt well to a network of machines. The traditional methods used for administration of a single machine are not readily extended for a networked environment. For example maintaining user information on a single machine requires modifications to the */etc/passwd* file. In a networked environment this further requires maintaining the consistency of this file across many machines.

2.2 A Network of Machines

The number of machines connected to networks and the number of networks of computers have grown exponentially in the last several years. Many of us are accustomed to interacting with hundreds of computers on a local area network that is in turn connected to hundreds of thousands of other computers through wide area networks.

2.2.1 Remote Access

Many machines do not even have local disks: files are kept on a central server, which is accessed over the network. There may be more than one server, and two machines may even act as servers for each other for different parts of their file system trees.

2.2.2 Distribution

Databases may not have a single location. The mapping between login names and login IDs may be distributed among several machines. The whole database may be duplicated for redundancy. Parts of it may be kept in different places, for local control. A tree structure may be used.

2.2.3 Heterogeneity

Networked environments tend to have machines with many different hardware types and many different variants of operating systems. One machine may have */etc/passwd* and another may use a distributed database. The possible parameters to an operation may differ. Byte orders and word lengths vary.

2.3 Specifications

A single interface specification is not sufficient for a networking model of system administration. Three things are needed:

2.3.1 Interface

A specification of a programming interface is needed for a networked model, just as for a single machine model. Additional commands may be required for a networked model. But the specification of what the commands for the interface do has to be more complex for a networked model.

2.3.2 Database

Because of differences among machines in a heterogeneous network, such as varying byte orders, word lengths, and options supported, a generic specification of the information to be managed is needed. It is not practical to provide specifications for every type of machine and software and translations between them, because the numbers of specifications needed would be very large.

2.3.3 Operations

Given the interface specification of a command, and the database specification of the information it is to affect, a specification is also needed of how to communicate the necessary operation across the network. This should be done in a manner that is not specific to any of the underlying systems, but that can be translated into appropriate actions on any of them.

2.4 Network Management Standards

These issues and this kind of model have been addressed for the purpose of managing networks. It is possible that the work can be adapted and extended for use by 1003.7. Two components, a management station and a management agent, work together to perform network management functions in the following two protocols. The management station monitors and controls network elements. Management agents perform functions requested by the management station on the network element.

2.4.1 CMIP

The Common Management Information Protocol is the emerging ISO standard for network management. It uses a MIB (Management Information Base) and defines operations to be performed on it over a network.

2.4.2 SNMP

The Simple Network Management Protocol is in use now with TCP/IP on NSFNET. It addresses many of the basic network management problems and presents at least preliminary solutions to them. It proves the concept of a MIB with operations to manipulate it over a network.

2.4.3 ASN.1

Abstract Syntax Notation 1 is the ISO standard for encoding of information at the presentation layer of the seven layer ISO networking model. It is similar to Sun's XDR (External Data Representation) or Apollo's NIDL (Network Interface Definition Language) or NDR (Network Data Representation), but is more general than either. "ASN.1 is useful for describing structures in a machine independent fashion. Additionally, ASN.1 definitions can be written which convey to the human reader the semantics of the objects they define."[2]

Both CMIP and SNMP are written in terms of ASN.1.

2.5 Scope

The responsibilities of system administrators vary widely among installations. In some environments the tasks of the system administrator are defined as "anything it takes to keep computing services available for the user community." This definition could encompass everything from hardware diagnostics to network management. In some situations the system administrator may be responsible for user support and consulting. In other situations the tasks of the system administrator could be rigidly defined to only include password file maintenance and backups. Because there is no commonly-accepted definition of the scope of system administration, the committee needs to define which specific areas are included as the functions of a system administrator. Scope and definitions are also required parts of an IEEE standard. These should be addressed before commands and facilities are defined.

The committee should consider previous work in network management. The OSI model for network management consists of five functional areas: configuration management, performance management, fault management, accounting management, and security management. These functional areas map very well from network management to operating system management.

2.5.1 Configuration Management

Configuration management in the network sense is defined as "detection and control of the state of the network, both the logical and physical configurations of the network."[1] Configuration management in a system administration context would refer to the management of the information which defines a machine's functions. Configuration information determines whether a machine is a file server or client, a timesharing service or single user, diskless or diskful. The configuration data identifies the location of other machines and services.

2.5.2 Performance Management

Performance management could be defined as the collection and analysis of information that determines a machine's performance. Examples could be disk throughput, service access times, or cpu utilization.

2.5.3 Fault Management

Fault management is "the detection, isolation, and correction of abnormal operations in the network."[1] For system administration this would be detection of a service's failure, notification of the user community of failure, and the initiation of a backup service.

2.5.4 Accounting Management

Accounting management would be the management of the information required to determine the cost of using the system. This type of information is traditionally collected in units of disk storage blocks, cpu usage, and connect time.

2.5.5 Security Management

Security management is composed of the functions required to regulate access to system resources. User authentication, server verification, and security logs are functions of security management.

2.6 Recommendations

We strongly recommend the adoption of a network model. We also recommend that the committee focus on the entities to be managed and not the underlying transport protocol.

2.6.1 Specifications

Every command should be specified in terms of an interface, an information database, and operations to be performed over a network. Although the first of these alone would be sufficient in a single machine case, it is not adequate in a networked environment. A network model can be reduced to handle a single machine as a special case, but a single machine model cannot readily be expanded to support a networked environment. This is the main reason that a network model should be adopted instead of a single machine model.

2.6.2 Network Management

The committee should examine the work done to date on SNMP and CMIP, and should follow the progress of the committees that are producing those protocols. The 1003.7 MIB should be written in ASN.1.

3. Prior Art

We present here some examples of areas in which there is prior art that the committee should consider. This is not an exhaustive list of either the areas to be covered or the prior art in a specific area. There are other such areas, and we encourage others to submit proposals to the committee outlining them.

The examples are grouped according to the OSI model described above. Because system administration covers a broader area than network management the categories have been extended. Additional categories may be required to completely include all system administration functions.

3.1 Configuration Management

In addition to the description above, configuration management could include user configuration information. This would include the information required to describe a user and their environment (i.e. the location of their home directory). This area could also include queueing systems.

3.1.1 */etc/passwd*

The simplest database of user information is */etc/passwd*. It is a single file which contains information about each user. */etc/passwd* contains a user's login name, user-id, group id, encrypted password, optional full name and additional information, home directory location, and program to be executed upon successful completion of the login process. User information is added, changed, or deleted by using the command *vipw* or one of many available shell scripts and programs. Access to the information is controlled by file permissions.

This scheme works well in a single machine environment. This method requires each machine to have an */etc/passwd* file. As the number of machines on a network and the number of users increases, maintaining the file entries on each individual machine becomes an overwhelming, if not impossible, task for the system administrator. Different methods have been proposed to handle the task of maintaining an */etc/passwd* file on each machine in a network.

3.1.2 Yellow Pages

Yellow Pages (yp) is a distributed network lookup service. The Yellow Pages provide configuration information for a group of machines called a domain. A machine requesting information is a yp client and the machine providing the information is a yp server.

The information for a particular domain is a set of maps. Commonly the */etc/passwd* and */etc/hosts* files are replaced by yp maps. However, yp is indifferent to the type of data in the maps. A master flat file resides on a master server machine. Updates to the master file are made there. *dbm* is used to transform the flat file into maps. The maps are then propagated to all slave server machines. The number of slave servers is dependent on network size and topology. A single machine may serve more than one domain.

Once yp services are available (i.e. the maps have been made and the server machines configured) routines on the yp client machine must be modified to initiate yp requests rather than reading local files. Yp requests are remote procedure calls to a yp server.

3.1.3 Moira

"The purpose of Moira is to provide a single point of contact for authoritative information about resources and services in a distributed environment."[3] Moira is used to store information about users, the location of network services, the information needed to create the configuration files for network servers, as well as other information. Updates to the database are made using an application interface which is based on curses. Validity checks are performed on data to be updated. Access to each object in the database is controlled by an access control list. Statistics are kept about who modified the object last.

Network server configuration files are created from the Moira database and sent periodically to the appropriate servers. This eliminates the need to modify configuration files on individual machines. The Hesiod (see below) database is also created from the information in the Moira database.

3.1.4 Hesiod

Hesiod provides a read only front end for user information and the location of network services. User information is extracted from the Moira database and formated into ASCII files in BIND-compatible resource record format. Modifications have been made to BIND to accept and process Hesiod type queries.

Hesiod is used by the login process to acquire user information. Note, however, that Hesiod does not authenticate the user. Authentication is performed by Kerberos. Hesiod is also used by *lpr* to retrieve printer information traditionally stored in the */etc/printcap* file.

3.1.5 Berkeley Print Spooling

The Berkeley print spooling system was intended for use with network print services where printers are connected directly to the network or to the serial port of a host machine on the network. The command *lpr* is used to start the printing process. Line printer daemons (*lpd*) run on each machine in the network to control the spool area, queue, printing, and network transfers.

lpr looks up information for the requested printer in the */etc/printcap* file. This file contains information about each printer, such as location, filters needed, header page format, etc. It determines how to print a file from this information.

The *lpc* command provides queue management functions. *lpc* is used to restart and flush queues, abort jobs, and check the status of queues and printers.

3.1.6 MDQS – Multiple Device Queueing System

MDQS provides for local printer support, remote printer support, local and remote batch job scheduling, conversion of troff to device specific format, and sending graphics data to plotters. MDQS consists of a queue management daemon, a general-purpose spooler, a set of device specific despooled-data processing slaves, and utilities for setting form types, disabling service, viewing queues, etc.

A queue/device mapping table contains the queue name, device name, and the command to be executed as a slave process for the dequeued data. Remote printing and execution are handled by having slave processes which respool the data into the remote MDQS queues. The mapping table provides the flexibility for multiple devices to process from the same queue or one device to process from multiple queues. If NFS (network file system) or some similar mechanism is used, a single spooling area and daemon with control files can reside on one machine. This eliminates the need for respooling data into remote queues and the overhead of maintaining a local spooling area, daemon, and control files. The remote devices simply process the queue from the remotely mounted file system.

3.2 Security Management

Personal computers can be protected by making the machine physically secure. In a timesharing environment the operating system is used to protect one user from another. In a networked environment there are three approaches to prevent unauthorized access to network services: rely on the host to authenticate the user and then trust the host; require the host to prove its identity and then trust the host as to who the user is; make the user prove who they are for each network service.

3.2.1 Kerberos

"In an open network computing environment, a workstation cannot be trusted to identify its users correctly to network services."[4] Therefore Kerberos uses the third approach to system security; make the user prove their identity for each network service. In order for a user to prove their identity, they must be authenticated by Kerberos, not the workstation they are using. Passwords are never sent over the network, but are used locally to decrypt the authentication message from Kerberos. To prevent unauthorized use the local workstation destroys the user's password after using it to decrypt the initial Kerberos message.

Once a user has been authenticated they have the keys to request various network services. Different applications can choose different levels of protection. The first is authentication at initiation but subsequent messages are just accepted if from the same network address. The second is where each message is authenticated but the contents of the message are not encrypted. The third level of security is private messages where each message is authenticated and encrypted.

The Kerberos database contains a name, private key, and expiration date for each entity that will use Kerberos services. The master Kerberos database is kept and modified on one machine. Slave servers have read only versions of the database and provide read only types of services. Modification to the master database is accomplished by the administration server (KDBM server). There are two parts to this service: a client which will run on any machine in the network and a server that must run on the machine which houses the master database.

3.3 Accounting Management

Accounting is the recording and reporting of resource usage. This information can then be used to determine appropriate charges for a user.

3.3.1 Harvard Accounting System

This system would track disk usage, cpu time, logins, connect time, printed pages, and budget on an account-by-account basis and charge the appropriate accounts. It was designed to run in a single machine environment.

3.4 Fault Management

In order to restore service after a disk failure a sensible backup procedure needs to have been followed by the administrative staff. Basic commands to move data from one medium to another are described below.

tar and *cpio* file archiving and data interchange formats are the only backup formats specified in 1003.1.

3.4.1 System V Interface Definition (SVID)

3.4.1.1 *volcopy*

The *volcopy* command will make a literal copy of a file system. Copies can be made to another disk location or to tape.

3.4.2 SVID & Berkeley

3.4.2.1 *tar*

The *tar* command is used to create an archive file. Multiple files can be saved to and restored from a single tarfile. The tarfile can reside on various physical media. *tar* will read from standard input and write to standard output so that it can be part of a pipeline. This feature can be used for moving directories.

3.4.2.2 *cpio*

cpio copies a list of files to or from a *cpio* archive file. Pathnames and status information are kept along with the files.

3.4.3 Berkeley *dump / rdump / restore / rrestore*

The *dump* and *rdump* commands will copy all files in a file system to backup media. The *restore* and *rrestore* commands will copy files stored via *dump* to a file system. *rdump* and *rrestore* provide the same functionality as *dump* and *restore* over a network. Remote dump devices are specified as a host-device combination. The *dump* command allows for different levels of backup. A level 0 dump copies every file in the file system. A level 5 dump would copy every file that has been modified since the last dump of a lower level.

3.5 Performance Management

Performance management analyzes the output from system statistics to determine problem areas and develop solutions.

3.5.1 Berkeley Performance Monitoring Commands

The following commands are executable directly on each machine to report local status.

3.5.1.1 *vmstat*

The *vmstat* command provides information on memory usage, process status, and disk utilization.

3.5.1.2 *iostat*

The *iostat* command reports statistics related to I/O operations. Both terminal and disk I/O are included.

3.5.1.3 *netstat*

The *netstat* command displays the contents of network-related data structures. Information is provided about established connections and gateways.

4. Work in Progress

4.1 OSF RFT

The Open Software Foundation will be issuing a Request for Technology (RFT) for System Administration software from the Munich office sometime in August 1989.

4.2 FDDI

A group is forming to determine which variables are appropriate for inclusion in the MIB for FDDI.

4.3 Network Management Language

"NML is seen as a canonical interface between the network management application programmer and the MIXP (Management Information Exchange Protocol)."[5] It isolates the applications programmer from the specific MIXP being used. Extending this to system administration would enable the underlying protocol to be changed without the system administrator's programming environment to be changed.

Acknowledgements

We would like to thank the following people for providing information, support, and inspiration: Carolyn D. Councill, John Lees, Jackie Carlson, Doug Gwyn, Keith Bostic, Clifford Neuman, Mark Ozur, Martin Schoffstall, Frank Cunningham, Paul Stutler, Ted Cook, and John Bossert.

Authors' Addresses

Susanne W. Smith
Windsound Consulting
6225 137th Place SW
Edmonds, WA 98020

smith@usenix.org

John S. Quarterman
TIC
701 Brazos, Suite 500
Austin, TX 78701-3243

jsq@usenix.org

References

1. Ben-Artzi, Amatzia, "The CMOT Network Management Architecture," *ConneXions* **3** pp. 14-19 Advanced Computing Environments, (March 1989).

2. McCloghrie, Keith and Ross, Marshall T., "Network Management of TCP/IP-based internets," *ConneXions* **3** pp. 3-9 Advanced Computing Environments, (March 1989).

3. Rosenstein, Mark A., Geer, Daniel E. Jr., and Levine, Peter J., "The Athena Service Management System," *USENIX Conference Proceedings*, pp. 203-212 USENIX Association, (February 9-12, 1988).

4. Steiner, Jennifer G., Neuman, Clifford, and Schiller, Jeffrey I., "Kerberos: An Authentication Service for Open Network Systems," *USENIX Conference Proceedings*, pp. 191-202 USENIX Association, (February 9-12, 1988).

5. Warrier, Unni, "A Network Management Language," *ConneXions* **3** pp. 33-39 Advanced Computing Environments, (March 1989).

Report to EUUG and USENIX on ISO JTC1 SC22 WG15 (POSIX) Meeting

May 1-3, 1989

Dominic Dunlop
The Standard Answer Ltd.

Introduction

This is the first of a series of reports which I shall be making on the activities of Working Group 15 of Subcommittee 22 of Technical Committee 1 of the International Standards Organisation (ISO TC1/SC22/WG15). It is this group which is taking the work of the Institute of Electrical and Electronic Engineers (IEEE) on POSIX, a portable operating system interface, from its current official status as an American national standard to its final goal as an international standard. I have been sponsored by the European UNIX systems User Group (EUUG) and USENIX to attend the meetings of the working group on your behalf, representing your views and reporting back on developments which affect your interests.

Meeting Report

Hosted in Ottawa by the Standards Council of Canada, May's three day meeting of ISO TC1 / SC22 / WG15 was attended by five "technical experts" (representatives) from the USA, three from the UK, two from Denmark, and one each from Canada, France, Japan, and the Netherlands. There were three "invited experts": myself, invited by the UK delegation to represent the EUUG and USENIX; Shane McCarron, invited by the USA on behalf of UNIX International; and Mike Lambert of X/Open Company Ltd.

Mike Lambert was invited by Jim Isaak, convener of the working group, to set out X/Open's mission and its position in relation to ISO's activities. It was clear that this was necessary as, in the responses to a previous ballot on the working group's work-in-progress, several respondents effectively asked "Why are we doing this? Doesn't it duplicate the work of X/Open?" What is more, the Comitė Europėn de Normalisation (CEN – European Committee for Standardisation) is in the process of voting on a proposal from West Germany that the whole of the *X/Open Portability Guide, Third Edition, 1988* (XPG3) should become a "draft European Prestandard" – one step away from being a European standard.

X/Open's position is clear: "X/Open is not," as the preface to each XPG volume states, "a standards-setting organisation." Instead, X/Open is committed to align itself with international standards as soon as these are agreed, suggesting that its members adhere to other, less formal, national or de-facto standards only when no international standard is in place. In order that national and international standards can be arrived at in a timely manner, X/Open fully endorses the activities of organisations such as the IEEE, ANSI, and ISO, and provides resources to aid in their activities, as it has done – and continues to do – in the case of the IEEE's 1003 (POSIX) developments. Consequently, the Working Group considers that it is inappropriate for an international standards body such as CEN to align itself with the XPG; the XPG is not itself intended to be a formal standard, but rather a series of moving pointers to other standards. As such, it performs a valuable service to industry by indicating areas where more formal standardisation work should take place in the future. Each XPG pointer keeps moving until the area it addresses has become the subject of an agreed international standard. It is unlikely that CEN would tolerate such moving pointers, and would effectively freeze the XPG in its current state.

Another problem is that XPG3 specifies C, COBOL, and FORTRAN – languages covered by other European Standardisation efforts. It also calls out communications protocols, media formats, and a graphics interface (X) which may or may not overlap or conflict with other standards. It is not clear that these matters were considered before CEN moved to a vote.

Happily, well-defined mechanisms exist for communication between ISO and CEN, and "maximum alignment with ... ISO ... DP9945" is a requirement of the European Community's "order form" to CEN requesting that a POSIX-based European Standard be produced. The working group is using the channels to suggest that DP9945, and, in the near future, the draft IEEE 1003.2 standard, replace XPG3 in their deliberations.

C++ Standardisation

The issue of C++ standardisation was raised in the working group, as there was a (rather vague) feeling that object-oriented facilities are essential for future developments in operating systems, user interfaces, communications systems, and the like. WG15's parent, subcommittee 22, has responsibility for language standardisation. A resolution was drafted recommending that work be started on standardisation of an object-oriented programming language based on C. (The bulk of any such work would probably be given to ANSI, just like the work on C itself.) However, several valid objections resulted in the resolution being dropped:

- It is not clear whether the best basis for such a standard would be AT&T's C++, Stepstone's Objective C, or something else. (The issue is known to excite religious fervour.)
- It is not clear whether or not the language (whatever it is) should be constrained to be a superset of C. Such a constraint would be desirable from the point of view of compatibility, but might compromise the ideological soundness of the language. (Religion again.)
- The business of WG22 is the definition of an operating system interface. It should not concern itself with the means of implementation of an operating system which presents that interface – even if almost everything that conforms to the definition happens to be written in one particular language – C.

All this may seem to be somewhat arcane – distanced from reality. What it boils down to is that WG22 does not think it is time for international standardisation of an object-oriented C derivative. More work needs to be done by industry groupings and national standards bodies – and more users need to vote with their feet – before the terms of reference for an international standard become clear.

A Language-independent Definition of POSIX

The working group discussed the path towards a language-independent definition of POSIX, an issue which took on added urgency because the working group's decision was required in order that the IEEE could determine the initial format of its 1003.4 standard (real-time extensions to 1003.1), which moves to ballot in January, 1990. Like IEEE 1003, WG15 intends that the standards it produces should *ultimately* be expressed in a form which is independent of any particular computer language. And also like 1003, WG22 is currently drafting standards in terms of the C language. Two questions arise: how independent, and how ultimate?

IEEE 1003.1 is working towards removing C-language dependencies from Std. 1003.1-1988, but is stopping short of using a Formal Definition Language (FDL). While this precludes the automatic generation of test procedures, which would be possible were a verifiable FDL used, it is do-able in the short term. Soon enough, in fact, to allow 1003.4 to go to ballot in a language independent form. If 1003.1 were to drop its work in favour of a FDL, results would be postponed for some years, and 1003.4 would have to be defined in terms of the C language, much to the distress of the Ada community.

WG22 decided that use of a FDL was most appropriate to an international standard. Consequently, the group had to decide whether it wanted

a. to ignore 1003.1's work (which could result in 1003.1 dropping the activity);

b. to recommend that 1003.1 adopt a FDL (with a resultant gross delay); or

c. to use 1003.1's work as a basis for subsequent WG22 progress towards a formal description of POSIX interfaces.

The last option was chosen, resulting in a resolution which exhorts 1003.1 to keep up the good work. Expect 1003.4 to be language-independent.

For its part, WG22 is going to look into FDLs – a particularly esoteric subject – in more detail at its next meeting in Brussels in October. Ultimately, its standards will have three levels:

1. Formal description (verifiable, but almost incomprehensible to mere mortals);

2. Informal, but computer language-independent, commentary; and

3. Series of language bindings, which may or may not implement the whole interface. (For example, a COBOL binding might well exclude the *fork* interface.)

This should keep us busy well into the 1990s.

Security

ISO, in order to exercise adequate control of activities dispersed both geographically and in time, tries to compartmentalize as much as possible, making sure that the responsibilities of each subcommittee and working group are very well defined. However, there are certain topics which just cannot be pushed into a single compartment: internationalisation is certainly one, affecting as it does almost every aspect of information technology; security – an issue which currently has many people extremely worried – is probably another. Despite this, ISO TC1, having decided that the issue needs an identifiable home, may be convening a new working group – probably WG27 – to handle all aspects of security. (There is much vagueness here: TC1's mailing mechanism appears to have failed, with the result that nobody is sure exactly what will be voted on at its meeting in Paris later in May.)

Of course, this has WG15 worried, both in its own right, and on behalf of other groups and subcommittees affected by issues of security. (Most notable among these is SC18, which manages the burgeoning ISO protocol stack.) Consequently, a resolution has been forwarded to TC1 via SC22 saying, in effect, "We're in this together. Let's work together." The means of working together is a rapporteur group, a mechanism which exists to allow one group to monitor the activities of another. WG22 has such groups covering verification and internationalization as well as security.

Application Environment Profiles

Jim Isaak, convener of WG22, is much concerned with the issue of *functional standards for applications portability*, or *Application Environment Profiles* (AEPs). Jim chairs IEEE 1003.0, which, in effect, is stocking the shelves of a standards supermarket from which users can pick the selection (or profile) needed to allow applications of a particular type to be realised in a portable manner. (X/Open, The Open Software Foundation, and more than a few governments are doing much the same sort of thing.) One example of such a profile might satisfy the needs of applications requiring distributed database services with reliable transaction processing and high security.

Already, the IEEE has working groups which are defining AEPs: 1003.10 for supercomputing and 1003.11 for transaction processing, and Jim is engaged in selling the idea to ISO. Again, there are two questions: "Are you interested?" and "If so, what profiles do you want to specify?"

It is early yet: the issue is to be raised at Technical Study Group 1's (TSG1's) meeting in Essen, Germany, in September. (TSGs are another ISO mechanism which is brought into play to handle interdisciplinary issues.) TSG1 is developing a framework for application portability, so it should consider AEPs worth adopting. In the meantime, feedback concerning useful and desirable AEPs is solicited by IEEE 1003.0.

Adoption of IEEE's Draft 1003.2 Standard

Finally, WG15 has decided that it is time to adopt IEEE's draft 1003.2 standard, *Shell and Application Utility Interface for Computer Operating System Environments* as the basis for a corresponding international standard. A little procedural gymnastics is involved: the first SC22 meeting that could authorise such an adoption is in September, and it is not clear which draft of 1003.2 will be current at that time: if things go badly it could be draft 8; if to plan, draft 9. Also, draft international standard 9945, which corresponds to IEEE 1003.1, must be renamed to 9945.1, allowing 1003.2 to form the basis of 9943.2. It took three separate resolutions to put this particular show on the road!

Those, then, are the issues I consider important to members of EUUG and USENIX. Beyond them, there was much procedural stuff – more, for example, than at an IEEE meeting, even though WG22 is apparently quite informal by ISO standards.

Your comments are welcome; email them to *domo@sphinx.co.uk*.

Comments Please

We would like to know if you find the previous reports useful. Please send your comments to the editor (*ellie@usenix.org*).

Summary of the Board of Directors' Meeting

Short Hills, NJ, 17-18 April 1989

Attendance

Stephen C. Johnson, Rob Kolstad, Marshall Kirk McKusick, Sharon Murrel, Michael D. O'Dell, Alan G. Nemeth, John S. Quarterman, Deborah K. Scherrer.

Judith F. DesHarnais, John L. Donnelly, Neil P. Groundwater, Ellie Young.

Software Management Workshop Report

Scherrer reported that there were 80 attendees, the technical content of the papers was satisfactory, and the overall evaluation by the attendees was good.

Baltimore Conference

Program. Groundwater stated that 60 submissions were received and 22 papers had been accepted. The ACM has asked to have abstracts of the papers. Quarterman requested that problems with papers appearing elsewhere be relayed to program chairs.

Exhibits. Donnelly discussed his revised projected finances. Kolstad asked for a discussion regarding the future of exhibits – will we sell less booth space. Donnelly stated that sales in Baltimore should match San Francisco, that the vendors think we're important, and that they are concerned about location. Future site discussions should take into account having the site in a more "technical region" of the country.

Tutorials. There are 15 scheduled per day. There was general consensus that the tutorial program is driving the conferences. Student discounts have been instituted and are being advertised.

FaceSaver Proposal

The FaceSaver service proposed for Baltimore would capture new and revised faces to update the UUNET-maintained database, and not produce an attendee list as in the past. There was general agreement that it is a benefit to the membership and draws people into the exhibits. It was agreed to allocate

$12,500 to the FaceSaver proposal for the Baltimore conference. Passed: 5 in favor, 1 opposed, 2 abstained.

San Diego Report

It was reported that attendance at tutorials was high, and that the conference worked well without UniForum. There was concern that some future sites do not have as much tutorial space. While there were some comments from attendees about the absence of exhibits, there was overall enthusiasm for the box lunches and warm location.

Washington D.C. '90

Scherrer reported that the program committee has been formed. There was a lengthy discussion on how program chairs get their papers, the problems of having quality papers, the time constraints with having full papers vs. extended abstracts, and the Board's role in providing guidelines to chairs. It was agreed that the type of papers needs to be decided beforehand and the chair notified.

A committee was struck to study the problem and make proposals regarding abstracts vs. full papers and report to the Board at the next meeting.

USENIX Room at UniForum in D.C.

Because of the problems with location and cost it was decided that we will not have a USENIX room at the 1990 UniForum in D.C.

Long Range Conferences

1993 Winter Conference. DesHarnais reported on three potential sites. The Board recommended that she pursue San Diego and Disneyland and choose between the two.

1993 Summer Conference. The Board recommended that we sign a contract with Cincinnati.

Future Workshops

O'Dell suggested that we make sure that either a Board member or staff person from the Executive office attend each workshop.

Transaction Processing: Murrel reported that everything was on track, and that she would be attending part of it. Young mentioned problems with not getting all the papers from the Chair.

Systems Administration III: Kolstad reported that we're trying an experiment by offering two tutorials (by Nemeth and Kolstad) the day before the actual workshop. Standard tutorial rates will apply and he estimated that 50 people may attend each.

Distributed Systems: Kolstad reported that the paperwork from the other sponsors would be forthcoming. The co-chairs are very active and seem to have things well under control.

C++ '90: Jim Waldo of Apollo has accepted the chair.

Since there were very few responses to the posting on the net for future workshop topics, it was agreed that individual Board members should actively search for future topics.

Quarterman and Kolstad will look into a joint workshop with EUUG on Systems Administration.

Future Conference Chairs

McKusick was asked to invite Allman to be the '92 San Francisco chair; Johnson was asked to contact Mashey about his plans for Anaheim in '90; Quarterman was asked to offer Grob / Shore Dallas in '91; O'Dell was asked to offer Adams San Antonio in '92; Scherrer was offered (and she accepted) Nashville '91.

The following appointments for Board liaisons were made:

Anaheim '90	Johnson
Dallas '91	Quarterman
Nashville '91	Nemeth
San Francisco '92	McKusick
San Antonio '92	O'Dell

Alix Vasilatos was appointed Informal Programming Chair for Baltimore.

Database Report and Conference Office's Machine

O'Dell reported on the committee's meeting in Berkeley, where a dataflow model for all three offices was discussed. O'Dell said there are two problems – the long term problem of the database and the short term one of the conference office's machine. It was decided to deal with the latter immediately and to authorize $30,000 for a machine for the conference office and that the database committee authorize these expenditures.

Executive Director

It was moved that Ellie Young be appointed Executive Director. Passed unanimously.

High School Computing Funding

It was approved unanimously to authorize $3,000 to fund Don Piele's International Computer Problem Solving Contest.

Standards

Quarterman reported on the negotiations in Brussels between USENIX and EUUG regarding a joint representative to the ISO Working Group 15 POSIX committee. The two groups have hired Dominic Dunlop to be the representative. Quarterman was thanked by the Board. He also reported that USENIX is coordinating a White Paper on system administration for IEEE 1003.7, a new standards committee in this area.

Quarterman announced that McCarron would not be able to edit USENIX Standards Watchdog committee reports. There was discussion about making these reports into another publication. The Board, however, agreed that we should continue to publish these reports in *;login:* and that Quarterman should hire someone else to do the reports.

Legal Business

In a letter from our attorney Dan Appelman, the Board was advised to amend the corporation's Certificate of Incorporation to limit the personal liability of its directors, and the Board did so.

UUNET Report

O'Dell had an updated version of UUNET's finances. He reported that they have secured a line of credit, moved into new offices, and Rick Adams is now working full time as UUNET's technical director.

Relationship with EUUG and European National Groups

Murrel, as the USENIX representative at the EUUG meeting in Brussels, gave a report.

EUUG-Publications. Murrel expressed EUUG's concern about the financial arrangement with USENIX for publications and the confusion about the initial arrangements. Young was instructed to work with EUUG on this and that future sales of publications would be coordinated with Philip Peake of the EUUG.

Relationship with EUUG and the European National Groups. Murrel reported that while EUUG encourages all informal contacts between user groups, they would like to be the sole contact for EUUG and their national group members for negotiations leading to reciprocal agreements and purchases of services. There was general agreement that such a policy would be impossible since any USENIX member can order any quantity of our publications.

Nemeth and Johnson suggested an exchange of publications with JUS so that we can abstract and index them.

Proposal for Second UNIX on Supercomputers Workshop

Lori Grob's proposal that USENIX sponsor a second UNIX on Supercomputers workshop in early Fall of 1990 was approved.

Publications and Membership

Manuals. Young reported that while customer service from Howard Press has been poor in the past, she had met with them to relay our concerns as well as to check the inventory. It was recommended that we advertise the manuals as "final printing" and push their sales at the conferences.

Reprinting Proceedings. Young reported that after three postings on the net, the number of responses was still too little to warrant reprinting. We will offer out-of-print proceedings at our cost to reproduce them on a per request basis.

Executive Office Report. Young went over the report. Total membership as of 4/1/89 was

2,712, up 30% from figures of the previous year.

Budget

There was a discussion regarding future costs for the journal. Young reported that the contract with UC Press calls for fees to be somewhat lower over the next two years, and with more library subscriptions our fees would be reduced even further.

The Board was satisfied with Young's efforts to provide a cash flow model for all three offices. There were some format suggestions for the reporting of membership services, and suggestions for other models to enable the Board to plan for future growth, projects, and have a better understanding of what has happened in the past.

It was decided to pay off the First Interstate Bank loan for the Sequent machine and make arrangements with UUNET for a schedule of payments.

Membership Fees / Dues Proposal

There was discussion regarding Young's proposal sheets. The general feeling of the Board was that the Association can continue to depend on the conference surplus to fund membership benefits. It was passed unanimously that the 1990 membership dues remain:

Student:	$15
Individual:	$40
Educational:	$125
Corporate:	$275
Supporting:	$1,000

It was also approved that a subscription to the proceedings be included as a benefit to all institutional members as soon as possible.

Election Subcommittee Proposal

Murrel, speaking for the election subcommittee, proposed a Bylaw change to limit the number of consecutive terms for board members. The following wording change to the Bylaws was approved:

Replace in section 4.2:

> Any eligible person may be reelected as an officer or director one or more times.

with

> Any eligible person may be reelected as an officer or director one or more times, but may not be elected to more than four terms in succession.

And that this Bylaw change will take effect after the next election, on July 1, 1990.

–EY

;login: 14:4

UUNET Source Archives on Tape

By popular demand, UUNET Communications Services is making its collection of freely redistributable UNIX source archives available on tape to any interested parties.

UUNET has over 500 megabytes of source archives on line for UUNET subscribers to access. These archives are now available to anyone. They are distributed on two 6250 bpi ½" tapes or FIVE ¼" cartridge tapes (QIC-24, Archive 60 megabyte tapes, *i.e.*, Sun compatible). All files on the tape are compressed (except the *compress* program itself) to save space. The all inclusive cost of these tapes with prepayment is $200 for the ½" tapes or $350 for the ¼" tapes. If you require us to process a purchase order or to invoice you, add $50 for processing costs (*i.e.*, $250 for the ½" tapes or $400 for the ¼" tapes).

All sources are the latest available versions at the time the tape is written. Included on the tape are the MIT X Window System, Version 11 Release 3 plus fixes and lots of contributed software (110 megabytes); the complete *comp.sources.unix* archive (56 megabytes); the TEX text processing system (46 megabytes); all available GNU software (35 megabytes); the complete *comp.sources.games* archive (20 megabytes); the freely redistributable software from the 4.3BSD-Tahoe & Networking releases of Berkeley UNIX (17 megabytes); various networking related programs (30 megabytes); all the Internet RFCs (10 megabytes); the USENIX *Facesaver* data (60 megabytes); the *comp.std.unix* standards archives (10 megabytes); and lots more.

To obtain the tape distribution or for further information contact:

UUNET Communications Services
3110 Fairview Park Drive, Suite 570
Falls Church, VA 22042

+1 703 876 5050 (voice)
+1 703 876 5059 (fax)
uunet@uunet.uu.net

USENIX Software Distribution Tape

The 1989 USENIX software tape (the final USENIX source distribution) contains software collected for USENIX by Plus Five of St. Louis. It has just been mailed to all institutional and supporting members of the Association. The tape is in *tar* format at 6250 bpi.

Individual members of USENIX who wish to obtain a copy of the tape may request it from the Association office. The price is $60 (includes domestic postage, foreign individuals will be billed for the additional postage). It requires no AT&T nor UC license. You will be sent a requestor "Tape Release Form" which should be returned to the Association. Check, purchase orders, or payment by VISA/MC are accepted. (For charge orders please include card number, expiration date, and your signature.) Please allow 2 weeks for receipt of your order.

Scholarship Winner

The Association is pleased to announce that James N. Griffioen is the recipient of the 1989-90 USENIX Scholarship. Griffioen is a Ph.D. student studying virtual memory operating systems at Purdue University.

Executive Office Staff

Andrea Galleni has been hired as the administrative assistant for the Executive office. Andrea has been working part time for the Association during the past six months and many of you have met her at our past two conferences. She will assist the executive director in bookkeeping, publications, and the in the day-to-day business of running the Berkeley office.

Local User Groups

The Association will support local user groups by doing an initial mailing to assist the formation of a new group and publishing information on local groups in *;login:*. At least one member of the group must be a current member of the Association. Send additions and corrections to *usenix!login*.

CA – Fresno: the Central California UNIX Users Group consists of a *uucp*-based electronic mailing list to which members may post questions or information. For connection information:

Educational and governmental institutions:

Brent Auernheimer (209) 294-4373
brent@CSUFresno.edu or csufres!brent

Commercial institutions or individuals:

Gordon Crumal (209) 875-8755
csufres!gordon (209) 298-8393

CA – Los Angeles: the Los Angeles UNIX Group meets on the 3rd Thursday of each month in Redondo Beach.

Drew Bullard (213) 535-1980
ucbvax!trwrb!bullard

Marc Ries (213) 535-1980
{decvax,sdcrdcf}!trwrb!ries

CO – Boulder: the Front Range UNIX Users Group meets monthly at different sites.

Steve Gaede (303) 938-2985
NBI, Inc.
P.O. Box 9001
Boulder, CO 80301

{boulder,hao}!nbires!gaede

FL – Coral Springs:

S. Shaw McQuinn (305) 344-8686
8557 W. Sample Road
Coral Springs, FL 33065

FL – Jacksonville/Northeast: UNIX Users of Jacksonville (uujax) meets the 2nd Thursday of each month.

Tom Blakely (904) 646-2820
uflorida!unf7!tfb

Emilie Olsen (904) 390-3621

FL – Melbourne: the Space Coast UNIX Users Group meets at 8pm on the 3rd Wednesday of each month at the Florida Institute of Technology.

Bill Davis (407) 242-4449
bill@ccd.harris.com

FL – Orlando: the Central Florida UNIX Users Group meets the 3rd Thursday of each month.

Mike Geldner (305) 862-0949
codas!sunfla!mike

Ben Goldfarb (305) 275-2790
goldfarb@hcx9.ucf.edu

Mikel Manitius (305) 869-2462
{codas,attmail}!mikel

FL – Tampa Bay: the Tampa UNIX Users Group meets the 1st Thursday of each month in Largo.

Bill Hargen (813) 530-8655
uunet!pdn!hargen

George W. Leach (813) 530-2376
uunet!pdn!reggie

GA – Atlanta: meets on the 1st Monday of each month in White Hall, Emory University.

Atlanta UNIX Users Group
P.O. Box 12241
Atlanta, GA 30355-2241

Marc Merlin (404) 442-4772
Mark Landry (404) 365-8108

MI – Detroit/Ann Arbor: The SouthEastern Michigan Sun Local Users Group meets jointly with the Nameless UNIX Group on the 2nd Thursday of each month in Ann Arbor.

Steve Simmons home: (313) 426-8981
scs@lokkur.dexter.mi.us office: (313) 769-4086

K. Richard McGill
rich@sendai.ann-arbor.mi.us

Bill Bulley
web@applga.uucp

MI – Detroit/Ann Arbor: dinner meetings the 1st Wednesday of each month.

Linda Mason (313) 855-4220
michigan!/usr/group
P.O. Box 189602
Farmington Hills, MI 48018-9602

MN - Minneapolis/St. Paul: meets the 1st Wednesday of each month.

UNIX Users of Minnesota
17130 Jordan Court
Lakeville, MN 55044

Robert A. Monio (612) 895-7007
pnessutt@nis.mn.org

MO - St. Louis:

St. Louis UNIX Users Group
Plus Five Computer Services
765 Westwood, 10A
Clayton, MO 63105

Eric Kiebler (314) 725-9492
plus5!sluug

NE - Omaha: meets on the 2nd Thursday of each month.

/usr/group nebraska
P.O. Box 44112
Omaha, NE 68144

Kent Landfield (402) 291-8300
kent@ugn.uucp

New England - Northern: meets monthly at different sites.

Emily Bryant (603) 646-2999
Kiewit Computation Center
Dartmouth College
Hanover, NH 03755

David Marston (603) 883-3556
Daniel Webster College
University Drive
Nashua, NH 03063

decvax!dartvax!nneuug-contact

NJ - Princeton: the Princeton UNIX Users Group meets monthly.

Pat Parseghian (609) 452-6261
Dept. of Computer Science
Princeton University
Princeton, NJ 08544

pep@Princeton.EDU

NY - New York City:

Unigroup of New York
G.P.O. Box 1931
New York, NY 10116

Ed Taylor (212) 513-7777
{attunix,philabs}!pencom!taylor

New Zealand:

New Zealand UNIX Systems User Group
P.O. Box 13056
University of Waikato
Hamilton, New Zealand

OK - Tulsa:

Pete Rourke
$USR
7340 East 25th Place
Tulsa, OK 74129

PA - Philadelphia: the UNIX SIG of the Philadelphia Area Computer Society (PACS) meets the morning of the 3rd Saturday of each month at the Holroyd Science Building, LaSalle University.

G. Baun, UNIX SIG
c/o PACS
Box 312
La Salle University
Philadelphia, PA 19141

rutgers!{bpa,cbmvax}!
temvax!pacsbb!{gbaun,whutchi}

TX - Dallas/Fort Worth:

Dallas/Fort Worth UNIX Users Group
Seny Systems, Inc.
5327 N. Central, #320
Dallas, TX 75205

Jim Hummel (214) 522-2324

TX - San Antonio: the San Antonio UNIX Users (SATUU) meets the 3rd Thursday of each month.

Jeff Mason (512) 494-9336
Hewlett Packard
14100 San Pedro
San Antonio, TX 78232

gatech!petro!hpsatb!jeff

WA - Seattle: meets monthly.

Bill Campbell (206) 232-4164
Seattle UNIX Group Membership Information
6641 East Mercer Way
Mercer Island, WA 98040

uw-beaver!tikal!camco!bill

Washington, D.C.: meets the 1st Tuesday of each month.

Washington Area UNIX Users Group
2070 Chain Bridge Road, Suite 333
Vienna, VA 22180

Samuel Samalin (703) 448-1908

;login:

The USENIX Association Newsletter

Volume 14, Number 5 **September/October 1989**

CONTENTS

The closing date for submissions for the next issue of *;login:* is October 27, 1989

;login: 14:5

Workshop on Experiences with Distributed and Multiprocessor Systems[†]

October 5-6, 1989, Marriott Hotel, Ft. Lauderdale, FL

The goal of this workshop is to bring together individuals who have built, are building, or will soon build distributed and multiprocessor systems, especially operating systems. The workshop will feature full presentations and work-in-progress presentations on aspects of building and using these systems. The workshop will provide a forum for individuals to exchange information on their experiences, both good and bad, in designing, building, and testing their systems. This includes experiences with coding aids, languages, distributed debugging tools, prototyping, reuse of existing software, performance analysis, and lessons learned from use of such systems.

Tentative Schedule

Thursday, Oct. 5

8:30 **Opening remarks.** George Leach, Workshop Chair

8:45 **Session I: Objects and Virtual Memory**

A Distributed Implementation of the Shared Data-Object Model by Henri E. Bal, M. Frans Kaashoek and Andrew S. Tanenbaum (Vrije Universiteit, Amsterdam)

An Implementation of Distributed Shared Memory by Umakishore Ramachandran and M. Yousef A. Khalidi (Georgia Institute of Technology, Atlanta)

An Object-Oriented Implementation of Distributed Virtual Memory by Gary M. Johnston and R. H. Campbell (University of Illinois at Urbana-Champaign)

10:45 **Session II: Process Control**

Experience with Process Migration in Sprite by Fred Douglis (University of California, Berkeley)

Dynamic Server Squads in Yackos by Debra Hensgen and Raphael Finkel (University of Kentucky, Lexington)

Fine-Grain Scheduling by Henry Massalin and Calton Pu (Columbia University, New York)

1:30 **Session III: Performance Considerations**

The Parallelization of Mach/4.3BSD: Design Philosophy and Performance Analysis by Joseph Boykin and Alan Langerman (Encore Computer Corp., Marlborough)

Efficient Implementation of Modularity in RAID by Charles Koelbel, Fady Lamaa, and Bharat Bhargava (Purdue University, West Lafayette)

Making libc Suitable for use by Parallel Programs by Julie Kucera (Convex Computer Corp., Richardson)

3:30 **Session IV: Concepts**

Revolution 89 -or- Distributing UNIX Brings it Back to its Original Virtues by Francois Armand, Michel Gien, Frederic Herrmann, and Marc Rozier (Chorus Systems, En Yvelines)

[†] Sponsored by the USENIX Association and the Software Engineering Research Center (SERC), in cooperation with ACM SIGOPS and ACM SIGSOFT, and with the IEEE-CS TC on OS and IEEE-CS TC on Distributed Systems.

A Network File System Supporting Stashing by Luis L. Cova, Rafael Alonso, and Daniel Barbara (Princeton University)

4:20 Work-in-Progress presentations.

Friday, Oct. 6

8:30 **Session V: Multiprocessors**

TUMULT-64: a real-time multi-processor system by Pierre G. Jansen and Gerard J. M. Smit (University of Twente, Enschede)

Experiences with a Family of Multiprocessor Real-Time Operating Systems by Prabha Gopinath and Thomas Bihari (Philips Laboratories, Briarcliff Manor)

Implementation Issues for the Psyche Multiprocessor Operating System by Michael L. Scott, Thomas J. LeBlanc, and Brian D. Marsh (University of Rochester)

10:30 **Session VI: Tools**

Experience with P/Mothra: A Tool for Mutation Based Testing on A Hypercube by ByoungJu Choi and Aditya P. Mathur (Purdue University, West Lafayette)

Debugging and Performance Monitoring in HPC/VORX by Howard P. Katseff (AT&T Bell Laboratories, Holmdel)

CAPS – A Coding Aid used with the PASM Parallel Processing System by James E. Lumpp, Jr., Samuel A. Fineberg, Wayne G. Nation, Thomas L. Casavant, Edward C. Bronson, Howard J. Siegel, Perre H. Pero, Thomas Schwederski, and Dan C. Marinescu (Purdue University, West Lafayette)

The Implementation of Aide: A Support Environment for Distributed Object-Oriented Systems by Rodger Lea and Johnathan Walpole (University of Lancaster, Bailrigg)

1:30 **Session VII: Object-oriented Construction**

Experience With Implementing and Using An Object-Oriented, Distributed System by D. Decouchant, M. Riveill, C. Horn, and E. Finn (Bull-IMAG, Gieres)

Prototyping a distributed object-oriented OS on UNIX by Marc Shapiro (INRIA, Le Chesnay)

The Clouds Experience: Building an Object-Based Distributed Operating System by C. J. Wilkenloh, U. Ramachandran, S. Menon, R. J. LeBlanc, M. Y. A Khaldi, P. W. Hutto, P. Dasgupta, R. C. Chen, J. M. Bernabeu, W. F. Appelbe, and M. Ahamad (Georgia Institute of Technology, Atlanta)

3:30 **Session VIII: Communications, Heterogeneous Systems, and the A-word**

Experiences with Efficient Interprocess Communication in Dune by Marc F. Pucci and James Alberi (Bell Communications Research, Morristown)

Using Transputer Networks to Accelerate Communication Protocols by Horst Schaaser (Hewlett-Packard Laboratories, Bristol)

ARCADE: A Platform for Heterogeneous Distributed Operating Systems by David L. Cohn, William P. Delaney, and Karen M. Tracey (University of Notre Dame)

A Decentralized Real-Time Operating System Supporting Distributed Execution of Ada Tasks by Roger K. Shultz (Rockwell International-Collins Divisions, Cedar Rapids)

The registration fee is $225. For information contact the USENIX Conference Office at (714) 588-8649 or *judy@usenix.org*.

Professional Development Seminars

October 30, 1989, Chicago, IL

The Association is initiating a series of Professional Development Seminars in major metropolitan areas of the United States that are not currently scheduled for USENIX conferences. The seminar program is a subset of the tutorials offered at the Conferences.

The first seminar will be held in Chicago on October 30, 1989, at the Westin Hotel. Descriptions of the three tutorials to be offered follow.

Mach Overview

Avadis Tevanian, Jr., NeXT Inc.

This tutorial is intended for people who would like to find out about Mach and its internals. People interested in doing a port of Mach should find it especially useful.

This tutorial will study the Mach Operating System and Environment in detail. Emphasis will be on the Mach kernel internals, including design and implementation philosophies, virtual memory management, thread scheduling, and inter-task communication. Both machine-dependent and independent parts of the kernel will be examined, including the machine dependent interfaces that must be implemented to port Mach to a new machine. UNIX compatibility, as implemented within the Mach kernel, will also be examined.

In addition to the Mach internals, the basic mechanisms available to users will be studied, including an introduction to the basic user level services such as the Network Message Server, the Mach Interface Generator, and general Mach programming hints. The tutorial will also include discussions of the latest Mach features, future plans, and distribution.

Introduction to Programming The X Window System,[†] Version 11

Oliver Jones, Apollo Computer, Inc.

This tutorial is for experienced C programmers who are familiar with graphics workstation technology and networks but unfamiliar with Version 11 of the X Window System. People preparing to design and develop application software to run under X will find this tutorial especially useful.

The tutorial will address Xlib, the C language interface to X. By covering low level X requests, the tutorial will lay the conceptional foundation for understanding and applying the various high-level human interface toolkits and user interface management systems available as layers on X. The tutorial will provide a basis for understanding the Xtoolkit.

An Introduction to C++

Robert Murray, AT&T Bell Laboratories

This tutorial is for technical persons with a fairly complete knowledge of C. Knowledge of objected-oriented programming or data abstraction is not required.

A survey of the main features of C++ will be presented, along with some short examples of how to use the features effectively. Most use of C++ falls into one of three flavors: a better C, data abstraction, and object-oriented programming. The tutorial will examine these flavors, starting with the features and paradigms that are closest to C and progressing to the more ambitious (and potentially more powerful) features. The relationship between C++ and the draft proposed ANSI C standard will also be discussed.

For further information or a registration form contact the USENIX Tutorial Office at (303) 499-2600, FAX (303) 499-2608, or *johnd@usenix.org*.

[†] The X Window System is a trademark of M.I.T.

5th USENIX Computer Graphics Workshop

November 16-17, 1989, Doubletree Hotel, Monterey, CA

The theme of the workshop is "personal graphics." By this, we mean the use of computer graphics to aid, benefit, or amuse a single person. Generally, personal graphics applications are highly interactive, so that the user has a great deal of control over the result. Furthermore, the graphics is frequently not an end product, but is instead a communication medium between the user and computer.

The presentations in this workshop span a wide range of applications and platforms, and range from the immediately practical to the visionary. Several of the presentations will include video tapes of interactive systems, and we hope to also have some live demos. Plenty of time will be included in the schedule for interaction between attendees and speakers.

The Workshop Chair is Spencer W. Thomas, University of Michigan.

Tentative Schedule

Thursday, November 16

Opening Session

Microfabrication on the Macintosh by Carlo H. Séquin

3D Animation on the Macintosh with 3DWorks by John F. Schlag and Julian E. Gomez

The Acorn Outline Font Manager by Neil Raine, David Seal, William Stoye and Roger Wilson

Programming Systems

NeWS Classes by Owen Densmore

Visual Programming with Arachne by John Danskin and Sally N. Rosenthal

The Panel Library by David A. Tristam

Friday, November 17

Lessons learned

Learning from a Visualized Garbage Collector by Mark Weiser, Barry Hayes and Jock Mackinlay

Design Considerations for Multitasking, Windowing, Networked, Multi-platform, Distributed Applications by Ron Reisman

The Render Button by Jon H. Pittman

Views of Other Worlds

Part-Task Flight Simulation on a UNIX Graphics Workstation by Steven H. Philipson and Stefan Jeffers

The Shape of PSIBER Space: PostScript Interactive Bug Eradication Routines by Don Hopkins

Virtual Reality by Jaron Lanier

For information on registration, contact the USENIX Conference Office.

Call for Papers: USENIX C++ Conference

USENIX is pleased to host its second full C++ conference in San Francisco, California, April 9-11, 1990. We intend this conference to be of interest to a broad range of C++ users and potential users. Even if you have never written a C++ program, you will probably be able to learn enough from the tutorials to follow the technical sessions. This announcement provides early information about the dates of the events as well as persons to contact for further information. The preregistration packet containing detailed Conference information and hotel reservation information will be mailed in January, 1990.

The meeting headquarters will be the San Francisco Marriott Hotel.

Schedule of Events

Tutorials, April 9

The tutorial program is ideal for people who have been thinking about using C++ but haven't had the opportunity to learn it, as well as experienced users of and researchers in the language.

Please contact the program chair if you are interested in giving a tutorial or have a topic you would particularly like to see covered.

Technical Sessions, April 10-11

The technical sessions will cover the spectrum of work on and with C++, spanning the diversity of its users and applications, and showcasing current research and development. The technical sessions will focus on the current strengths and weaknesses of the language, show where it is and where it is going, and act as a forum for discussion of its future.

Papers are solicited on all aspects of C++, including:

Applications
Libraries
Programming environments
Case studies
New or improved implementations

Extended abstracts (no more than 2 pages) or papers (9-12 pages) must be received, either electronically (preferred) or on paper, by Friday, **January 12, 1990**. Authors will be notified of acceptance by February 5 and must submit a full paper electronically and in camera-ready form by April 9.

Queries about the technical program and all electronic submissions (*n/troff*, TEX, or PostScript preferred) or camera ready copies should be directed to:

Jim Waldo
CHR 03 DE
Apollo Computer
300 Apollo Drive
Chelmsford, MA 01824

waldo@apollo.com
decvax!apollo!waldo
(last resort) (508) 256-6600, ext. 5747

Program Committee:

Jim Waldo	Apollo Computer, chair
Andy Koenig	AT&T
James Coggins	Univ. of North Carolina Chapel Hill
Martin O'Riordan	Microsoft
Geoff Wyant	Apollo Computer
Roy Campbell	Univ. of Illinois Urbana-Champaign
Peter Canning	Hewlett Packard

Long-Term Calendar of UNIX Events[†]

1989 Oct 5-6	*Distributed Systems Workshop	Marriott Marina, Ft. Lauderdale, FL
1989 Oct 16-20	IEEE 1003	Brussels, Belgium
1989 Nov 1-3	UNIX Expo	Javits Conv. Ctr., New York, NY
1989 Nov 9	NLUUG	The Netherlands
1989 Nov 9-10	14th JUS UNIX Symposium	Osaka, Japan
1989 Nov 15	POSIX APP Workshop	NIST; Gaithersburg, MD
1989 Nov 16-17	*Graphics Workshop V	DoubleTree Hotel, Monterey, CA
1989 Nov 24	AFUU	Paris, France
1989 Dec 5-6	JUS UNIX Fair 89	Tokyo, Japan
1989 Dec 8-9	UNIX Asia '89	Sinix; Singapore
1989 Dec 11-13	UKUUG	Cardiff, Wales, UK
1989 Dec 11-15	OSI Implementors Workshop	NIST; Gaithersburg, MD
1990 Jan	UNIX in Government	Ottawa, Ont.
1990 Jan 22-26	USENIX	Omni Shoreham, Washington, DC
1990 Jan 23-26	UniForum	Washington Hilton, Washington, DC
1990 Jan 29	IEEE 1003	New Orleans, LA
1990 Mar 5-6	X3J11	New York, NY
1990 Mar 26-30	AFUU	Paris, France
1990 Apr	IEEE 1003	Montreal, Que.
1990 Apr 9	POSIX APP Workship	NIST; Gaithersburg, MD
1990 Apr 9-11	USENIX C++ Conference	San Francisco, CA
1990 Apr 23-27	EUUG	Munich, Germany
1990 May	UNIX 8x/etc	/usr/group/cdn; Toronto, Ont.
1990 Jun 11-15	USENIX	Marriott Hotel, Anaheim, CA
1990 Jun 11-13	UKUUG	London, UK
1990 Sep 11-14	AUUG	Southern Cross, Melbourne, Australia
1990 Oct 22-26	EUUG	Nice, France
1990 Nov 15	POSIX APP Workship	NIST; Gaithersburg, MD
1991 Jan 21-25	USENIX	Grand Kempinski, Dallas, TX
1991 Jan 22-25	UniForum	Infomart, Dallas, TX
1991 Feb	UNIX in Government	Ottawa, Ont.
1991 May	UNIX 8x/etc	/usr/group/cdn; Toronto, Ont.
1991 May 20-24	EUUG	Tromso, Norway
1991 Jun 10-14	USENIX	Opryland, Nashville, TN
1991 Sep 16-20	EUUG	Budapest, Hungary
1992 Jan 20-24	USENIX	Hilton Square, San Francisco, CA
1992 Jan 21-24	UniForum	Moscone Center, San Francisco, CA
1992 Spring	EUUG	Jersey, UK
1992 Jun 8-12	USENIX	Marriott, San Antonio, TX
1992 Autumn	EUUG	Amsterdam, Netherlands
1993 Jan	USENIX	Town & Country, San Diego, CA
1993 Mar 2-4	UniForum	Washington, DC
1993 Jun 21-25	USENIX	Cincinnati, OH

[†] Partly plagiarized from John S. Quarterman of TIC and Alain Williams of EUUG by EY.
* USENIX Workshops

Book Review: Programming in C++

by Stephen C. Dewhurst and Kathy Stark

(Englewood Cliffs, NJ: Prentice-Hall, 1989, ISBN 0-13-723156-3)

Reviewed by George W. Leach
uunet!pdn!reggie

If you have been looking for a book for the purpose of learning C++ that also explains the concepts behind data abstraction and object-oriented programming, then this book is for you. It is one of several new books coming out of Bell Labs concurrently with the release of version 2.0 of the C++ Translator. The authors have been involved in the development of a C++ compiler at Bell Labs for several years and have been privy to many of the design decisions made by Bjarne Stroustrup. They bring a unique perspective on C++ and insight on the philosophy behind many of the features of the language and how to effectively utilize them.

The book is organized in a progressive manner, which gradually introduces new concepts without overburdening the reader with inappropriate details. The examples are carefully chosen to reflect a progression of design that one might experience when using a language such as C++ for the first time. The book parallels Stroustrup's presentation of Object-Oriented Programming and expands upon those themes.[1] It is assumed that the reader has a background in procedural programming. It is not a requirement that the reader be well versed in C, but it wouldn't hurt.

Chapter 0, "Introduction," provides some background on C++, a brief discussion of programming paradigms, and an overview of the organization of the remainder of the book.

Chapter 1, "Data Types and Operations," will seem familiar to most C programmers, especially those who are well informed concerning ANSI C. This chapter immediately immerses the reader in the vernacular of the C++ world. For example, the concept of overloading of operators is introduced by examining arithmetic operations on the built-in types, `int` and `float`. While the overloading is a compiler and not a programmer directed activity in this context, it is a familiar concept with which the new concept may be explained. This provides the reader with a familiar point of reference for understanding a new concept that will later appear as a feature of the language. This pattern of introducing new concepts using familiar ones is repeated throughout the book.

The features new to those familiar with C introduced in this chapter include the function call style cast, user defined types (classes) by way of the ubiquitous complex number data type example, new and delete operators, and references.

Chapter 2, "Procedural Programming," begins with a cursory overview of functional decomposition and structured programming, which provides a familiar basis for discussing other programming paradigms in future chapters. A String `typedef` is utilized as an example to illustrate this style of programming. A simple program to input, sort, and output an array of Strings is presented. This example is also used to present the reader with those features of C++ that are applicable to writing programs in a procedural style. These are the features that normally are presented to support the view of C++ as a better C.[2] They include overloading, inline functions, and type checking, conversion and default initialization of function arguments and return types.

Chapter 3, "Classes," covers the language features of C++ that support the class concept. A class is the mechanism for realizing data abstraction,[3] which is further expanded upon in the next chapter. The topics covered are class types (public and private), data members, function members, operator functions, access protection of class members (public, private, protected), friend functions, initialization and conversions, and pointers to class members. A String class and a binary tree Node class are developed as examples in this chapter. These classes will be utilized in later chapters.

Chapter 4, "Data Abstraction," introduces data abstraction by examining the complex number class from Chapter 1 and the String class from Chapter 3. The key concept that is stressed is the separation of behavior, or the public interface to the abstract data type, from the implementation, which is encapsulated within the private part of the class definition. Sorted collections of integers and Strings are discussed next. This leads into the topic of generic or parameterized types. C++ does not yet support this concept.[4] The authors present a limited form of this capability utilizing the existing language features. Control abstraction is introduced using the example of an iterator for a linked list. This is an important discussion. Although many have heard of ADTs, few people realize that different forms of abstraction exist.[5] The application of control abstractions can make an ADT all that more powerful, both in isolation and in usage with other data types.

Chapter 5, "Inheritance," discusses inheritance as a means of realizing a new abstract data type from one that is almost what we want, but not quite. The specific topics of this chapter are base and derived classes, class hierarchies, virtual functions, protected members, inheritance as a design tool, inheritance for interface sharing, multiple inheritance, and virtual base classes. The linked list type from the previous chapter and the Node type introduced earlier are utilized throughout this chapter along with some more concrete examples from the problem domains of compiler and operating systems design.

Much as the discussion of classes in chapter 3 set the stage for presenting data abstraction in chapter 4, this discussion leads into the next chapter on object-oriented programming by providing a C++ context within which it may be explored.

Chapter 6, "Object-Oriented Programming," discusses the object-oriented design paradigm as an extension of data abstraction. A couple of brief examples are provided to illustrate this approach. The clean mapping of objects in a C++ program to real world objects is presented in the form of an operating system kernel's view of the world as processes and devices. It was just at this point in the book where my mind started to think about past experiences with simulation and GPSS[6] and how nice it would be to provide the same capabilities in C++. Then I turned the page to a discussion of the C++ task library and a complete airport simulation!

Chapter 7, "Storage Management," discusses the creation, destruction, and accessing of instances of objects. Constructors, destructors, and the new and delete operators are examined in greater detail than in previous encounters. Allocation and deallocation of arrays of class objects are discussed by presenting the standard C++ library functions `_vec_new` and `_vec_delete`. This is followed by a discussion of providing class-specific implementations of the new and delete operators. A powerful mechanism known as "smart" pointers is presented as a mechanism for checking as well as accessing objects that are available through indirection. And finally, some techniques for efficiently using storage when creating new objects is presented by way of realizing the copy semantics for objects of a class by applying the `X(X&)` ("X of X ref") argument to a constructor.

Chapter 8, "Libraries," presents new ways to think about libraries. First, the concept of creating an envelope around existing C libraries for access via C++ syntax is presented. Next the reader is treated to a discussion of how application specific libraries can be built to obtain the effect of a special purpose language with C++ (GPSS?). The chapter and the book are wrapped up with discussions of extensible and customizable libraries. These two sections provide the reader with an interesting side effect, the distinction between the two. Often both are thought of as being one and the same thing. However, they are not. The example provided of an extensible library is the C++ standard *streamio* library and its capability to deal with user-defined types. Libraries can be customized by the application of inheritance to arrive at the desired behavior.

The appendix provides the answers to selected exercises from each chapter of the book.

I found this book to be enjoyable and stimulating reading. Too often the discussions would lead me to think of ways of applying the features of the language to past problems that a procedural paradigm just could not deal with properly. I would find similar problem solutions in the examples following the discussions. Was this coincidence? I don't think so. The authors have done a fine job in crafting this book.

This book has emerged just as C++ is gaining popularity. As such it is filling an important niche at just the right time. The content is current with Release 2.0 of the C++ Translator from AT&T. The authors state in the Preface that they avoided details that could confuse users of different versions of C++. They further alert readers to the fact that certain new features such as multiple inheritance and refinement of the language may differ from the implementation that may be accessible to the reader. However, what is missing is an appendix of differences between versions 1.2 and 2.0 of the C++ Translator as is found in Lippman's book.[7]

I would like to acknowledge the assistance of Andrew Koenig of AT&T Bell Laboratories and Hillary Leach, my wife, in reviewing this review. Special thanks to Andrew Koenig for providing me with the C++ macro.

References

[1] Stroustrup, Bjarne, "What is 'Object-Oriented Programming"?," *Proceedings of the USENIX C++ Workshop*, Santa Fe, NM, November 9-10, 1987, pp. 159-180.

[2] Stroustrup, Bjarne, "An Overview of C++," ACM SIGPLAN Notices, 21(10), October, 1986, pp. 7-18.

[3] Stroustrup, Bjarne, "Classes: An Abstract Data Type Facility for the C Language," Computing Science Technical Report No. 84, AT&T Bell Laboratories, Murray Hill, NJ, April 3, 1980.

[4] Stroustrup, Bjarne, "Parameterized Types in C++," *Journal of Object-Oriented Programming*, January/February, 1989, pp. 5-16.

[5] Liskov, B., and Guttag, J., *Abstraction and Specification in Program Development*, MIT Press and McGraw-Hill, 1986.

[6] Bobillier, P. A., Kahan, B. C., and Probst, A. R., *Simulation with GPSS and GPSS V*, Prentice-Hall, 1976.

[7] Lippman, Stanley B., *C++ Primer*, Addison-Wesley, 1989.

Summary of the Board of Directors' Meeting

Baltimore, MD, June 11-12, 1989

Attendance

Rob Kolstad, Marshall Kirk McKusick, Sharon Murrel, Michael D. O'Dell, Alan G. Nemeth, John S. Quarterman, Deborah K. Scherrer; Ellie Young, John L. Donnelly, Neil P. Groundwater, Judith F. DesHarnais; Duncan McEwan, Dan Klein, Mike O'Brien, Dan Appelman, Donnalyn Frey, Mark Seiden, Dominic Dunlop.

Online Index/Library Update

Scherrer reported that all USENIX and related publications have been indexed and are available on UUNET, and that EUUG may be able to donate papers they have online. She would like to have *UNIX Review* indexed.

Standards

Quarterman reported that the contract with EUUG, USENIX, and Dominic Dunlop has been signed, and Dominic has completed his first "Snitch" report on the ISO JTC1 SC15 WG22 (POSIX) meeting in Ottawa in May. Quarterman has hired Jeff Haemer to edit the USENIX Standards Watchdog committee reports.

Budget

Young went over the cash flow model for the first six months. She explained that we were on target in most categories and pointed out that: income from proceedings sales was over what was budgeted for the entire year; the Software Management workshop netted $4,800; and Transaction Processing netted $4,000. While attendance at the Baltimore conference will be less than projected, the higher attendance at San Diego would balance out the shortfall somewhat and give us approximately $30,000 in additional discretionary funds for the fiscal year. McKusick felt that lower attendance at Baltimore may be due to the program. O'Dell felt that location might be a factor as well.

Current meeting

Groundwater gave a report on the speakers. Donnelly reported that the Baltimore convention center had a good floor plan, there are 66 exhibitors, 26 tutorials, and 2,000 would mostly likely attend the conference.

Executive Office Report

Young went through her report. Out of the 622 people who became members at the San Francisco conference in 1988, 31% had renewed as of May, 1989. It was decided to open up 4.3BSD manual sales to anyone who wishes to purchase them.

Transaction Processing Workshop

Murrel reported that there were good technical papers, and 102 people attended.

Future Meetings

Washington, D.C. '90. DesHarnais reported that there are not enough meeting rooms at the Shoreham, and hence only eight tutorials can be scheduled per day. She expects 1600-1800 attendees.

1993 Winter & Summer Conferences. DesHarnais has contracts for winter in San Diego and summer in Cincinnati.

Future Workshops

Systems Administration III. Kolstad reported that he and Evi Nemeth will be giving two concurrent tutorials the day before the workshop.

Distributed Systems. Kolstad reported that we had attained co-sponsorship with ACM, IEEE, and SERC.

C++ 1990. It was agreed that we could not limit the number of attendees given the attendance in the past, and Young was asked to see if Waldo would be interested in a larger format. [He was; see page 7 -EY]

New Workshop Topics

Nemeth reported that the SIGMA project in Japan is reaching its conclusion. It is a major effort within the Japan computer industry focusing on software environments in UNIX. He suggested that USENIX offer a workshop for SIGMA people to inform us on what they have done. There was discussion regarding logistics. Nemeth and Kolstad offered to do a proposal.

Scherrer suggested a workshop on Mach, and will prepare a proposal.

Young had contacted Matt Bishop, and he was interested in doing another Security workshop in 1990.

Quarterman suggested a UNIX and documentation workshop. Nemeth said it may be time for a hypertext version of the UNIX manuals. Quarterman felt this task should be discussed in a workshop and suggested using a questionnaire at the conferences to ascertain who these folks are – 1) users of documentation or 2) producers of documentation – and how do we deal with variations? O'Dell thought there was a problem of heterogeneity. Nemeth said it might be an interesting workshop and needs some reshaping. Kolstad felt it can't win. Quarterman and O'Dell will look for a person to submit a proposal.

Journal Report

O'Dell described the contents of Volume 2:2 which is an all Bell Labs issue. He stated that the papers were starting to come in better and that over 50% of the submissions were rejected. Young went over the UC Press promotion and circulation report. Nemeth congratulated O'Dell on his efforts.

Abstracts vs. Extended Papers Committee Report

Murrel reported on behalf of the committee that they had agreed that both types of papers arc important to our conferences, and that we need to decide which type for each conference. There was a lengthy discussion about past history, logistics, quality of papers, and assumptions relating to which type of paper makes a better conference.

Murrel offered the following summary: 1) that the program chair has the choice on types of papers, and 2) the Board needs to provide predictable guidelines. It was agreed that we not permit any two successive conferences to require full papers and to notify future conference chairs.

Nominating Committee Suggestion

Kolstad expressed his concerns that reports from the nominating committee can be construed by the membership as endorsements rather than slot-filling. All the Board present, except Kolstad, agreed that the nominating committee should be an endorser, and that the formal charge to the committeee is to find enough eligible people (decent candidates) to fill the slots.

EUUG and a World UNIX Users Group

Scherrer had several notes from Teus Hagen regarding the formation of a world UNIX users group, and there was a discussion regarding our joining such an organization. Quarterman stated that a world group would be good for standards and networking activities. It was agreed that if we want to do more joint activities with EUUG, it would not require a world group. Scherrer suggested we have better networking between organizations.

Sales of Books at Conferences

Quarterman expressed concern that Jim Joyce was using the Association's name in publicity for his hospitality suite at the Hyatt in Baltimore. Young was instructed to send a letter informing him that we are aware of his activities and misuse of the Association's name on the net.

Washington, D.C., Program Report

Dan Klein reported that the Call for Papers had been posted to various groups, and that he hoped to have Jim Tamayko as keynote to talk about computers and spacecraft. He thinks the conferences are losing some of the fun and suggested, among other things, having a computer game contest.

Quarterman suggested that the informal and technical program chairs coordinate between each other and report to the Board liaison. Sonya Neufer was invited to be the informal programming chair for D.C. He also mentioned that a committee had been formed to organize parallel sessions at the conferences.

Professional Development Seminars Proposal

Donnelly proposed that the Association fund two "trial" seminars in 1989. The initial format would be a one day session of three tutorials. There was discussion about speaker compensation, the format, and registration fees. The Board agreed to allocate $40,000 to be made available for the two seminars.

Speaker's Bureau

Donnelly stated that the purpose of a Speaker's Bureau would be to provide a source of speakers for educational groups who could discuss a variety of UNIX-related topics in a colloquium-type setting. It would be primarily an educational endeavor initially directed at universities, high schools, and local users groups. There was discussion regarding audiences, topics, and speakers. It was agreed to allocate $6,000 to fund a Speaker's Bureau.

Sybase Report

Mark Seiden stated that the overall problem is that the Association is running several databases which he has been consolidating. While Sybase has not yet been used in the office, he had finished the user interface and hoped to have it up and running soon.

Legal

There was a discussion with our attorney, Dan Appelman, about our exposure under Maryland law with regard to a person not affiliated with USENIX selling books at a conference and not paying sales tax. Appelman stated that we are not liable for sales tax as long as it is clear that we are not associated with that person.

Young and Appelman discussed their meeting with the Vice Chancellor of the University of Capetown (UCT) regarding their wanting a UUNET/USENIX connection. Appelman stated that the export regulation laws aren't clear regarding UCT's status under the 1986 Comprehensive Anti-Apartheid Act. Appelman felt that the Board would not have much liability if the connection were open, but that a more secure route is to wait until the regulations are changed. After discussion it was decided to do nothing at this time.

UUNET Report

Rick Adams stated that their biggest problem is not being able to grow fast enough to meet the demand. Adams was informed of the Association's desire to pay off the FIB loan and work out a direct schedule of payments with them. Adams requested an additional $20,000 loan to add more processors. After discussion, it was agreed to lend UUNET $90,000 at a variable interest rate, and that USENIX send approximately $70,000 to FIB to pay off the loan, and the balance be sent to UUNET, and that this loan be secured by UUNET's Sequent machine.

Standards – WG15 Report

Dominic Dunlop had been invited to the last working group meeting as a "Category A" liaison to monitor the group activities on behalf of users of the UNIX operating systems in Europe and North America.

Next Board Meeting

Quarterman suggested having the next Board meeting in Vienna. Since many of the group were already going to attend the EUUG Conference, most felt that having a Board meeting concurrent with an EUUG meeting would enable the two groups to have a joint meeting/reception.

–EY

An Update on UNIX and C Standards Activities

Jeffrey S. Haemer, Report Editor

USENIX Standards Watchdog Committee, August 1989

ANSI X3J11 C Language

Doug Gwyn (*gwyn@brl.mil*) reports:

There's not much new on the X3J11 (ANSI C) front.

As of about a week ago [i.e, mid-May, 1989 - jsh], X3 had not yet finished the reballoting caused by having to respond to a previously lost, public comment letter from Russell Hansberry. X3J11 discussed these comments with Hansberry at the Seattle meeting, voted on some resulting proposals, and, in summary, reaffirmed previous resolutions of and decisions about all his issues. In all, no changes were made to the December 1988 draft proposed standard and rationale documents. An official response was sent to Hansberry, who had 15 working days to respond to X3, after which X3 would again ballot on whether or not to send the proposed C standard to ANSI for ratification. Hansberry replied, requesting a full formal review process. Since this was previously approved, we expect the same outcome for the reballot, but the people involved in the appeals process are not the same as the ones with technical expertise who drew up the standard, so anything could happen. Certainly there will, at least, be a substantial delay in obtaining final approval of the submitted standard as an ANSI standard.

ISO WG14 met concurrently in Seattle. A Danish proposal for an alternative to trigraphs was defeated by both X3J11 and WG14; although one might hope that we've heard the last about this, the delay on the ANSI side might permit more hassle from the Danes. WG14 also agreed to submit the same proposed standard as ANSI's for ISO approval, with the understanding that British concerns about excessive instances of "undefined" behavior would be addressed early in the X3J11 "interpretations" phase. Specifically, the British would like all such instances clearly identified. X3J11 is working with them to prepare an "information bulletin," which would clarify the standard without forcing a revision of the proposed standard itself.

X3J11 work for the foreseeable future will concentrate on answering questions about the standard and providing rulings on interpretations.

No new instances of X3.159/1003.1 conflict have arisen, to my knowledge, since the "great 'environ' problem." There have been several varying interpretations of how vendors should define __STDC__ (if at all) in an "extended" implementation of X3.159, such as most POSIX vendors will be doing for reasons of backward compatibility. X3J11 certainly intended all positive integral values of __STDC__ to be reserved for strictly standard-conforming implementations of C; there is some disagreement whether non-positive values should be used by vendors to indicate "ANSI C except with extensions." Unfortunately there is no way to constrain non-conforming implementations via wording in the standard.

A proposal that X3J11 undertake standardization of C++ was rejected; although there was a consensus that C++ was ready for a standards effort to begin, it was not felt that C++ should be undertaken by X3J11 itself, for a variety of reasons.

Rex Jaeschke has formed a "Numerical C Extension Group," which has begun work on identifying extensions needed for C to fully serve the numerical computing community. This is not yet officially under X3 auspices, but it could become so.

The X3J11 meeting slated for September, 1989 in Salt Lake City was canceled due to the approval delay; the next scheduled meeting is in New York City on March 5-6, 1990.

IEEE 1003.5 Ada Language

Ted Baker (*tbaker@ajpo.sei.cmu.edu*) reports of the April 1989 meeting:

The Minneapolis meeting started off poorly. The chair, co-chair, and technical editor were absent, though each for good reasons. ("Co-chair" is POSIX for vice-chair.) Only one of the members present had received a copy of the latest draft (2.0). Many of the changes agreed upon at the last meeting (Fort Lauderdale) were not yet reflected in this draft. There was no agenda.

Despite these handicaps, the group made considerable progress. Steve Deller acted as chair, working up an agenda and holding the group fairly closely to it. (Indeed, Steve Deller has now become an official co-chair, but is still doing a good job.)

By the second day copies of Draft 2.0 had been made. This draft was reviewed completely, and several changes were approved. The hottest issue was how signals would be mapped to Ada task entries. Several semantic gaps in the P1003.1 C-language binding were discovered, and passed on to the P1003.1 working group.

Most major semantic issues were, at this point, resolved.

1. Each Ada program consists of a single POSIX process, or at least appears to be so through the POSIX/Ada interface.

2. POSIX signals are handled by Ada tasks via the same mechanism as hardware interrupts, as logical entry calls.

3. POSIX character and string types are distinct from the standard Ada character and string types.

4. The C-binding's "errno" values are translated into distinct Ada exceptions.

5. The Ada-binding need not follow the organizational and naming conventions of the C-binding, especially where they violate principles of data abstraction.

What remains is filling in a lot of details, including most of the text of the document, and making it stylistically consistent.

Group members volunteered to edit the agreed-upon changes into the draft document, while filling in missing text. This work was to have been completed before May 10-12, at which time a subset of the working group would meet in Bedford MA for a "writing party." Its goal would be to catch up and complete all missing portions of the draft, so that it could be submitted for mock ballot before the July P1003 meeting. There was some question whether this goal would be met. (The mock ballot date was missed, so it appears 1003.5 won't have an official Ada language binding that corresponds to 1003.1 by end-of-year 1989.)

There were also coordination meetings (BOFs) with the groups working on language-independent specifications (P1003.1) and threads (P1003.4). The Ada group seemed generally pleased with progress on the language-independent specification, and hopes that the draft Ada-binding will provide some guidance to that activity. The group is less pleased with the tendency of other groups (e.g. P1003.2 and P1003.4) to aggravate the problem of C-dependencies in their draft documents.

The Ada group is very interested in having the 1003.4 standard include multi-threaded processes, but is very concerned that any such standard be compatible with the semantics of Ada tasks. Some of the preliminary proposals coming out of the threads working group do not seem to be compatible with this goal.

IEEE 1003.8 Networking

Steve Head (*smh@hpda.hp.com*) reports on the April 1989 meeting:

Overview

P1003.8 is the IEEE POSIX networking standards committee, working on network standard interface definitions for POSIX. The committee is divided into several subcommittees, including transparent file access, remote procedure call, network IPC, and MAP. There were approximately 30 attendees at P1003.8. This is a report on the network IPC subcommittee, which is creating both a "sophisticated" interface and a "naive" interface for interprocess communications. Because it is not yet known whether the group's work will all go into a single standard, the word "standard" should probably be "standard(s)."

At the April meeting, the group redefined the goals of the two interfaces, and adopted a top-down methodology to avoid factional deadlock. It went on to set initial milestones for the end-product standards, complete a first pass of functionality and objects of interest, and initiate discussion and cooperation with other organizations and committees working in related areas.

Detail

At this meeting, the main topics of discussion were:

1. Goals
2. Methodology
3. Milestones
4. Functionality and Objects
5. Relationships to Other Organizations, Standards, and Evolving Standards
6. Naming
7. Async Events
8. XTI versus sockets
9. e-mail distribution list
10. Future Agenda

Note: in this report, "XTI" refers to X/Open's Transport Interface, a networking interface definition for UNIX based primarily on AT&T's TLI (Transport Library Interface). "CNI" refers to the Chemical Abstracts Company Network Interface, an independently developed transport level interface which is designed run not only on UNIX but other operating systems as well. "Sockets" refers to the popular, 4.3-BSD-based networking interface.

1. Goals

Several new goals were added over the week to the list of existing goals that had been developed for the sophisticated interface at the previous meetings.

- timeliness of getting the standard to the industry
- usability – the standard must be fully usable, without dangling dependencies
- quality – not repeat the "mistakes" of predecessors (XTI, sockets, and CNI)
- compatibility – preserve user investment in existing interfaces (XTI, sockets, etc.)

In review, the two interfaces share the following goals:

- ability to provide client-server support
- virtual circuit- or datagram-level service
- accommodate POSIX to non-POSIX datacomm
- support for multiple protocol suites and multiple networks in one machine
- few "system calls" per logical operation, though the naive interface will probably be less efficient than the sophisticated interface

In addition, the sophisticated interface wants:

- protocol-independent access to protocol-specific features
- sophisticated (POSIX real-time) event management of protocol interface
- provision for support of [existing] protocol-specific features
- "clean" feature availability
- integration with POSIX I/O routines (`read()` / `write()`)
- easy extensibility to future protocols
- access to network management functions, such as statistics
- access to network debugging functions, such as tracing

In contrast, the naive interface will have:

- no access to protocol specific features
- no provision for sophisticated event management
- potential support for known, existing protocols, but will not support user access to all protocol features

- less coupling to the POSIX I/O routines

Many of the new goals are relevant to both and may be formally adopted as time permits, but the committee did not have time to discuss how many of the new goals were also goals for the naive interface.

This is an issue in its own right. Part of the reason for the lack of time is the need to divide attention between the two interfaces. This halves the time one would otherwise have for any given topic. The committee hopes to overcome this problem in time by merging the two interfaces into one or by dividing the committee into subgroups to work on the two interfaces in parallel. It is too early to decide which (if either) tack to take yet; neither interface is well enough defined.

2. Methodology

Someone suggested a top-down approach, for these advantages:

- form and order in the production of the standard
- avoidance of deadlocks, such as sockets versus XTI
- cleaner final design

Favorably disposed to the suggestion, the group informally adopted it.

3. Milestones

Several official milestones were set.

• starting the draft	1989
• finishing the first draft	1990
• mock ballot	1991
• full ballot	1992

Earlier dates are possible if more working members can be found to share the expected workload. (Readers, wake up: this is your chance to pitch in and help the committee make progress!)

4. Functionality and Objects

The group presented and discussed the functionality and objects for the "naive" and "sophisticated" standards. The lists generated were rough supersets of the functionality and objects in XTI, sockets, CNI, and UNI, and are available from Steve Head (*smh@hpda.hp.com*) on request. (This has progressed to a skeleton outline Draft, as of the San Jose meeting.)

The discussions laid a framework for the next tasks before the group: to separate out specific "sophisticated" from "naive" features, and to group the functionality and objects in a quasi-language-independent way. Only after this is done will the group generate C bindings to the standard.

5. Relationships to Other Organizations

The Chair of P1003.8 made contact with the ISO committees on ISO protocols. Apparently the rumor that ISO would object to a transport-level interface on the basis that it is not entering the top of the ISO stack is unfounded. The chair found no objections among those he contacted on this issue.

Several parallel efforts at a transport standard were discussed:

- OSF
- UI
- X/Open XNET's XTI
- P1003.4 (real-time) Messages

Steve Head, acting chair of the OSF SIG on Base Communication Services / Transport Interfaces Subgroup, sketched OSF status in this area. Petr Janocek, X/Open XNET chair, described XNET status, and Kathy Bohrer, leader of the P1003.4 messages working group, gave an overview of its effort.

Holes in each of these efforts currently prevent the adoption of any of them as a standard by the group. 1003.8/IPC will address major networking-specific interface issues left unresolved by other groups, and will continue to work work on an interprocess communication standard that is usable, protocol-independent, and well-integrated with the rest of POSIX.

P1003.4 (real-time) messages were especially controversial. It came as a surprise to many group members (and, frankly, many other POSIX members) that 1003.4's charter includes "system extensions." There seems to be a general feeling that "real-time" is a misleading name, and 1003.4 may not receive adequate coverage in the balloting procedure.

The group felt that this could be a real problem for extensions that are intended to solve problems involving multiple nodes in a network. For example, though the message interface is primarily for real time and generic, messaging-application needs on a single node, it can also include operation over networks that share file systems, and enable rendezvousing using the 1003.1 file system (assuming messages are supported by POSIX Transparent File Access – which is not at all clear at this time). A file system name space is generally inadequate for general network rendezvous purposes, requiring, as it does, mounts for every possible node, special files or clone files for every possible endpoint, potentially performance- and reliability-impacting extensions to the internal file name resolution routine (e.g., `namei()` or its equivalent), the adoption of new, complex protocols to handle requests, and other considerations.

The committee also worried that several aspects of the 1003.4 messaging interface seemed redundant or inefficient.

The 1003.4 messages subgroup scheduled a joint meeting with 1003.8 in July to discuss these problems. In addition, all actively attending 1003.8 working group members were to be placed on the balloting list for the May real-time mock ballot.

6. Naming

P1003.8 is forming a "naming" subgroup which will meet in July.

This group isn't likely to solve the name resolution problem from scratch (lack of time, not inspiration) so they may continue to address it until the naming subcommittee takes over. The subgroup may decide to meet with them jointly and include them on its balloting rather than give them a problem they can't ramp up to in time for a solution. Incidentally there are many name resolution issues, not just a single problem or single interface likely to solve all problems.

7. Asynchronous Events

John Barr, the leader of the asynchronous events subgroup, presented their model of asynchronous event handling to the group. This was mostly a formality; group members had already been exposed to POSIX real-time async events handling.

Some concern was raised about `select()`. Members pointed out that the real-time draft for async events implied more syscall overhead than occurs in `select()` in BSD or `poll()` in V.3; the real-time group will resolve the issue, in possible conjunction with the supercomputing group, which gave us an interesting presentation the `listio()` routine, which can be used to fire off multiple I/O transfers operating on a list of file descriptors.

8. XTI versus sockets

The "XTI versus sockets" issue is so important to users and vendors that it couldn't be left unaddressed. Here is the official committee consensus:

> We make no decision at this time on the sophisticated interface's actual relationship to the existing socket and XTI interfaces, but it will have a flavor and functionality and granularity similar to that provided by the socket and XTI interfaces.

In other words, the group feels that there are advantages to both XTI and sockets, and that POSIX will adopt features from both, but has not yet addressed whether there will be a straightforward adoption or direct extension of either, or will take some new form. (One hopes that a new form would be a functional superset of the other two.)

The group is quite aware that there are several camps and many potentially conficting goals in this highly sensitive area. Getting XTI and socket advocates to agree on a common interface will probably be a monumental task, fraught with potential dangers and traps. Any new interface would be likely to need a clear migration path from XTI and/or sockets to minimize code changes needed for existing applications: for example, sets of macro routines or public domain layer routines published in appendices. The group is aware of the possible precedent set by POSIX 1003.1 with regard to System V and 4.2 BSD (the termios section in particular). The group will study the potential benefits and drawbacks of all identifiable options before making any decisions.

The adage that "everyone wants things to get better, but no one wants anything to change" applies here. The sophisticated interface will require some compromises. The various camps must realize the benefits of joining forces and agreeing on a common standard if the working group is to be successful in this endeavor.

9. E-mail distribution list

The group will use e-mail distribution lists to expedite work and communication between meetings. The U.C. Berkeley representatives volunteered to organize this effort and maintain the lists on their machines.

Anybody may join the list by mailing to *posix-net-ptp-request@ucbvax.berkeley.edu*.

10. Future Agenda

At the San Jose meeting, P1003.8/IPC will:

- separate the functionality and objects list into ones for the "naive" and "sophisticated" interfaces;
- obtain (from action items between meetings) a more detailed list of objects, and a first cut at grouping the functionality and objects into functions for the two interfaces, and continue work from that point;
- continue to work with P1003.4 on the issues of message interface and async events.

USENIX Software Distribution Tape

The 1989 USENIX software tape (the final USENIX source distribution) contains software collected for USENIX by Plus Five of St. Louis. It has been mailed to all institutional and supporting members of the Association. The tape is in *tar* format at 6250 bpi.

Individual members of USENIX who wish to obtain a copy of the tape may request it from the Association office. The price is $60 (includes domestic postage, foreign individuals will be billed for the additional postage). It requires no AT&T nor UC license. You will be sent a requestor "Tape Release Form" which should be returned to the Association. Check, purchase orders, or payment by VISA/MC are accepted. (For charge orders please include card number, expiration date, and your signature.) Please allow 2 weeks for receipt of your order.

Out-of-Print USENIX Proceedings Now Available

The Association has photocopied, bound sets of most of its past workshop and conference proceedings available for purchase.

CONFERENCE		COST
San Diego	1983 Winter	$28.00
Toronto	1983 Summer	32.00
Washington D.C.	1984 Winter	25.00
Salt Lake City	1984 Summer	29.00
Dallas	1985 Winter	15.00
Portland	1985 Summer	45.00
Denver	1986 Winter	25.00
Atlanta	1986 Summer	37.00
Phoenix	1987 Summer	35.00
Dallas	1988 Winter	26.00
San Francisco	1988 Summer	29.00
WORKSHOP		
Systems Admin. I	1987 Philadelphia	4.00
Systems Admin. II	1988 Monterey	8.00
Security	1988 Portland	7.00
Graphics II	1985 Monterey	7.00
NOT AVAILABLE		
Delaware Conference		
Graphics I Workshop		

The above prices include domestic postage. (Orders outside the USA should contact the Association for rates.) Prepayment is required. Check, purchase orders, or payments by VISA/MC are accepted. (For charge orders please include card number, expiration date, and your signature.) Please allow two weeks for receipt of your order. Send orders to the USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.

Reprints of individual papers from all proceedings are available for $5.00 each; contact the Association Office.

Subscription Offer to USENIX Members from the EUUG

USENIX members who wish to subscribe to the European UNIX systems User Group (EUUG) Newsletter may do so at a discounted rate of $45.00 (for four issues). Orders may be placed through the USENIX Association office. See above instructions for method of payment.

Final Printing of 4.3BSD Manuals

The 4.3BSD manuals offered by the USENIX Association[†] are now available to everyone.

The 4.3BSD manual sets are significantly different from the 4.2BSD edition. Changes include many additional documents, better quality of reproductions, as well as a new and extensive index. All manuals are printed in a photo-reduced 6″×9″ format with individually colored and labeled plastic "GBC" bindings. All documents and manual pages have been freshly typeset and all manuals have "bleed tabs" and page headers and numbers to aid in the location of individual documents and manual sections.

A new Master Index has been created. It contains cross-references to all documents and manual pages contained within the other six volumes. The index was prepared with the aid of an "intelligent" automated indexing program from Thinking Machines Corp. along with considerable human intervention from Mark Seiden. Key words, phrases and concepts are referenced by abbreviated document name and page number.

While two of the manual sets contain three separate volumes, you may only order complete sets.

The costs shown below do not include applicable taxes or handling and shipping from the printer in New Jersey, which will depend on the quantity ordered and the distance shipped. Those charges will be billed by the printer (Howard Press).

To order, return a completed "4.3BSD Manual Reproduction Authorization and Order Form" to the USENIX office along with a check or purchase order for the cost of the manuals.

Manual	Cost*
User's Manual Set (3 volumes)	$25.00/set
User's Reference Manual	
User's Supplementary Documents	
Master Index	
Programmer's Manual Set (3 volumes)	$25.00/set
Programmer's Reference Manual	
Programmer's Supplementary Documents, Volume 1	
Programmer's Supplementary Documents, Volume 2	
System Manager's Manual (1 volume)	$10.00

* Not including postage and handling or applicable taxes.

[†] Tom Ferrin of the University of California at San Francisco, a former member of the Board of Directors of the USENIX Association, has overseen the production of the 4.2 and 4.3BSD manuals.

4.3BSD Manual Reproduction Authorization and Order Form

This page may be duplicated for use as an order form

Purchase Order No.: ____________

Date: ____________

Pursuant to the copyright notice as found on the rear of the cover page of the UNIX®/32V Programmer's Manual stating that

> "Holders of a UNIX®/32V software license are permitted to copy this document, or any portion of it, as necessary for licensed use of the software, provided this copyright notice and statement of permission are included,"

I hereby appoint the USENIX Association as my agent, to act on my behalf to duplicate and provide me with such copies of the Berkeley 4.3BSD Manuals as I may request.

Signed: ______________________________

Ship to:

Name: ______________________

____

____

____

____

Phone: ______________________

Billing address, if different:

Name: ______________________

____

____

____

____

Phone: ______________________

The prices below **do not** include shipping and handling charges or state or local taxes. All payments must be in US dollars drawn on a US bank.

4.3BSD User's Manual Set (3 vols.)	______	at $25.00 each	=	$________
4.3BSD Programmer's Manual Set (3 vols.)	______	at $25.00 each	=	$________
4.3BSD System Manager's Manual (1 vol.)	______	at $10.00 each	=	$________
Total	______			$________

[ ] Purchase order enclosed; invoice required.
(Purchase orders **must** be enclosed with this order form.)

[ ] Check enclosed for the manuals: $________
(Our printer will send an invoice for the shipping and handling charges and applicable taxes.)

Send your check or purchase order with this order form to:

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710

Local User Groups

The Association will support local user groups by doing a mailing to assist the formation of a new group and publishing information on local groups in *;login:*. At least one member of the group must be a current member of the Association. Send additions and corrections to *login@usenix.org*.

CA – Fresno: the Central California UNIX Users Group consists of a *uucp*-based electronic mailing list to which members may post questions or information. For connection information:

Educational and governmental institutions:
Brent Auernheimer (209) 294-4373
brent@CSUFresno.edu or csufres!brent

Commercial institutions or individuals:
Gordon Crumal (209) 875-8755
csufres!gordon (209) 298-8393

CO – Boulder: the Front Range UNIX Users Group meets monthly at different sites.

Steve Gaede (303) 938-2985
NBI, Inc.
P.O. Box 9001
Boulder, CO 80301

{boulder,hao}!nbires!gaede

FL – Coral Springs:

S. Shaw McQuinn (305) 344-8686
8557 W. Sample Road
Coral Springs, FL 33065

FL – Fort Lauderdale/Miami: The South Florida UNIX Users Group meets the 2nd Tuesday of each month.

Tony Vincent, John McLaughlin (305) 776-7770
{sun,novavax,gould}!sunvice!tony
jmclaughlin@sun.COM

John O'Brien (305) 475-7633
gatech!uflorida!novavax!john

Don Joslyn (305) 476-6415
gatech!uflorida!novavax!rm1!don

FL – Jacksonville/Northeast: UNIX Users of Jacksonville (uujax) meets the 2nd Thursday of each month.

Tom Blakely (904) 646-2820
uflorida!unf7!tfb

Emilie Olsen (904) 390-3621

FL – Melbourne: the Space Coast UNIX Users Group meets at 8pm on the 3rd Wednesday of each month at the Florida Institute of Technology.

Bill Davis (407) 242-4449
bill@ccd.harris.com

FL – Orlando: the Central Florida UNIX Users Group meets the 3rd Thursday of each month.

Mike Geldner (305) 862-0949
codas!sunfla!mike

Ben Goldfarb (305) 275-2790
goldfarb@hcx9.ucf.edu

Mikel Manitius (305) 869-2462
{codas,attmail}!mikel

FL – Tampa Bay: the Tampa UNIX Users Group meets the 1st Thursday of each month in Largo.

Bill Hargen (813) 530-8655
uunet!pdn!hargen

George W. Leach (813) 530-2376
uunet!pdn!reggie

GA – Atlanta: meets on the 1st Monday of each month in White Hall, Emory University.

Atlanta UNIX Users Group
P.O. Box 12241
Atlanta, GA 30355-2241

Marc Merlin (404) 442-4772
Mark Landry (404) 365-8108

MI – Detroit/Ann Arbor: The SouthEastern Michigan Sun Local Users Group meets jointly with the Nameless UNIX Group on the 2nd Thursday of each month in Ann Arbor.

Steve Simmons home: (313) 426-8981
scs@lokkur.dexter.mi.us office: (313) 769-4086

K. Richard McGill
rich@sendai.ann-arbor.mi.us

Bill Bulley
web@applga.uucp

MI – Detroit/Ann Arbor: dinner meetings the 1st Wednesday of each month.

Linda Mason (313) 855-4220
michigan!/usr/group
P.O. Box 189602
Farmington Hills, MI 48018-9602

MN - Minneapolis/St. Paul: meets the 1st Wednesday of each month.

UNIX Users of Minnesota
17130 Jordan Court
Lakeville, MN 55044

Robert A. Monio (612) 895-7007
pnessutt@nis.mn.org

MO - St. Louis:

St. Louis UNIX Users Group
Plus Five Computer Services
765 Westwood, 10A
Clayton, MO 63105

Eric Kiebler (314) 725-9492
plus5!sluug

NE - Omaha: meets the 2nd Thursday of each month.

/usr/group nebraska
P.O. Box 44112
Omaha, NE 68144

Kent Landfield (402) 291-8300
kent@ugn.uucp

New England - Northern: meets monthly at different sites.

Peter Schmitt (603) 646-2999
Kiewit Computation Center
Dartmouth College
Hanover, NH 03755

decvax!dartvax!nneuug-contact

NJ - Princeton: the Princeton UNIX Users Group meets monthly.

Pat Parseghian (609) 452-6261
Dept. of Computer Science
Princeton University
Princeton, NJ 08544

pep@Princeton.EDU

NY - New York City:

Unigroup of New York
G.P.O. Box 1931
New York, NY 10116

Ed Taylor (212) 513-7777
{attunix,philabs}!pencom!taylor

New Zealand:

New Zealand UNIX Systems User Group
P.O. Box 13056
University of Waikato
Hamilton, New Zealand

OK - Tulsa:

Pete Rourke
$USR
7340 East 25th Place
Tulsa, OK 74129

PA - Philadelphia: the UNIX SIG of the Philadelphia Area Computer Society (PACS) meets the morning of the 3rd Saturday of each month.

G. Baun, UNIX SIG
c/o PACS
Box 312
La Salle University
Philadelphia, PA 19141

rutgers!{bpa,cbmvax}!
temvax!pacsbb!{gbaun,whutchi}

TX - Dallas/Fort Worth:

Dallas/Fort Worth UNIX Users Group
Seny Systems, Inc.
5327 N. Central, #320
Dallas, TX 75205

Jim Hummel (214) 522-2324

TX - Houston: the Houston UNIX Users Group (Hounix) meets the 3rd Tuesday of each month.

Hounix answering machine (713) 684-6590

Bob Marcum, president (713) 270-8124

Chuck Bentley, vice-president (713) 789-8928
chuckb@hounix.uucp

TX - San Antonio: the San Antonio UNIX Users (SATUU) meets the 3rd Thursday of each month.

Jeff Mason (512) 494-9336
Hewlett Packard
14100 San Pedro
San Antonio, TX 78232

gatech!petro!hpsatb!jeff

WA - Seattle: meets monthly.

Bill Campbell (206) 232-4164
Seattle UNIX Group Membership Information
6641 East Mercer Way
Mercer Island, WA 98040

uw-beaver!tikal!camco!bill

Washington, D.C.: meets the 1st Tuesday of each month.

Washington Area UNIX Users Group
2070 Chain Bridge Road, Suite 333
Vienna, VA 22180

Samuel Samalin (703) 448-1908

;login:

The USENIX Association Newsletter

Volume 14, Number 6 November/December 1989

CONTENTS

The closing date for submissions for the next issue of *;login:* is January 3, 1990.

USENIX THE PROFESSIONAL AND TECHNICAL
UNIX® ASSOCIATION

1990 Elections for Board of Directors

The biennial elections of the Association will be held in the Spring of 1990.

After the D.C. Conference, nominations from the membership will remain open until February 2, 1990. The procedure for nominations by the membership is a written statement of nomination signed by at least five (5) members in good standing (or five separate nominations), to be submitted to the Executive Director at the Association office, and received by noon, PST, February 2. Please include a Candidate's Statement for inclusion with the ballots as well.

Ballots will be sent to all paid-up members as of March 1, 1990, on or about March 12. Members will have until April 6 to return their ballots, in the envelopes provided, to the Association office. The results of the election will be announced at the Anaheim Conference and in the May/June issue of *;login:*.

The Board is comprised of eight directors, four of whom are "at large." The others are the President, Vice President, Secretary, and Treasurer. The balloting is preferential, with those candidates with the largest number of votes being elected. Newly elected directors will take office immediately following the Anaheim conference in June.

Report of the Nominating Committee

A nominating committee was chartered by the current Board of Directors in accordance with the By-Laws of the Association to nominate a slate of candidates for the upcoming election of Directors and Officers. The committee's charge was to ensure that there were at least as many suitable candidates nominated as there are positions on the Board. The committee solicited suggestions for nominees, interviewed all of those suggested plus several other people, and have nominated the people listed below. All of these nominees want to serve on the Board and have indicated to the committee that they have the support of both their employers and families for the time commitment involved.

There are certainly many other qualified candidates. The committee did not attempt to nominate all of the potentially good Board members, but nominated what it felt to be a good slate of candidates. Any member of the Association may be nominated by petition for any board position (see above instructions).

The candidates nominated by the committee are:

President	Steven C. Johnson, Stardent Computer
President	Marshall Kirk McKusick, University of California
Vice President	Michael D. O'Dell, Prisma, Inc.
Secretary	Rob Kolstad, Prisma, Inc.
Treasurer	Sharon Murrel, AT&T Bell Laboratories
Director	Peter Collinson, Hillside Systems
Director	Ed Gould, mt Xinu
Director	Daniel Klein, Software Engineering Institute, Carnegie Mellon University
Director	Evi Nemeth, University of Colorado
Director	Sonya D. Neufer, Canstar
Director	Barry Shein, Software Tool and Die
Director	Dave Taylor, Intuitive Systems
Director	Alix Vasilatos, Open Software Foundation

Members of the nominating committee are:

Ed Gould, mt Xinu, Chair
Tom Ferrin, Univ. of California, San Francisco
Charlie Sauer, Dell Computer
Wendy Thrash, University of Washington
Pat Wilson, Consultant
Elizabeth Zwicky, SRI International

USENIX Winter Conference Program
Omni Shoreham Hotel, Washington, D.C., January 22-26, 1990

Tutorials

Monday, January 22

UNIX on Modern Architectures
Curt F. Schimmel, Amdahl, Key Computer Labs

Creating User Interfaces with OSF/Motif
Kee Hinckley & Brian Holt, Apollo Computer, Inc.

UNIX Network Programming
Richard Stevens, Health Systems International

Introduction to 4.3BSD Internals
Thomas W. Doeppner, Jr., Brown University

UNIX System V Release 4.0 Internals - Introduction
Steve Buroff & Mike Scheer, AT&T

Mach Overview
Avadis Tevanian, Jr., NeXT, Inc.

An Introduction To C++
Robert Murray, AT&T Bell Laboratories

Introduction To Programming the X Window System,* Version 11
Oliver Jones, HP Apollo Systems Division

Tuesday, January 23

An Introduction to Object-Oriented Programming
David Taenzer, U.S. West Advanced Technologies

Open Systems Interconnection (OSI) Principles
Colin I'Anson, Hewlett Packard Laboratories

Software Contracts and Intellectual Property
Daniel Appelman, Heller, Ehrman, White & McAuliffe

Beyond 4.3BSD: Advanced Kernel Topics
Mike Karels & Marshall Kirk McKusick, University of California, Berkeley

Topics in System Administration
Rob Kolstad, Prisma Inc., &
Evi Nemeth, University of Colorado

Mach Virtual Memory Internals
Nawaf Bitar, Hewlett-Packard Company

Using C++ Effectively
Andrew Koenig, AT&T Bell Laboratories

X Toolkit Intrinsics
Paul E. Kimball, Digital Equipment Corporation

Special Note for Full Time Students: A limited number of spaces in each tutorial class have been reserved for full time students at a special fee. Please contact the Conference office for full details.

* The X Window System is a trademark of M.I.T.

Technical Conference Program

Wednesday, January 24

9:00-10:30 **Introductory Remarks**
Daniel Klein, Software Engineering Institute, CMU
Ellie Young, USENIX Association

KEYNOTE: NASA's Manned Spacecraft Computers
Jim Tomayko, Software Engineering Institute, CMU

10:30-11:00 Break

11:00-12:30 **Virtual Memory** Chair: Chet Juszczak

A Dynamic File System Inode Allocation and Reclaim Policy
Ron Barkley & T. Paul Lee, AT&T Bell Laboratories

Insuring Improved VM Performance: Some No-Fault Policies
Danny Chen, Ron Barkley, & T. Paul Lee, AT&T Bell Laboratories

An External Pager Implemented as a UNIX System V,
Release 4 Virtual File System
Dean Thomas, Unisys Corporation

12:30- 2:00 Lunch

2:00- 3:30 **Architecture & Debuggers** Chair: John Mashey

Implementing a Mach Debugger for Multithreaded Applications
Deborah L. Caswell, Hewlett Packard Company,
David L. Black, Carnegie Mellon University

pdb: A Network Oriented Symbolic Debugger
Paul Maybee, Solbourne Computer, Inc.

Some Efficient Architecture Simulation Techniques
Robert Bedichek, University of Washington

3:30- 4:00 Break

4:00- 5:30 **Applications** Chair: Susanne Smith

Software Tickerplants on UNIX
Mark Luppi, Robert Berkley, Skip Gilbrech,
Tim Hunt, & Richard Plevin, Fusion Systems Group

GENESIS and XODUS – General Purpose Neural Network Simulation Tools
John Uhley, U. S. Bhalla, M. A. Wilson, D. H. Bilitch,
M. E. Nelson, & J. M. Bower, California Institute of Technology

Keynote – A Language and Extensible Graphical Editor for Music
Tim Thompson, AT&T Bell Laboratories

Thursday, January 25

9:00-10:30 **Utilities** Chair: John Devitofranceschi

Integrated Interactive Access to Heterogeneous Distributed Services
Joel S. Emer & William E. Weihl
MIT Laboratory for Computer Science

The UNIX System Math Library, A Status Report
Joel Silverstein, Steve Sommars, & Yio-Chian Tao
AT&T Bell Laboratories

Tcl: An Embeddable Command Language
John K. Ousterhout, University of California, Berkeley

10:30-11:00 Break

11:00-12:30 **Kernel Internals** Chair: Charlie Perkins

An Event-based Fair Share Scheduler
Raymond B. Essick, Prisma, Inc.

Parallel STREAMS: a Multi-Processor Implementation
Arun Garg, Sequent Computer Systems

Implementing Berkeley Sockets in System V, Release 4
Ian Vessey & Glenn Skinner, Sun Microsystems

12:30- 2:00 Lunch

2:00- 3:30 **Networks** Chair: Alix Vasilatos

Two Network Management Tools -or- (How Many Packets Would a Packet Router Route if a Packet Router Could Route Packets?)
Jeff Okamoto & Allan Lienwand, Hewlett Packard Company

Packet Trains on NSFNET National Backbone – A Traffic Characterization
Steven A. Heimlich, University of Maryland

Pseudo-Network Drivers and Virtual Networks
Steven Bellovin, AT&T Bell Laboratories

3:30- 4:00 Break

4:00- 5:30 **Ethics in the Computer Industry** Moderator: Rob Kolstad

A panel composed of a lawyer, a CEO, an ethicist and others will discuss various questions about ethics in the computer industry.

Friday, January 26

9:00-10:30 **User Interface Management Systems** Chair: Dan Geer

The Serpent User Interface Management System
Brian Clapper, Erik Hardy, Rick Kazman, & Robert Seacord,
Software Engineering Institute

Parallel Object-Oriented UIMS with Macro and Micro Stubs
Masami Hagiya & Kouji Ohtani, Kyoto University

MTX – A Shell that Permits Dynamic Rearrangement of Process Connections and Windows
Stephen A. Uhler, Bell Communications Research

10:30-11:00 Break

11:00-12:30 **File Systems** Chair: Kirk McKusick

Using UNIX as One Component of a Lightweight Distributed Kernel for Multiprocessor File Servers
David Hitz, Guy Harris, James Lau, & Allan Schwartz,
Auspex Systems Inc.

A Highly-Parallelized Mach-based Vnode Filesystem
Alan Langerman, Joseph Boykin, Susan LoVerso, & Shashi Mangalat,
Encore Computer Corporation

Disk Scheduling Revisited
Margo Seltzer, Peter Chen, & John Ousterhout,
University of California, Berkeley

12:30- 2:00 Lunch

2:00- 4:00 **Languages & Software Engineering** Chair: Dan Klein

Postloading for Fun and Profit
Stephen C. Johnson, Ardent Computer Corporation

Multiple Site Source Reconciliation
Dodi Francisco & Lois C. Price, TRW Financial Systems, Inc.

CVS-II: Parallelizing Software Development
Brian Berliner, Prisma, Inc.

Ada and Binary UNIX Standards
Mitchell Gart, Alsys Inc.

New Concurrent Sessions

USENIX is pleased to introduce a new component to its technical conference. These experimental concurrent sessions will enable people to exchange ideas and information in a more informal atmosphere. Attendees will be free to migrate between all sessions. If there is sufficient interest, these new sessions will continue as a regular event.

Wednesday, January 24

11:00-12:30 **Regular Expressions**

Andrew Hume, AT&T Bell Laboratories

The general history of regular expressions, the best known algorithms at this time, and the history of regular expressions on UNIX will be discussed. The different types of regular expression syntaxes used by various UNIX commands (*sh*, *ed*, *lex*, *grep* etc.) will be examined and examples given of their use.

make

Andrew Hume

This talk is a tutorial for generic *make*, including macros and built-in rules. Also included are some dirty tricks and discussion of various other makes.

2:00- 3:30 **Submitting and Presenting Papers at USENIX**

This talk will give you clues on getting your paper accepted: what we look for and why we accept or reject papers, as well as offering suggestions on alternative places to submit papers. It will also cover what happens once your submission has been accepted: how you can ensure that your paper looks good in the proceedings, and hints for giving a good talk at the conference. This talk is given by a group of people who have been active in USENIX for several years.

Thursday, January 25

11:00-12:30 **Getting the Most from Support**

Mary Seabrook, UniSoft Corporation

Buying a support contract isn't enough. As a technical person, you need to learn how to use support as effectively as possible. This session describes how best to present your problem to enable your support department to find a solution. This includes some thoughts on how to detail the problem and information that may be most useful in tracking down bugs.

Surviving in Networkland

John Quarterman, Texas Internet Consulting

This is a brief overview of some of the principal networks you can reach by electronic mail from an average UNIX machine, some hints on how to do that, and some of the uses that you might want to make.

2:00- 3:30 *nawk* – **A New Version of** *awk*
Richard Stevens, Health Systems International

This talk describes the differences between *awk* and *nawk*, patterns and regular expressions, flow control, expressions, variables and functions, input/output capability, and interaction with shells.

4:00- 5:30 **Works-in-Progress Session** Chair: Clement Cole

Ten minute presentations of current work.

Friday, January 26

11:00-12:30 **Perl – A System Administration Language**
Tom Christiansen, Convex Computer Corporation

Perl is an interpreted language specifically designed for system administrators. In this talk it will be introduced and an overview of the syntax, as well as some examples of its use, will be given.

2:00- 4:00 **Works-in-Progress Session** Chair: Michelle Dominijanni

Ten minute presentations of current work.

Terminal Room at D.C. Conference

The Winter USENIX Conference will once again have a terminal room providing Internet and dialout access for attendees to touch base and read their mail. Attendees will have to pay their own long distance charges by using an AT&T, MCI, Sprint, or another phone credit card. Local calls, however, will be free!

Facilities will be available to create cartridge tapes of miscellaneous, GNU, and public domain software.

During the conference electronic mail sent to *John_Doe*@conference.usenix.org will be printed on a laser printer in the terminal room and posted on the USENIX Message Board.

Many thanks to our terminal room sponsors: AT&T, Encore/Xylogics, IBM, OSF, QMS, Sun, and Telebit.

Sonya Neufer
USENIX Terminal Room Coordinator

USENIX Association Student Attendee Grant

The Association will award a limited number of travel and accomodation grants to full-time students interested in attending the Winter USENIX Technical Conference.

Interested **full-time students** should contact the Association's Executive office (*office@usenix.org*) for an application form soon. Applications must be returned no later than January 3, 1990.

Ellie Young
Executive Director

Call for Papers: Summer 1990 USENIX Conference

Anaheim, California, June 11-15, 1990

USENIX continues to seek papers describing new and interesting work. However, the Summer 1990 Technical Conference also seeks to include papers that emphasize retrospectives, analyses of tradeoffs, and critical thinking about where we are, how we got here, and why we're here. Thus, the theme is:

Beyond Mere Data:
Perspective, Insight, Understanding.

Some sessions will follow the normal 3-paper format, with questions following each talk. In other sessions, the speakers will form a panel, following the talks, first to compare approaches, and then to take questions from the audience. In some cases, other experts may be added to the panel to broaden the discussion. Especially desirable are sessions where several important different viewpoints are represented, and proposals for such sessions are welcome.

Appropriate topics include, but are not limited to:

Software release systems
User interfaces, windowing, graphics
Compilers, debuggers, tools, run-time issues
File systems
Distributed systems
UNIX kernel approaches
Fault-tolerancy, reliability, or security
Computer architectures that stretch UNIX

We will accept full papers, but require at least an abstract and outline, in a form that gives the committee confidence in the final paper. A submission should be 2-3 typewritten pages and include the following:

1. Author names, addresses, telephone numbers and E-mail addresses.

2. Abstract: 100-300 words (half a page) to be included in the final paper.

3. Outline: 1.5-2.5 pages, giving the major headings of the paper, plus a few sentences per section that give the major points that will be covered in that section in the final paper.

The following is a sample outline, which is not necessarily appropriate for all papers, but which illustrates the important topics. The purpose of an outline should be to convince the committee that something interesting and important will be said in the final paper.

1. Introduction
   - Background.
     Introduce the problem to be solved;
     why is it important?
     Reference previous work; make sure the committee knows the wheel is not being reinvented.
2. How We Solved the Problem
   - More details on the problem and its issues.
   - Design decisions and tradeoffs, and why they were made.
   - Implementation issues.
3. Evaluation
   - Data, on performance, effort required.
   - How well does it work?
   - What would we do differently?
   - If it failed, why? and what can we learn from it?
4. Conclusion
   - Summarize the paper, emphasizing why it is important, and what was learned.
5. References
   - List at least a few key references, preferably to other people's work.

The final paper should retain the 100-300 word abstract, include illustrations (where needed), and citations to relevant literature. Only previously unpublished submissions will be considered, although "retrospective" papers may describe work done years ago. Thinly-disguised product announcements are rarely accepted. Final papers should contain 8-12 pages of single spaced typeset materials. All final papers must be submitted in a camera-ready format or electronic format (*troff*-ms if possible). Typewritten or dot-matrix output is not acceptable. For authors without access to a laser printer or typesetter, appropriate facilities will be provided by the program chair.

Please submit abstracts with outline and proposals for sessions as soon as possible, and mail one hard-copy and one electronic-copy to the addresses below. The final deadline for receipt of submissions is **February 7, 1990.** Abstracts received after this deadline will not be considered. Notification of acceptance or rejection will be made by March 9, 1990. Final camera-ready papers are due by April 17, 1990.

John R. Mashey
Anaheim USENIX Technical Program
MIPS Computer Systems
930 Arques Ave
Sunnyvale, CA 94086

Internet: anaheim@mips.com
UUCP: uunet!mips.com!anaheim

Phone: (408) 991-0253
FAX: (408) 720-9809

Please include your physical and electronic mail address in all correspondence.

Program Committee:

John R. Mashey, (Chair)
MIPS Computer Systems
Clem Cole
Cole Computer Consulting
Doug Comer
Purdue University
Tom Ferrin
Univ. of CA - San Francisco
James Gettys
Digital Equipment Corp.
Lori Grob
Chorus Systems
Douglas P. Kingston III
Morgan Stanley & Co., Inc.
Heinz Lycklama
Interactive Systems Corp.
M. Douglas McIlroy
AT&T Bell Laboratories
Joe Moran
Legato Systems, Inc.
Pat Parseghian
Princeton University
Lawrence Rosler
Hewlett Packard
Bill Shannon
Sun Microsystems, Inc.

Executive Office Staff Changes

The Association has hired Carolyn Carr to be its publications manager. She will coordinate the production of the newsletter and workshop proceedings, as well as provide advice and assistance to the Executive Director on a variety of issues and new projects. Carolyn also owns her own creative services business, which provides graphic design and marketing communications consulting and production services. We should be able to put her expertise to good use, as the Association's activities continue to grow!

Toni Veglia has been hired to replace Eeva McFeely as the new receptionist for the Berkeley office. She will be handling most of your requests, so keep those cards and letters coming.

Call for Papers: USENIX C++ Conference

USENIX is pleased to host its second full C++ conference in San Francisco, California, April 9-11, 1990. We intend this conference to be of interest to a broad range of C++ users and potential users. Even if you have never written a C++ program, you will probably be able to learn enough from the tutorials to follow the technical sessions. This announcement provides early information about the dates of the events as well as persons to contact for further information. The pre-registration packet containing detailed Conference information and hotel reservation information will be mailed in January, 1990.

The meeting headquarters will be the San Francisco Marriott Hotel.

Schedule of Events

Tutorials, April 9

The tutorial program is ideal for people who have been thinking about using C++ but haven't had the opportunity to learn it, as well as experienced users of and researchers in the language.

Please contact the program chair if you are interested in giving a tutorial or have a topic you would particularly like to see covered.

Technical Sessions, April 10-11

The technical sessions will cover the spectrum of work on and with C++, spanning the diversity of its users and applications, and showcasing current research and development. The technical sessions will focus on the current strengths and weaknesses of the language, show where it is and where it is going, and act as a forum for discussion of its future.

Papers are solicited on all aspects of C++, including:

Applications
Libraries
Programming environments
Case studies
New or improved implementations

Extended abstracts (no more than 2 pages) or papers (9-12 pages) must be received, either electronically (preferred) or on paper, by Friday, **January 12, 1990**. Authors will be notified of acceptance by February 5 and must submit a full paper electronically and in camera-ready form by April 9.

Queries about the technical program and all electronic submissions (*n/troff*, TEX, or PostScript preferred) or camera ready copies should be directed to:

Jim Waldo
CHR 03 DE
Apollo Computer
300 Apollo Drive
Chelmsford, MA 01824

waldo@apollo.com
decvax!apollo!waldo
(last resort) (508) 256-6600, ext. 5747

Program Committee:

Jim Waldo	Apollo Computer, chair
Andy Koenig	AT&T
James Coggins	Univ. of North Carolina Chapel Hill
Martin O'Riordan	Microsoft
Geoff Wyant	Apollo Computer
Roy Campbell	Univ. of Illinois Urbana-Champaign
Peter Canning	Hewlett Packard

Call for Papers: AUUG Conference and Exhibition 1990

Melbourne, Australia, September 25-28, 1990

The 1990 Conference and Exhibition of the Australian UNIX systems User Group will be held at the World Congress Centre in Melbourne. Tutorial sessions will be held on the 25th and the conference proper from the 26th to the 28th of September 1990. The conference theme is:

UNIX: the Computing Platform for the 90s

Papers are invited on topics which will interest an audience of either Research, Technical, Industry, or Commercial UNIX users. Some suggested topics are:

Future Directions	Standards
Networking	Security
Project Management	Productivity Tools
Database	System Administration
User Interfaces	Windowing Systems
Real Time Systems	Multiprocessing

Papers that describe current Work in Progress, and papers on other topics relevant to the UNIX user community are also welcome.

Authors of each paper accepted will receive ONE complimentary admission to the conference and the dinner.

AUUG will again hold a competition for the best paper by a full time student at an Australian educational institution. The prize will be an expense paid return trip from within Australia to the conference to present the winning paper. A cash prize in lieu of this may be made at the discretion of AUUG. Students should indicate with their abstract whether they wish to enter the competition. AUUG reserves the right to not award the prize if no entries of a suitable standard are forthcoming.

A special issue of the group's newsletter AUUGN containing the conference proceedings will be printed for distribution to the attendees at the conference and mailed to AUUG members who do not attend.

A 1000-2000 word extended abstract is required which describes the nature of the paper and a summary of conclusions and/or results.

Acceptance of papers will be based on the abstract and will be subject to receipt of the final paper by the due date. The Programme Committee Chair reserves the right to withhold final acceptance until the final paper is received. Abstracts and final papers should be submitted to:

John Carey
AUUG 90 Programme Committee Chair
Labtam Information Systems Pty. Ltd.
43 Malcolm Road
Braeside Victoria 3195 AUSTRALIA

Phone: +61 3 587 1444
Fax: +61 3 580 5581
Telex: LABTAM AA335500

Internet: john@labtam.oz.au
UUCP: uunet!munnari!labtam.oz!john

Final Papers should contain a 100-300 word abstract and 10-20 pages of 10 point single spaced text.

Important Dates

Receipt of Abstracts	5 Feb. 1990
Letters of Acceptance	5 Mar. 1990
Receipt of Final Papers	6 Aug. 1990
Tutorials & Conference	25-28 Sep. 1990

People wishing to present tutorials should contact:

Chris Maltby
AUUG 90 Tutorials
Softway Pty. Ltd.
79 Myrtle Street
Strawberry Hills NSW 2012 AUSTRALIA

Phone: +61 2 698 2322

All enquiries regarding registration, accommodation, and the Exhibition:

AUUG 90 Secretariat
c/o ACMS
26 Hopewell Street
Paddington NSW 2021 AUSTRALIA

Phone: +61 2 332 4622
Fax: +61 2 332 4066

;login: 14:6

Long-Term Calendar of UNIX Events[†]

1989 Dec 5-6	JUS UNIX Fair '89	Tokyo, Japan
1989 Dec 6-8	Sun Users Group Conf	Anaheim, CA
1989 Dec 8-9	UNIX Asia '89	Sinix; Singapore
1989 Dec 11-13	UKUUG	Cardiff, Wales, UK
1989 Dec 11-15	OSI Implementors Workshop	NIST; Gaithersburg, MD
1990 Jan 8-12	IEEE 1003	New Orleans, LA
1990 Jan 9-10	UNIX in Government	Ottawa, Ont.
1990 Jan 20-26	DECUS	Toronto, Ont.
1990 Jan 22-26	USENIX	Omni Shoreham, Washington, DC
1990 Jan 23-26	UniForum	Washington Convention Ctr., Washington, DC
1990 Feb 14	UKUUG Sys. Admin. Workshop	Inst. of Ed, London,UK
1990 Mar 5-6	X3J11	New York, NY
1990 Mar 26-29	DECUS	Vasteras, Sweden
1990 Mar 26-30	AFUU	Paris, France
1990 Apr 9	POSIX APP Workship	NIST; Gaithersburg, MD
1990 Apr 9-11	USENIX C++ Conference	San Francisco, CA
1990 Apr 23-27	EUUG	Munich, Germany
1990 Apr 23-27	IEEE 1003	Salt Lake City, UT
1990 May 7-11	DECUS	New Orleans, LA
1990 May 30-Jun 1	UNIX/90	/usr/group/cdn; Toronto, Ont.
1990 Jun 11-15	USENIX	Marriott Hotel, Anaheim, CA
1990 Jun 11-15	ISO WG15 (POSIX)	Paris, France
1990 Jul 9-11	15th JUS Symposium	Toyko, Japan
1990 Jul 11-13	UKUUG	London, UK
1990 Jul 16-20	IEEE 1003	Danvers, MA
1990 August	*Security	Portland, OR
1990 Autumn	*Mach	
1990 Sep 25-28	AUUG	World Congress Centre, Melbourne, Australia
1990 Oct 22-26	EUUG	Nice, France
1990 Oct 31-Nov 1	UNIX Expo	New York, NY
1990 Nov 5-9	Computer Communication Conf.	ICCC; New Delhi, India
1990 Nov 15	POSIX APP Workship	NIST; Gaithersburg, MD
1990 Nov 15-16	16th JUS Symposium	Osaka, Japan
1990 Dec 4-5	JUS UNIX Fair '90	Tokyo, Japan
1990 Dec 10-14	DECUS	Las Vegas, NV
1991 Jan 21-25	USENIX	Grand Kempinski, Dallas, TX
1991 Jan 22-25	UniForum	Infomart, Dallas, TX
1991 Feb	UNIX in Government	Ottawa, Ont.
1991 Feb 18-22	DECUS	Ottawa, Ont.
1991 May	UNIX 8x/etc	/usr/group/cdn; Toronto, Ont.
1991 May 6-10	DECUS	Atlanta, GA
1991 May 20-24	EUUG	Tromso, Norway
1991 Jun 10-14	USENIX	Opryland, Nashville, TN
1991 Sep 16-20	EUUG	Budapest, Hungary
1991 Dec 9-13	DECUS	Anaheim, CA
1992 Jan 20-24	USENIX	Hilton Square, San Francisco, CA
1992 Jan 21-24	UniForum	Moscone Center, San Francisco, CA
1992 Spring	EUUG	Jersey, UK
1992 May 4-8	DECUS	Atlanta, GA
1992 Jun 8-12	USENIX	Marriott, San Antonio, TX
1992 Autumn	EUUG	Amsterdam, Netherlands

[†] Compiled with the assistance of Alain Williams of the EUUG and Susanne Smith of Windsound Consulting.

* USENIX Workshops

USENIX Board Studies UUCP

At the recent USENIX board meeting in Vienna, USENIX and EUUG agreed to jointly study UUCP, and I have agreed to be the contact and collection point for thoughts, proposals, suggestions, and flames.

Most people would agree that UUCP has many problems. Compatible versions are not available throughout the entire UNIX community, and its penetration of non-UNIX systems is minimal. Maintaining and administering UUCP threatens the sanity of even reasonably stable individuals, and is seriously damaging to UNIX hackers. The robustness and performance of the transmission protocols is open to question. The CPU and disk load that UUCP places on the operating system can and probably should be improved. ISO and X.25 compatibility are of interest to the Europeans. The list goes on.

So what can USENIX do about this? As you recall, a similar series of discussions about Usenet led to sponsorship of the Stargate experiments and eventually establishing and spinning off the very successful UUNET service. Some of the concrete actions that we have discussed are:

- Sponsoring a public-domain re-implementation of UUCP.
- Picking up and distributing one of the existing re-implementations.
- Hiring people to make studies or specific proposals.

As Treasurer of USENIX, I naturally objected to the third of these alternatives, which is why I got stuck with doing it.

In my view, there are several things that a YACP (Yet Another Communication Protocol) program should do:

- Be able to send and receive from existing UUCP sites.
- Be sensitive to the security risks of network communication.
- Be written for today's machine memories, disks, and network traffic.
- Talk at least a few other protocols; ideally, make it easy to add new protocols through streams or dynamic linking.
- Allow administration of incoming and outgoing traffic that is both easy and helpful for the naive, and not sadistic to the full-time administrator.
- Be widely available, even for non-UNIX licensees, through some form of flexible licensing scheme.
- Be robust enough that the hackings of cretins not disrupt the network, and produce clear error messages.

From the organizational point of view, there are also some non-technical questions:

- What should we do, in detail? Can we do the work in stages?
- When we decide what to do, who does it?
- How much does it cost? How do we pay for it?
- How do we distribute the final product? On what terms?
- If distributed in source form, how do we keep people from "improving" it into incompatibility or worse?
- Is this really the way we should be spending our money?

USENIX is fortunate to have significant financial reserves, and can afford to do this project right, if we decide to do it at all. That is where you come in. We would like to hear from our members on all aspects of this project – technical, organizational, the works. Alternative projects are also gratefully accepted. Please send mail to:

scj@usenix.org

We will be discussing this project at the next board meeting in January, and hope to decide then how (or whether) to move forward.

Steve Johnson

Summary of Board of Directors' Meeting

Vienna, Austria, September 17 and 19, 1989

Attendance: Stephen Johnson, Marshall Kirk McKusick, Sharon Murrel, Michael O'Dell, Alan Nemeth, John Quarterman, Deborah Scherrer, Ellie Young, John Donnelly, Daniel Klein, Judith DesHarnais, Ernst Janich, Johan Helsingius, Rick Adams, Alain Williams, Philip Peake, Neil Todd.

Workshops

Systems Administration III. Donnelly reported that the dual track format of the tutorials was well received. Attendance figures: 219 total; 183 for tutorials; 199 technical sessions only; 20 tutorials only.

Distributed Systems. O'Dell said there were good papers, and this topic might become a recurring workshop.

C++ '90 Conference. Young reported that a program committee was formed, and the Call for Papers had been mailed.

Security '90. Nemeth would contact Matt Bishop with recommendations for format guidelines, and he volunteered to serve as board liaison.

Software Development Environments Workshop. Nemeth reported that he had received a favorable reply from the technical head of the SIGMA Project in Japan, regarding co-sponsorship of an international workshop to be held in the Fall of 1990.

D.C. '90 Conference

Klein reported that 83 submissions were received and 27 papers had been accepted. He felt that extended abstracts worked very well. A panel on ethics in the computer industry would be offered.

Quarterman wanted to advertise daycare as available in D.C. It was decided to allocate $5,000 to offer and possibly underwrite this service to interested participants.

Abolish Winter Conferences

Quarterman said he put this item on the agenda because it was perceived that there was a problem of a weak technical program at the recent conferences. Nemeth said that perhaps we've really switched to having ten conferences per year (e.g., workshops have increased). O'Dell felt that the paper quality isn't the issue, that the problem is one of drawing conclusions, and our need for strategic planning as an organization. Nemeth stated that we need risk profiles in order to ascertain loss of income and possible penalties. A subcommittee of Johnson, O'Dell, and Nemeth was formed to discuss this issue.

Professional Development Seminars

Donnelly stated that the first one would be held in Chicago and consist of three sessions. He was optimistic about enrollment.

Standards

Quarterman reported on the activity of the ISO/IEC JTC1 SC22 WG15. Johnson wanted to know why we are supporting this activity. Quarterman replied that it is an attempt to prevent standards from prohibiting innovation.

Vendor Sales at Conferences

Young went over our attorney's findings that the general policy prohibiting vendors from making sales at the conferences is unnecessary from a tax perspective, and asked if there would be another reason for prohibiting sales on the floor. It was decided to allow selling on the floor, and that Donnelly should continue to screen vendors and be the regulator of taste.

Executive Office Report

It was suggested that we post on the net the dates of the upcoming board meetings with a set of topics, and suggest that members contact board members with input.

Budget – Revised Projections for 1989

Young went over the budget which provided an overview of the current finances as of July 1989, as well as projections for where we'll stand in November. In most expense categories we had realized savings. The board had also earmarked $162,500 in discretionary funds during this fiscal year.

Proposed Budget for 1990

The board went over Young's list of assumptions in preparing the budget for next year. This led to a discussion about making adjustments in compensation, fees, and offering discounts. The speakers' compensation committee would meet again, and Young would prepare a proposal for discounts. It was also decided to leave the projected net change figure for next year in place subject to getting a risk profile. The budget was approved.

EUUG Relations

Johan Helsingius ran through the structure of the EUUG boards. Their governing board consists of two representatives, who are board members of a national group in each country (approximately 20 countries). They give direction to the executive committee, and strategic/overall planning for the EUUG as a whole. The executive group is a subcommittee of the governing board that handles the day-to-day matters and is self-elected.

EUUG Executive Board:
Neil Todd (events/tutorials)
Ernst Janich (events)
Kim-Biel Nielsen (pr)
Teus Hagen (co-chair)
Michel Gien (co-chair)
Nigel Martin (finances)
Daniel Karrenberg (networks)
Philip Peake (publications)

Co-opted members (on trial):
Johan Helsingius
Norman Hall
Francis Brower

Conference Chair Proposal

Quarterman's proposal for a model for assignment of conference chairs was approved, as follows:

1. Assign no chair to any conference without a specific item for that purpose on the agenda of the board meeting.
2. Place assignment of the chair for any conference without a chair on the agenda of the board meeting 18 months before the conference.
3. State reasons in the minutes of the board meeting when assigning a chair to a conference more than 18 months in advance.
4. Require anyone who asks to chair a conference to submit a written proposal, and assign a chair to a conference without a written proposal only in exceptional cases with reasons stated in the minutes.
5. Post (in *;login:*, on Usenet, and in posters) a request for proposals to chair a conference to coincide with the conference 24 months in advance of any conference that needs a chair.

Quarterman provided a sample proposal with various points.

It was also agreed that while the board liaison must be an actual member of the Board at the time that the proposal is accepted, and the chair appointed, he/she may continue to be liaison even after retiring from the board.

Joint Workshop – EUUG and USENIX

The EUUG representatives expressed their desire to hold a joint workshop with USENIX in Europe at a location without a national group, some time in the near future. The primary goal would be to get technical developers together to exchange ideas and bring people in that are more leading edge. It was agreed to extend to EUUG an expression of interest in a joint workshop, the topic, date and location to be established by joint subcommittees of each group, and that we allocate up to $10,000 to be expended in matching funds with EUUG in the planning and preparation for this event. Any profit or loss will be split between each group. Nominal time frame would be Fall of 1990.

Public Domain UUCP Implementation

It was decided to allocate $2,500 to pay for the cost of generating a full proposal for the implementation, management, and production of a public domain version of UUCP, and that USENIX would proceed unilaterally, but would be willing to work jointly with EUUG. Johnson volunteered to collect information.

Next Board Meeting

It will be held at the Omni Shoreham Hotel in Washington, D.C., on January 21, 1990, and continuing on January 22.

EY

Audio I/O with the NeXT Computer

Michael Hawley
NeXT Inc. / MIT Media Laboratory

ABSTRACT: The NeXT machine is the first widely available computer with a built-in microphone. It is the first with a DSP, and with high-quality audio output. As such, it helps to usher in the great age of audio-rich computing, something like the precedent set by Al Jolson for film. Like movies, applications of sound in computing will not be limited to crude "talkie" interfaces, but will grow to include sound design of all kinds. The fact that these resources are available at the lowest common denominator means that applications can be written which can rely on reasonable digital audio facilities. In this paper we will outline some of the system tools for working with audio – the Sound Kit, the Music Kit, and related code – and discuss some audio-intensive applications which are emerging.

Introduction: *Al Jolson* is to *NeXT* as *THX* is to ...?

Computing is at last moving out of the silent era and into the great age of "talkies." Glancing back at the history of cinematic technology, our work in inventing audio-rich computers today seems just as balkan as the skirmishing that went on from 1900 to 1930. In 1895, Edison introduced the *Kinetophone*, which supplied musical accompaniment for a "peep show." There was no synchronization. It flopped. Shortly after that, Leon Gaumont presented the *Chronophone* in France in 1902. The Chronophone played sync sound and picture, and in a smart entrepreneurial move, Gaumont filmed vaudeville acts as the material to bootstrap his invention. The Motion Picture Patents Company licensed the technology, but Chronophone failed because the system was expensive, insufficiently amplified, produced coarse sounds, and drifted out of sync. This was about 1913, and Edison and Gaumont were only two of dozens (*Cameraphone, Vivaphone, Synchroscope, ...*). All tried to mate the silent movie to the phonograph, and uniformly failed because of combinations of cost, amplification, bad synchronization, and lack of quality.

In 1913 Edison announced the Kinetophone again. He claimed to have solved the talkies problem. He used a giant phonograph for maximum amplification, and belts and pulleys between the projection booth and the stage to sync the phonograph with the projector: some current attempts to integrate audio in computing are not unlike this! Again the technology proved inadequate: during performances, the sound slipped out as much as 10 or 12 seconds; audiences booed the picture off the screen. Contracts were rejected, Edison's factory in West Orange burned to the ground, and that was that. In the early 20's, *Phonofilm* was invented by Lee DeForest, who also had patented the audion amplifier tube in 1907. Phonofilm was a major advance – voice was recorded *onto the film,* in sync – but DeForest was a lackluster entrepreneur, and failed to secure the key deals and patents required.

Eventually, of course, it was AT&T that succeeded, through its daughter company Western Electric, and contracts with Harry Warner and sundry other Warner Brothers. Experiments began in 1925, and flourished with the formation of *Vitaphone*, a Warner subsidiary. By 1927, Vitaphone premiered *The Jazz Singer* starring Al Jolson, and a number of other films – the first true talkies. Even though The Jazz Singer got lukewarm reviews, Jolson's songs became hits in their own right (and Jolson instantly signed a $100,000 contract for three more movies).

The point of all this is that the invention of a successful recording and reproduction system for sound in movies took place over a span of three decades and with considerable skirmishing – and that was only the pioneering work that led to the earliest talking films. There is much more to sound in movies than just speech, though, and the art form and technology have been evolving steadily since then. Jack Foley is remembered as the developer of "Foley effects" in the 1930s – human non-vocalic sound effects, like footsteps, nose crunches, eating noises – and with Star Wars, Ben Burtt launched the field of "sound design." Film music has also evolved into its own genre, employing a huge industry of musicians and composers; theater sound systems have evolved in fifty years from *Vitaphone* to *THX*.

With this in mind, think about computing, and the noises made by most computers compared to the potential experience of an audio-rich computer. We are at a point now where technology that can support decent audio processing in a general and widely used computing system is becoming available. The lesson from the past is not that we should let inventors slug it out, and wait for AT&T to solve the sound problem like they did in 1925, but rather that there are compelling reasons why general-purpose audio processing of the highest quality should be made a kernel element of computer systems. NeXT, of course, is the first ambitious example, but not the last. It is important to keep this in mind because sound will contribute enormously in shaping the personality of machines to come. What follows is an overview of the facilities packaged with the NeXT computer, hard and soft.

Overview

NeXT provides sufficient hardware and software for a wide variety of basic audio applications, from voice mail to speech synthesis, speech recognition, or sound effects design in the user interface. The software available for sound processing is C or Objective-C (an object-oriented dialect of C), and the MACH operating system provides considerable support in low-overhead scheduling and driver code.

Hardware

Voice-Quality Input

The NeXT has a bundled microphone (to be mounted in the bezel at the bottom of the monitor) and a high-impedance microphone jack. These feed into a CODEC a/d converter. The CODEC part has an anti-aliasing prefilter and generates 8012.8Hz 8-bit mu-law coded input – that is, about 8,000 bytes per second for telephone quality speech input. The mu-law coding provides a 12-bit effective dynamic range compressed to 8 bits. I/O is interfaced through DMA implemented in a custom gate array.

High-Quality Sound Output

The stereo D/A converter operates at 44.1KHz in each channel with 16-bit linear quantization, just like a commercial CD player. A 1KHz maximum-amplitude sinusoid played through the DAC generates a 2V RMS signal at the audio jack. The converter includes deglitching and anti-aliasing filters. A speaker is built into the base of the monitor and provides surprisingly good sound. Additionally, stereo headphone (mini) jacks and a pair of gold-plated RCA stereo audio jacks are accessible in the back of the monitor for high-fidelity. Cheap sound (i.e., 8KHz or anything else lower than 44.1KHz) is of course interpolated up to 44.1KHz to feed the converters.

High-Quality Sound Input

There is none. With present technology (and cost) it does not make sense to force this into the lowest-common-denominator machine. However, it is easy to feed high quality data directly into the DSP port. There are already relatively inexpensive third-party products (around $600) that make it easy to flow analog and digital audio at high sampling rates directly into the machine.

DSP

The digital signal processor comprises a Motorola DSP56001 running at 25MHz; memory-mapped DMA access at 5Mbytes/second to the host interface; 8K 24-bit words of zero-wait-state RAM (local to the DSP); and a D-15 connector providing access to the DSP's SSI and SSC ports.

The DSP executes 12.5 million instructions per second and, in a single instruction, can perform a 24x24 bit multiply, a 48+56 bit addition, two parallel data moves, an instruction fetch, and two index updates. 24-bit data paths are well suited to high-quality audio processing. The DMA access (the interface between host processor and DSP) provides access to the eight byte registers of the DSP. For more information about the DSP chip, read the documentation from NeXT or Motorola.

Software

Overview

For the purposes of this discussion, software can be divided into four main components: DSP-related software (e.g., driver and other direct DSP support), the Sound Kit, the Music Kit, and application-level code.

Sound Kit

The Sound Kit is a library for accessing basic sound capabilities in the NeXT computer. Like the other major software "Kits," the Sound Kit is object-oriented, which, among other things, facilitates the handling of various formats of sound. Casual use of sound is easy:

```
id nyuk = [Sound newFromSoundfile:
          ''ThreeStooges.snd''];
[nyuk play];
```

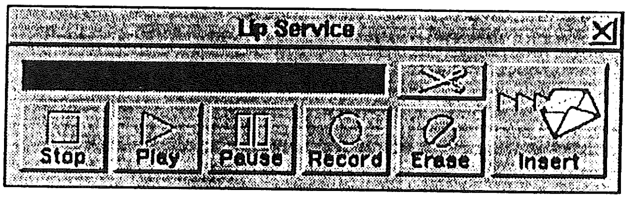
The Sound Kit also makes possible detailed, nitty-gritty access to sound in all its formats. It manages playing, recording, reading, writing, copying, etc., and makes much use of operating system primitives for virtual memory management, interprocess communication, and thread-level scheduling to efficiently process sound. For example, to record sound into a Sound object, send the object a `record` message. When the message is received, a thread (a lightweight process) is activated to fetch and store the samples, typically reading from the CODEC. This typically happens asynchronously, so that the calling process can continue doing other things (like display management, as when implementing bouncing VU meters). When sound input finishes, a message can be sent to the parent in a similar way.

Formats

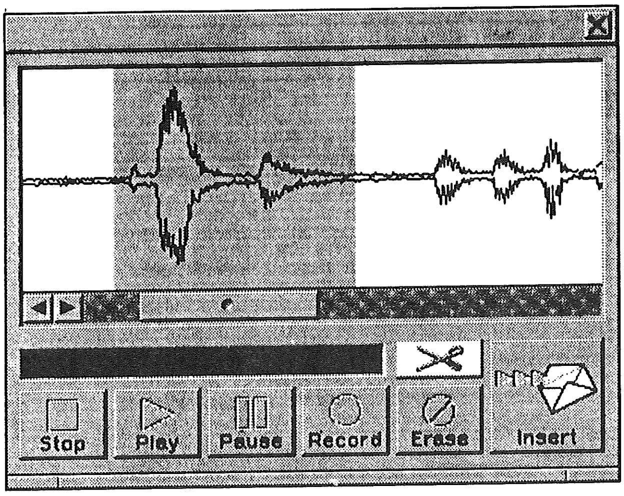
A variety of formats are supported, from low-quality to high, mono, stereo, etc. The Sound object uses the DSP for run-time format conversion of sampled sounds, which takes some of the load off the main CPU. The Sound Kit also supports DSP sound synthesis instructions – sounds which are described not by lists of samples, but by DSP algorithms and data streams. In any event, `[sound play]` and similar Kit routines work transparently. In theory, sounds may be multi-channel, but in practice the processor and disk bandwidths won't sustain more than about two channels of 44.1KHz stereo. (In fact, the optical disk does not write sufficiently fast to permit stereo recording in realtime at this rate; Ethernet barely sustains speech).

Views

The SoundView Class provides some display facilities that are compatible with the rest of the NeXT user interface conventions. The SoundView can draw, scale, select, scroll, etc. Sound is, at the moment, typically displayed as a waveform or amplitude trace, but other display methods can easily be applied. The Application Kit and Interface Builder (user interface construction tools) make it possible to stitch together sounds, views, and other interface objects easily. Associating an arbitrary sound effect to a button click (say) is simply a matter of dragging a sound file onto the button. These are the building blocks that are the foundation for other applications. For instance, the NeXT Mail program supports a simple form of voice mail, which looks like this:



The horizontal black bar holds a peak meter which bounces when you speak. Pressing the scissors button flips open an editable SoundView, which lets you scroll and edit the sound, as shown in the illustration on the next page.



Music Kit

The Music Kit provides library access for building music applications. Support for music representation, performance, and DSP-based synthesis and processing are all available. The general design emphasis integrates the gesture-level of control (e.g., MIDI and similar control-level encodings) with the low-level timbral control made possible by academic sound synthesis systems (like MUSIC-5), and cultivate it all in a rich application system. Computer music is a fruitful application area since it demands not only a mixture of technology, art, and aesthetics, but also (unlike most speech work, say) really pushes issues of attention and beauty. The speech community has had to invent a special pigeonhole for research to make digital speech captivating - prosody - and it is too often neglected in general. In music, a lousy-sounding piece may be either a failure, or deliberate, but in any event, the question of producing compelling or evocative sound is central. Moreover, the demands of musical research are typically not as narrow as those of speech. Together with sound effects, mixing, and processing, these are the main streams of flow required for audio rich computer of cinematic quality.

Like the Sound Kit, the Music Kit controls instrument generators in the DSP, but in a way that is more general than commercially packaged music synthesizers. Music is represented as a hierarchy of Score, Part, and Note objects. We will not discuss the Music Kit's elements in deep detail (one can read the copious NeXT documentation, or *Sound and Music on the NeXT Computer*, by Smith, Jaffe, and Boynton, AES 1989). What is of interest here, though, is the fact that the Music Kit manages general-purpose code for controlling the DSP. *Unit Generators* and *Synth Data* elements are the basic algorithmic building blocks for audio networks. They are typically expressed as little algorithms of calls to 56000 assembly code macros. *Synth Patches* are networks of these; and *Synth Instruments* are renderers that play notes by assigning them to instances of *Synth Patches* - this is to say, there is considerable software support for writing, loading, and scheduling networks of signal processing elements. Arching over all of this is an Orchestra class which oversees all the instrument processing done in the DSP. Given a general setup such as this, it is easy to exceed the realtime bandwidth of the DSP, so the Music Kit makes it possible to generate compute-intensive sound files out of realtime when necessary, without loss of generality.

DSP Software

The DSP software presently falls into two categories - Music Kit support and array processing support (and consequently, driver-level code for setting up the DSP to manage computation like this). Over time, specialized support for speech, signal processing, etc, will certainly evolve. The Monitor for the Music Kit implements things like DMA support, buffering of sound, unit generators (which are DSP programs) and other things needed by the music kit. Unit generators include components like adders, multipliers, allpass filters, basic oscillators, delays, etc. The array processing software includes various vector and array function macros, like FFTs, digital filters, etc. A program called dspwrap translates a DSP macro to a C callable function (that is, the host program is given a hook to call corresponding code in the DSP). There is a substantial body of code for supporting host/DSP communication via interrupts, messages, FIFOs, etc.

Conclusions

At the moment, most computers are like silent movies, and the audio channel is virtually unused. There is a *PostScript* for graphics, but not for sound. NeXT is the first computer to provide a facility for fairly general-purpose sound I/O, and even before the 1.0 release, has already shown applications like voice mail, CD-quality storage and playback, speech recognition (the *Sphinx* project at CMU has been ported to the NeXT machine; the printer can talk when it runs out of paper), sound effects in the interface (e.g., physical simulations of Billiards or Cessna flights including sound effects), real-time FFT and scope displays, etc. Certainly over the coming year or two, the NeXT will begin to recognize its owner's voice (or gender), and respond to simple spoken menu commands, but uses of audio in computers go far beyond simple speech processing and will eventually recapitulate many of the developments in cinema. In ten or twenty years, we think using a computer without sound will be like experiencing *Star Wars* without a soundtrack, so computer systems need to be designed with appropriate generality in mind.

Acknowledgements

Julius Smith, Gregg Kellogg, David Jaffe, Dana Massie, and Lee Boynton are the chief architects of the sound, music, and DSP code. Michael McNabb and Roger Dannenberg provided numerous features and design insights in the Music Kit; Doug Keislar, Doug Fulton and Richard Crandall created documentation and demos of the audio software. It was Bud Tribble who sparked NeXT into putting the DSP56001 in every machine, and finally, Steve Jobs who had the sense of vision and commitment to make general-purpose high-quality audio a key building block in the NeXT computer.

References

More information on the specifics of NeXT audio software can be found in *Sound and Music on the NeXT Computer*, by Smith, Jaffe, and Boynton, AES 1989, and in the documentation provided by NeXT. Motorola provides documentation for the DSP. One of many good books on sound in movies is *Film Sound: Theory and Practice*, edited by Elisabeth Weis and John Belton, Columbia University Press, 1985.

Book Review:

!%@:: A Directory of Electronic Mail Addressing and Networks

by Donnalyn Frey and Rick Adams

($26.95; O'Reilly and Associates, Sebastopol, CA, 1989)

Reviewed by Peter H. Salus
Open Software Foundation

How many times each day does one get an email message bounced?

Where can one look for information on the myriad electronic mail networks around the world?

Is there a way to stop your postmaster from going mad?

The answer to the first question may not be accessible in this ungainly, invaluable book; Frey and Adams have (in just under 300 pages) answered the others. If you don't want to read this review, here's the bottom line: If you use electronic mail outside of your own site, **buy this book**. It will redeem its cost in but a few days.

In fact, together with J. S. Quarterman's *The Matrix*, which is complementary to Frey and Adams, *!%@:* will yield a genuine understanding of both the ways in which email works and the links among the global networks.

Frey and Adams [F&A] – not exactly strangers to the UNIX or the USENIX communities – have organized their handbook in a sensible way:

Chapter 1. "A User Introduction to Electronic Mail," serves as a first-rate tutorial to message formats and addressing. If you ever wanted to understand the differences between @ and % or @ and !, here you are. F&A not only explicate the various addressing formats, they expound clearly and concisely on the nature of local names, mailboxes, and domains. They even manage (p. 11) to be light-hearted about the British (and New Zealand) peculiarity of writing their mail addresses "backwards." There are a few overly friendly footnotes, e.g. the explanation of the "happy-face," but these are bearable.

Chapter 2. "Networks," comprises over two-thirds of the book (pp. 23-231). It lists (in alphabetical order) all the nets I've ever heard of – and a number I'd never heard of before. [Actually, ATTMail is missing. When I asked Frey about this, I was told that they had never responded to (repeated) requests for information. *Tant pis.*] If you need information on NorthWestNet – the Northwestern States Network, with nodes in Alaska, Idaho, North Dakota, Oregon, and Washington ("a ring network with a satellite link to the Alaska site ... [maintaining] a satellite link from Oregon State University to NCAR in Boulder, Colorado, USA"), or on ILAN – the Israeli Academic Network – where the contact person is "Avi Cohen, Director, InterUniversity Computer Center, Tel Aviv University ...", it's here. So far as I can tell, the information is accurate. It is concisely presented and there are maps to go with every network. Oh boy! There are misprints, but a month's use of F&A disclosed none that wasn't self-correcting.

Appendices. There are five appendices and a glossary of terms. Second level domains and ISO codes are covered in Appendices A through D (pp. 233-265); Internet Address handling is covered in Appendix E (pp. 265-268). I suppose that one should quibble with definitions and explanations in the glossary, I find it hard to do so.

The volume concludes with two indices: by name or type of network and by notation [=abbreviation] of network.

!%@: is useful, well-organized and complete. F&A have done the entire user community a tremendous service in producing this volume; Tim O'Reilly deserves a vote of thanks for publishing and distributing it. Buy one today!

Report to EUUG and USENIX on ISO/IEC JTC1/SC22/WG15 (POSIX) Meeting

October 11-13, 1989

Dominic Dunlop
The Standard Answer Ltd.

Introduction

Working Group 15 of Subcommittee 22 of Joint Technical Committee 1 of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC22/WG15) met in Brussels, Belgium, from October 11-13 in order to further the POSIX standardization effort. I was present at the meeting as an observer with the brief of reporting back to you. This report is the second jointly commissioned by the European UNIX systems User Group (EUUG) and USENIX. If you have any comments, or need clarification or further information, please contact me at the mail address above.

First, a summary of the most important aspects of the meeting.

Summary

- The big news is that the working group has recommended that ISO accepts the POSIX operating system interface in its current form as international standard (IS) 9945-1. Assuming that this recommendation is accepted, an international standard which is identical to IEEE Std 1003.1-1988 should be registered by ISO in the next few months.

- During the balloting of the standard at the international level, a number of comments were raised. These will be addressed by the production of a revised International Standard on short order – by next June according to the current schedule. The result will be a version of POSIX in which known problems are fixed, but which is not extended in any way.

- Extensions, such as real-time facilities, transparent network file access, and security features will be added in future releases of the international standard.

- The cooperation of the IEEE POSIX project in producing standards which are acceptable to ISO and to its members is critical to the timely production of ISO standards. Steps were taken to make sure that IEEE documents are produced in a format that is acceptable to ISO, and that IEEE work on the revision of its 1003.1 standard is synchronized with the work of the ISO working group.

- Draft 9 of IEEE 1003.2, the proposed IEEE shell and utilities standard, has been accepted as Draft Proposal (DP) 9945-2. This means that the movement towards an international standard in this area is now officially under way.

- The problems raised by the suggested adoption of the whole of issue 3 of the X/Open Portability Guide as a European prestandard (see report on May, 1989 meeting) seem to have receded: European alignment with a number of formal international standards is finding acceptance as a viable and more useful alternative.

- The working group has set up "rapporteur groups" on conformance testing, internationalization, and security in order to ensure that future international standards for POSIX take account of the developments in, and of the requirements of, these important areas.

- The next meeting of the working group does not take place until June, 1990. Making a virtue of necessity, the group hopes to achieve much before that time.

POSIX as an International Standard

The international ballot period for Draft International Standard (DIS) 9945, *Portable operating system interface for computer environments*, closed at the beginning of September. The DIS is identical to draft 13 of the

IEEE 1003.1 POSIX standard, which in turn is identical, except in details of layout, to Std 1003.1-1988 published by the IEEE.

Of 26 national standards bodies entitled to vote, 19 approved the standard, one (South Africa) abstained, and one (Japan) voted against. (The five remaining countries did not vote.) Broadly speaking, ISO rules require only 75% of those voting to vote in favor in order that a standard is accepted. Where there are only one or two votes against, as in this case, the situation is even more clear-cut. Nevertheless, ISO rules require the technical committee responsible for the standard to show that it has considered the concerns of the objectors, even if it has decided not to address them by amending the draft standard.

Japan's major worry was simply that the document did not *look* like an International Standard – a matter on which France, despite voting in favor, and ISO's Central Secretariat, had also voiced concern. Instead, DIS 9945 looks like what it is – the draft of an IEEE standard – and may consequently be difficult to navigate for those used to ISO's standard format for standards.

This editorial issue could be handled simply by instructing ISO's Central Secretariat to re-enter the document text, and set it in the required format. This would take perhaps a year, and would not address the large number of "non-normative" changes already known to be required in the document as a result of work done by the IEEE over the past year. These changes are currently under discussion within the IEEE as P1003.1a. They are thought not to affect substance of the standard, merely clarifying it, fixing a number of small errors, and adding standard C function prototypes. However, ISO procedures sensibly require that any change to a draft standard must result in a new vote on the amended document, and consequently a further delay to the acceptance of a final standard.

Judging that it was more important to get a POSIX standard out in the field as soon as possible, rather than to ensure that its format and content was perfect in every way, the working group decided on a two step process:

1. Recommend that DIS 9945 is accepted in its current form as IS 9945-1. (The request to split the POSIX standard into multiple documents came as the draft standard was being balloted, with the result that its number has sprouted a -1.) ISO may decide to reprint the existing document, adding cover material to say that it is a standard. Alternatively, the standard may be published as a reference document: a few pages which tell the reader to go and look at a particular ANSI standard. (There is a precedent for this: the International Standards for COBOL and PL/1 simply point to ANSI documents.)

If ISO accepts the recommendation, POSIX should become an International Standard within the next six months.

ISO may turn down the request if it judges that the working group's plans to resolve outstanding issues are inadequate. Hopefully, this will not happen, because:

2. The working group has undertaken to produce and ballot an amendment to the standard by 1st June, 1990. The amendment – actually 1003.1a produced by the IEEE – will fix all issues raised during the balloting of DIS 9945. What is more, the working group – or rather, the hard-pressed editor for the IEEE's POSIX project – will merge the addendum with the existing standard, producing a single document in a format acceptable to ISO. This, it is hoped, will be published as a revised standard late next year.

The Future of International Standards for POSIX

In my last report, I noted that the working group had requested that its project be split into several parts, resulting in several standards, numbered 9945-1, 9945-2 and so on, rather than a single standard 9945. This has happened, with the result that the operating system interface will be covered by 9945-1; shell and utilities by 9945-2, and system administration by 9945-3. No other numbers have yet been allocated. It is important to note that the apparent one-for-one correspondence between 1003.1 and 9945-1 will grow more tenuous as time goes on: facilities for real-time processing (1003.4), security control

(1003.6) and transparent file access (1003.8) will be added to fiıture versions of 9945-1. While 9945-2 corresponds to 1003.2, there is no connection between 1003.3 (Test Methods) and 9945-3. Instead, 9945-3 – when it gets off the ground – will be based on the IEEE's 1003.7 work.

I also mentioned last time that ISO standards are supposed to be independent of any particular computer language. 9945-1 will probably lose its ties to C with its second amendment (that is, the amendment after the one described in the previous section). This will introduce a need for a new standard to describe its C bindings, and further standards to describe bindings for Ada, FORTRAN, and so on. While the IEEE language bindings are part of the 1003 project (1003.5 for Ada, and 1003.9 for FORTRAN), ISO practice is to allocate a completely new standard number for bindings work. Consequently, a request for a new number, with three designated parts, has been made. We will not know this number until next June.

Table 1 summarizes correspondence between ISO and IEEE standards.

A word about windowing is in order. Work in a number of JCT1 SCs nibbles at the edges of the issue:

SC2 (Code sets):
Encoding of pictures. There is no connection between this work and X's bitmap distribution format.

SC18 (Office systems):
Office system user interface; Font and character information interchange (lots of this); page layout and document structure (even more of this).

SC22 (Languages):
Form interface management system – a new project involving interactive screen forms and such.

SC24 (Graphics):
No work – even though SC24 looks like the obvious place to put windowing standardization.

It is an article of faith that no international standard may encroach on another's territory, and that the terms of reference of each SC do not overlap. This presents difficulties in dealing with new (well, new in ISO terms) and widely-applicable technologies such as windowing. Perhaps it may be possible to hand the issue to SC24 without upsetting other SCs. Alternatively, it may be necessary for JCT1 to set up a whole new SC to run with it, and bring the currently fragmented work together. (This recently happened on security issues – see below.) Again, watch this space for more news.

9945-2 Shell and Tools Standard

The majestic machinery of JTC1/SC22 has sanctioned the use of draft 9 of IEEE 1003.2 as a draft proposal (DP), which embarks forthwith on a six-month balloting period. This period is to be synchronized with the IEEE's ballot, with the result that 1003.2 and 9945-2 move forward in lock-step, and should hit the streets simultaneously as identical American and international standards.

Document Format

In order to avoid future wrangles over document format with ISO's Central Secretariat, and to avoid time wasted in recasting IEEE standards into ISO's mold, *all* 1003 standards are to be created and balloted in a format acceptable to ISO. (And to the IEEE. And to the POSIX working groups. But mostly to ISO.)

WG15 is concerned that ISO's standards for standards were drawn up with relatively short documents in mind. For example, ISO's Central Secretariat objects to the line numbers which appear in draft 13 of 1003.1 – even though it used the line numbers in referencing other changes that it wanted! Hopefully, an acceptable compromise will be reached. Working group chairs and editors will be told what the changes mean to them just as soon as a decision is reached.

Rapporteur Groups

The concept of rapporteur groups is an ISO invention. It refers to a group of "technical experts" (another ISO term) from a number of related standards efforts, or concerned with

ISO	IEEE	Topic	Notes
9945-1	1003.1	OS interface	Now
	1003.1a	Clean-up	1990
	1003.1b	Extensions, language independence etc.	Future
	1003.4	Real-time	Future
	1003.6	Security	Future
	1003.8	Transparent file access	Future
9945-2	1003.2	Shell & tools	First release
	1003.2a	User Portability Extension	Future
9945-3	1003.7	System administration	First release
1xxxx-1	–	C bindings	Future (probably to be done by new 1003 working group)
1xxxx-2	1003.5	Ada bindings	Future
1xxxx-3	1003.9	FORTRAN bindings	Future
–	1003.0	POSIX environment	Some overlap with ISO DP 10000, *International Standardized Profiles*
–	1003.3	Test methods	Under consideration by rapporteur group
–	1003.8	(Aspects besides T.F.A.)	Work elsewhere in ISO on RPC
–	1003.10	Supercomputing	Profile: relevant to DP 10000
–	1003.11	Transaction Processing	Profile; also relevant to SC21/WG3 database work
–	1201.x	X Window	See below
–	1224.x	Interfaces to OSI services	Not clear where these fit in ISO work: SC21 (OSI) seems to be against working on bindings

Table 1: Correspondence between ISO and IEEE Activities

a specialized topic within a single standards effort, which meets to discuss its area of interest. Members of the group then report back to their own groups, in order to integrate the work of the rapporteur group and the standards efforts that it links.

WG15 has three rapporteur groups: Conformance, Internationalization, and Security. Each addresses areas known to have applicability in fields broader than POSIX itself. For example, JCT1 has just created a whole new subgroup (SC27) to handle security, bringing together separate developments in SC18 (Office systems), SC20 (Data encryption), SC21 (Open Systems Interconnection), SC22 (Languages)* – and anything else which turns out to have security implications. (I mentioned this development in my last report, but managed to garble some of the references. Sorry about that...) Similarly, there is work on conformance testing and internationalization both inside and outside ISO.

In Brussels, the rapporteur groups all held informal meetings separate from the main business of WG15. Since all three have only just gotten off the ground, there is little to report as yet, but watch this space!

X/Open Portability Guide as a European Standard?

At the May meeting of WG15, our minds were much exercised by a proposal from CEN (Comité Européen pour la Normalization –

* Why is the POSIX project a subdivision of the languages subgroup? Because it was the least unsuitable place in the ISO structure to put it at the time...

The European Committee for Standardization) that the whole of the third edition of the X/Open Portability Guide (XPG3) should become a draft European prestandard. The arguments against doing this center on the fact that the XPG is not a formal standard reached (slowly) through consensus, but an informal document which references formal standards where it can, but which then goes on to fill the gaps with *de facto* and suggested standards material. Increasingly, the European countries which form CEN's membership have come to realize that a document of this type, while useful in its own right (arguably more useful than existing formal standards, in fact), cannot be adopted as a European standard for both legal and practical reasons.

XPG3 has, however, helped to focus European minds on areas where formal standards are lacking. At the moment it looks as though the CEN project charged with producing a POSIX standard will build on the output of WG15. In addition to this, Germany is in favor of adopting as prestandards those parts of XPG3 which do *not* correspond to existing or emerging international standards – for example, ISAM, curses and X Window. The argument for this is that some kind of standard is urgently needed in these areas. The argument against, coming from Britain, Denmark, the Netherlands and others, is that CEN can only adopt standards which are public – *de facto* just isn't good enough, and besides, such things are outside the scope of the original work order for a POSIX standard. At the moment, it looks as if this point of view will prevail.

As a sidelight to this issue, it seems that ISAM will eventually make it into the POSIX standard, as X/Open has expressed a desire to submit a base document to the 1003.1 working group.

Harmonization and Synchronization

The three previous headings – 9945-2, rapporteur groups, and the CEN standard for POSIX – highlight a couple of important issues identified by JCT1:

Harmonization:
Standards covering identical or related topics should be in agreement; and

Synchronization:
Development work on standards covering identical or related topics should be developed in step with one another, both so that there is no unnecessary delay between the appearance of one standard and the appearance of another, and to avoid duplication of work – for example, the same ballot objection being made to and fielded by two separate groups.

WG15 has taken steps to synchronize its activities with those of the IEEE 1003 working groups, its main feeder. In some cases this means that WG15 will set IEEE timetables – almost a case of the tail wagging the dog, but necessary in order to arrive at international standards as quickly as possible.

To address the issue of harmonization, WG15 discussed a new category of liaison to JCT1. Liaison is a mechanism which allows transnational and international setters and users of standards to monitor or to contribute to the work of ISO. Participation is otherwise the province of national standards bodies such as ANSI, JISC and DIN – ISO is currently bad at dealing with regional standards bodies such as CEN. The proposal embodied a combination of sticks and carrots which would allow other types of standards bodies to participate on the condition that they undertook to align with relevant international standards within some reasonable time after publication. The working group reached no conclusion on this radical idea, and will discuss it again at its next meeting.

It will be a while before JCT1 gets around to considering any proposal of this nature. In the meantime, WG15 will continue to invite observers such as myself to its meetings.

Language Independence

As at the previous meeting, this topic was discussed at some length. The policy of JCT1 is that, ultimately, in the interests of precision and verifiability, all base standards should be written in some formal language which is itself the subject of an ISO standard. There is a small problem here: no formal language suitable for use in the POSIX project is yet the subject of a standardization effort (although IEEE P1003.7, System Administration, is making use of ASN-1, a standardized formal language developed for use in describing communications systems). If POSIX were to wait for a formal language to be standardized before breaking the current links between POSIX and C, nothing could be done for a couple of years. However, it is necessary to break the links with C as soon as possible, in order that additional bindings for Ada and FORTRAN can be defined. The break will be made informally, by using English along with language-independent data types, and so on.

In parallel with this development by WG15, a research project funded by the European Community (EC) looks like it will be funding the development of a description of the POSIX operating system interface in VDM-SL (Vienna Definition Method Specification Language). SC22 is actually thinking about standardizing this formal language, which is already being used in the production of an ISO standard for Modula 2 by SC22/WG13. Welcoming what is, in effect, an offer to discover the problems involved in defining POSIX using a formal language, WG15 has sent a message of encouragement to the EC, while emphasizing to SC22 that, as far as POSIX is concerned, the coming language-independent description is a necessary step on the path towards a formal definition.

The Portable Common Tools Environment

Another research project supported by the EC concerns the Portable Common Tools Environment, PCTE. Essentially a very sophisticated and all-encompassing object-based workbench for the support of Computer-Assisted Software Engineering (CASE), PCTE is the result of six years' work, and the investment of several million European Currency Units (ECUs) by government and industry – with more years and mega-ECUs to come. Among other organizations, NATO is a strong champion of the technology. The European Confederation of Computer Manufacturers (ECMA) has, over the last couple of years, been working on a PCTE standard which may (just) be ready in 1991, and which may then be offered to ISO.

What has this to do with POSIX? Well, PCTE was originally aggressively host-independent – independent, that is, both of hardware and, on systems where it was not to run native, of operating system. This made excellent sense six years ago when development started – using UNIX as a development host. Versions are currently available for several UNIX hosts, with VMS and IBM mainframe versions on the way. Times move on, however, and there is now (ISO Central Secretariat permitting) an international standard hardware-independent operating system which looks like it will become the predominant host for PCTE. It makes sense, therefore, for PCTE to align itself closely with POSIX, so avoiding unnecessary duplication or conflict of functionality. Following a morning of presentations by PCTE experts, WG15 agreed to keep members of the ECMA PCTE working group informed of its activities.

Next Meeting

The next meeting of WG15 is to be held in Paris, France from June 11-15, 1990, and is to be hosted by AFNOR, the French national standards body.

An Update on UNIX and C Standards Activity

Jeffrey S. Haemer
Report Editor, USENIX Standards Watchdog Committee

IEEE 1003.0: POSIX Guide Update

An anonymous correspondent reports of the April, 1989 meeting:

The April session of 1003.0 was fruitful. The most significant accomplishment was the proposal and development of definitions the committee feels it needs to describe an open systems environment properly and adequately. Five definitions were developed:

- open system environment
- application environment
- application environment description
- application environment profile
- POSIX open system environment

Group consensus was that the first four would be submitted to the JTC1 Application Portability Study Group as a draft proposal for its work. The committee added the caveat that these were draft definitions, subject to change. A key clarification by these definitions was the distinction between an application profile and an open system environment: a profile is a subset of the environment.

The guide document, being developed by 1003.0, is nearly mature. Significant strides were made in the architecture section, which focuses on the operating system interface, languages, and network services. In the following months, 1003.0 will turn its attention to database management, data interchange, and graphics. The user interface section will be closely coupled to the work of the newly formed, IEEE 1201.1 (Xwindows) working group. Similarly, the the transaction processing section will track the on-line transaction processing (OLTP) group (1003.11).

There is some worry about the length of the guide – currently 135 pages and growing. If the document becomes unwieldy, some attention will be turned to scaling it down.

The committee also created an Internationalization study group, to cut across groups and help increase inter-group coordination in this area. The study group intends to become a full working group in Brussels, this October.

IEEE 1003.0: POSIX Guide Update

Kevin Lewis <klewis@gucci.enet.dec.com> reports on the July 10-14, 1989 meeting in San Jose, California:

As 1003.0 passes the mid-point of calendar year 1989, progress can be earmarked by the arrival of line numbers to the guide document. I remember the first time I saw line numbers on a document within the IEEE 1003 arena. My first thought was "this committee is really doing precise, exacting work." Thus was my reaction again when I saw line numbers on our document. My balloon was burst, when one of our active members – and by "active member" I mean someone who commits contributions in writing, not just someone who comes to voice an opinion in a talk-show-like atmosphere – pointed to our ISSUE LOG, which states that the committee needs to do more work. (There's that word again.) Alas, I came back down to earth. I have "miles to go before I sleep."

Dot Zero continues to converge. Our document is finally beginning to tie together the standards and elements that comprise a POSIX open system. Key events continue to be the definition of terms that will eventually make it to the IEEE Glossary and the identification of areas where terms still need definition.

The group is still generating discussion/debate/argument/food-fights over behemoth macro-questions such as, "What is the role of the guide?" and, "What is the PROPER audience?" In addition, the group has made valiant attempts at addressing specific areas such as graphics and data interchange without the benefit of focused expertise. We now agree on our ignorance in these areas, and will seek help and/or point to other committees that, we believe, can come up with the answers.

Overall, we must meet our objective of going to ballot in October 1990, because that is what I told my wife, who is still trying to figure out what in the world a "dot zero" might be.

IEEE 1003.1: System Services Interface Update

Shane McCarron <ahby@bungia.mn.org> reports of the April, 1989 meeting:

"After thinking about it, I realized that 1003.1 did actually do some stuff this quarter." [April -ed]

1003.1 is preparing two supplements, A and B, to 1003.1-88.

At the 1003.1 meeting in Minneapolis, the group reviewed draft 0.1 of 1003.1, supplement A. This supplement contains only clarifications and editorial comments, and will be balloted in the Summer. It will be provided to the ISO as the United States' comments on the International Standard IS 9945, which is the same as 1003.1-1988. Its goal is to ensure that the ISO standard and the IEEE standard (with supplement) are functionally identical.

Supplement B, to be balloted later, contains substantive changes: new facilities absent in IEEE Std 1003.1-1988. Some were missing from 1003.1-88 because they weren't completely specified in time to be included in the first release of the standard. Others are being introduced due to requests from other standards committees or the user community. For example, the ISO working group responsible for POSIX has requested a new archive format. It argues both that the archive formats in the first standard are insufficient for the future needs of POSIX systems and that a dual solution is unacceptable. The new format uses ANSI standard labeling, but extends it to permit POSIX filenames, security information, etc.... Supplement B also includes symbolic links, *truncate*(), *ftruncate*(), *putenv*(), *clearenv*(), *getpass*(), *seekdir*(), *telldir*(), *chroot*(), *fchmod*(), *fchown*(), and *fsync*().

Supplement B will also contain additional clarifications and edits to the base standard. The ISO will probably designate this supplement an addendum to IS 9945. All this maneuvering ensures that the different standards stay in sync, and prevents large delays in getting the ISO standard approved.

Although 1003.1-88 is now official, the 1003.1 committee's work will continue for some time yet. As other POSIX standards gel, their committees uncover requirements for additional functionality or semantics in the base standard, to support their work. As these committees point out such cavities in the standard, P1003.1 works to fill them. Everyone's hope is that no root canals will be necessary.

IEEE 1003.3: Test Methods Update

Doris Lebovits <lebovits@attunix.att.com> reports on the July 10-14, 1989 meeting in San Jose, California:

Overview

This was the thirteenth meeting of P1003.3. Monday through Wednesday, the group began work on a verification standard corresponding to 1003.2 (Shell and Tools). Following the close of the formal meeting, the technical reviewers of the draft 10 ballot met for the remainder of the week.

Meeting Summary

This was the first meeting to develop the verification standard for P1003.2, which will contain test methods and test assertions for measuring 1003.2 conformance. This standard will ultimately form part III of P1003.3. (Part I contains definitions, generic test methods, and so forth; part II is test methods for measuring P1003.1 conformance, including test assertions. As other P1003 standards reach maturity, their verification will, in turn, be covered in new parts of the P1003.3 standard.)

The chair's aggressive goal is to be ready to bring part III to ballot after four quarterly meetings. A detailed schedule and milestones will be developed at the next meeting.

Attendees included representatives of AT&T, NIST, OSF, Mindcraft, IBM, DEC, HP, Data General, Cray Research, Unisys, Perennial, and Unisoft Ltd. The meeting agenda included:

- the confirmation of new officers for the the part III work
  Chair: Roger Martin
  Vice-chair: Ray Wilkes
  Technical Editor: Andrew Twigger
  Secretary: Lowell Johnson

- the rough scheduling and setting of general milestones for part III

- a meeting with the P1003.2 working group to discuss testing issues

- action item assignments

- identification of items for the next meeting

In addition, small groups formed to discuss and resolve three specific issues. One group investigated the difficulty of thorough testing of the more complex utilities: *awk*, *bc*, *ed*, *lex*, *make*, *pax*, *sh*, and *yacc*. (The resulting action item was to produce a prototype set of assertions.) A second group scoped the writing of assertions for BNF type structures: [] expressions, regular expressions, and extended regular expressions. The third reviewed "Verification of Commands Interface," a paper by Andrew Twigger of Unisoft Ltd. The paper covers:

- character set and locale
- internationalized utilities
- underlying OS primitives
- regular expression testing
- pattern matching notation
- utility syntax rules
- errors from P1003.1 associated functions
- environment variables
- standard output format
- standard error format
- environmental changes
- symbolic limits
- obsolescent features
- job control
- read-only variables
- signal numbers

NIST has contributed its current 1003.2 test assertions to provide a basis for the 1003.2 verification work. Sheila Frankel of NIST gave a short presentation on the current state of these assertions, which include approximately 900 Mindcraft test assertions, plus 2600 newly-created assertions, all based on P1003.2 Draft 8.

Technical Reviewer's Meeting

In parallel to the verification work for P1003.2, balloting and revision is taking place on draft 10 of parts I and II.

As of July 6, 1989, 77 responses had been received from the 125 members in the balloting group. Eighteen additional responses will bring this to the 75% response needed to officially close the ballot.

The tally of the 77 responses:

28 positive	(36%)
31 negative	(40%)
18 abstain	(24%)

The technical reviewers held a plenary session to evaluate and respond to the comments and objections to this draft. Group consensus decided each issue and each decision was final. Part I was reviewed completely but only a few chapters of part II were reviewed. The remaining part II work was assigned to volunteers.

Draft 11 is planned for ballot recirculations in October, 1989, and an approved standard for parts I and II is anticipated by the first quarter of 1990.

IEEE 1003.4: Real-time Extensions Update

John Gertwagen <jag@laidbak> reports on the July 10-14, 1989 meeting in San Jose, California:

The P1003.4 meeting in San Jose was very busy. The meeting focused on resolving mock ballot objections and comments. Despite limited resources for documenting changes, a lot of work got done. Here's what stood out.

Shared memory

The preferred interface falls somewhere between shared-memory-only and a mapped-files interface, such as AIX's *mmap*(), which allows files to be treated like in-core arrays. Group direction was to reduce the functionality to support only shared memory, so long as the resulting interfaces could be implemented as a library over *mmap*().

Process memory locking

The various region locks were clarified and, thus, simplified; the old definitions were fuzzy and non-portable. For those who haven't seen it, there is actually a memory residency interface (i.e., fetch and store operations to meet some metric) instead of a locking interface. Most vendors will probably implement it as a lock, but some may want it to impose highest memory priority in the paging system.

Inter-process communication

Members questioned whether the interface definitions could really support a broader range of requirements; they're like no others in the world today. Having been designed to meet the real-time group's wish list, there are lots of bells and whistles – far more than in System V IPC – but it's not clear, for example, that they are network extensible. Discussions in these areas continue.

Events and semaphores

Members were concerned about possible overlap with other mechanisms, especially those being considered for threads. The question is basically, "Should there be separate functions for different flavors or a single function with multiple options?" General sentiment (including our snitch's) seems to be for multiple functions; however an implementation might choose to make them library interfaces to a common, more general system call. There is, however, a significant minority opinion the other way.

Scheduling

Many balloters found process lists and related semantics confusing. An attempt was made to re-cast the wording to be more strictly in terms of process behavior.

Timers

Inheritance was brought in line with existing (BSD) practice.

* * *

Outside of the mock ballot, there were two other major news items.

First, there is a movement afoot to make the .4 interfaces part of 1003.1. They would become additional chapters and might be voted separately or in logical groups. This would bring P1003 in line with the ISO model of a base standard plus application profiles. P1003.4 would become the real-time profile group. This is a non-trivial change.

Up to now, the criterion for .4 has been that of "minimum necessary for real-time," and has coincidentally been extended to support other requirements "where convenient." This is not a good starting point for a base interface. For example, *mmap*(), or something very much like it, is probably the right base for "shared storage objects," but real-time users want an interface for shared memory, not for mapped files. Our snitch worries that things might look a bit different had the group been working on a base standard from the beginning.

Second, the committee officially began work on a threads interface, forming a threads small group and creating a stub chapter in the .4 draft. A working proposal for the interface, representing the consensus direction of the working group, will be an appendix to the next draft.

A lot of work remains to be done before .4 can go to ballot and the current January '90 target may not be realistic. If the proposed reorganization occurs, a ballot before the summer of 1990 seems unlikely.

IEEE 1003.5: Ada-language Binding Update

Ted Baker <tbaker@ajpo.sei.cmu.edu> reports on the July 10-14, 1989 meeting in San Jose, California:

The Ada-language binding for 1003.1 is progressing steadily, though behind schedule. The group agreed to try to prepare a document for the October meeting in Brussels that is ready for mock ballot. Those at the meeting will decide whether the document has achieved this goal. If not, we will try again at the January meeting in New Orleans.

The slow progress is mainly due to the long time between meetings and the limited work force available to do the writing. The members, all volunteers, must steal time for POSIX from their "real" (i.e. paying) jobs. Attending quarterly meetings already puts most members near the limit of time they can spare.

Most significant technical issues seem to be resolved; the remaining controversies center on almost-religious issues, such as the exact grouping of interface declarations into Ada packages, naming, capitalization conventions, and where to strike the balance between providing full functionality and idiot-proofing the interface.

Each chapter has been assigned to a person for review and editing, based on decisions made at the San Jose meeting. Quite a lot of writing still needs to be done. Chapter 7 ("Device- and Class-Specific Functions" - i.e. terminal interfaces) is still empty, and some others are still mostly just Ada code, with no discussion. Most of the rationale remains to be written. Mitch Gart has agreed to coordinate this, including a chapter on "meta-issues" - design decisions affecting the entire interface. David Emery will combine the chapters to produce the next draft.

Interaction with 1003.4 (Real-Time Extensions) has heated up, with 1003.4's consideration of support for multi-threaded processes. Ada language implementations must support multiple tasks (i.e. threads) within a POSIX process, to comply with the Ada language standard. Neither the 1003.1 standard nor the 1003.4 draft that just completed mock balloting supports multithreaded processes, so the Ada implementor is currently forced to hack out some sort of internal concurrency scheme, with its own layer of dispatching, for each Ada process. This tends to run aground when one Ada task makes a blocking system call, since the whole process is forced to wait. Naturally, Ada implementors and users would be pleased if the POSIX interface provided for concurrency within a process.

The Ada group is very interested in the threads proposal, and most members would like to see some support for threads in the 1003.4 standard that goes to formal ballot. Some members are a little bit concerned that those working on the proposal may not understand Ada tasking well enough to ensure that the proposed threads will be adequate to implement Ada tasking semantics. This has been very frustrating for members of the Ada group, since the discussions of the threads proposal were all in parallel with meetings of 1003.5. The best the Ada group was able to do was to keep one observer present (on rotation) at the review of the threads proposal. It is not clear whether this was adequate.

[Editor's note: What's going on here, and in the second paragraph, is that some groups are much larger than others. 1003.5 is among the smallest. The 1003.4 session I saw had about forty overworked attendees. The 1003.5 sessions I saw had five to ten.

1003.5 could use a lot more participation from the Ada community. Unfortunately, this may be a case of "once burned, twice shy." For years, there's been a lot of talk about "Ada environments," all of which seem, from a UNIX perspective, like enormous, cumbersome projects that might actually come into widespread use in, if not our children's lifetimes, perhaps their children's.

Make no mistake about it: the Ada community is huge. And easy availability of machines with implemented, Ada-language bindings to POSIX-conformant operating systems would be immensely useful to that community. The ability to buy a box, off-the-shelf, with a portable environment for running Ada programs in the next couple of years, would make Ada programmers' lives immensely easier and even be a big aid in implementing the richer and more complex environments mentioned in the previous paragraph.

Still, you can guess what the average, UNIX-naive, Ada programmer must think: "Whoopie, another standard/environment. I'll have to take a look at it in a few years to see how it's coming along." If the IEEE could make some non-vanishing fraction of the Ada community understand that POSIX is on the verge of being here, now, dot 5 might get a lot more help.

This seems to us (that's the editorial "we," folks) like a quintessential marketing problem. If 1003.5 could enlist the help of 1003.0 in this matter, they might be able to make some real headway here.]

The 1003.5 group is also very interested in the progress of the language-independent versions of the POSIX standard. Much of the labor of the Ada binding group has been devoted to separating the essential semantics of the 1003.1 interface from the details of its expression in the C language (for example, *setjmp*(), *longjmp*(), and signal handlers). This labor may be of use to those working on the language-independent version of 1003.1, but the Ada group does wish that new standards, such as 1003.4, would start out with a language-independent document, rather than adding to the language-bias problem.

There was one change in the leadership of the 1003.5 working group. Stowe Boyd, of Meridian, had been vice-chair but is no longer able to spare time from his job to work on this project. Steve Deller, of Verdix, has agreed to replace him. This is a very important job, since the vice-chair of the 1003.5 group takes direct responsibility for setting the technical agenda and running meetings.

IEEE 1003.6: Security Extensions Update

Ana Mariá de Alvaré (anamaria@lll-lcc.llnl.gov) reports of the April, 1989 meeting:

P1003.6 covered these global issues at the five-day Minneapolis meeting:

1. Supplements to 1003.1 will address portability, data interchange format, and symbolic links. This means 1003.6 must also consider those areas.

2. 1003.6 would like to define a system variable that tells what security policies are allowed on the system, and a function that returns which security-related attributes (e.g., MAC, ACL) are currently in operation. Such changes would need to be made in collaboration with 1003.1.

3. Other pieces of 1003.1 and its supplements may conflict with security extensions. A short-term subgroup was proposed to review these documents and propose additions or changes. 1003.6 is looking for volunteers for this work.

[Ed. – If you have, or can imagine, the orange book and the ugly green book side-by-side on your bookshelf, now's the time you should work to ensure that only their colors clash. The chair of 1003.6 is Dennis Steinauer, who, we believe, would be happy to hear from you if you're willing to help (steinauer@ecf.ncsl.nist.gov).]

4. Two members of the networking group (1003.8) joined 1003.6 for half a day to list and explain their areas of concern: transparent file access, authentication at mount time, setuid programs, file format, local id contents, and who does the audit. These issues were scheduled to be revisited at the San Jose meeting in July in a joint meeting of the two groups.

5. Charlie Testa gave a status report on TRUSIX. The TRUSIX working group responded to Tom Parenty's paper, which summarized the TRUSIX efforts. The members felt compelled to clarify certain sections that they believed misconstrued their real objective: the creation of a trusted UNIX operating system. This response is also published in the December, 1988, Data Security Letter Journal.

There are serious conflicts and points of contention between POSIX and TRUSIX. POSIX is worried that systems conforming to TRUSIX recommendations will get preferential treatment during product evaluation, that vendors who currently plan only Class B2 systems or below are excluded from TRUSIX, and that participants in TRUSIX share proprietary information. TRUSIX takes the position that the marketplace should be the final judge. TRUSIX will be POSIX compliant, and will make no attempt to force vendors to be both POSIX and TRUSIX compliant. If customers force a *de facto* standard of dual compliance for even non-DOD applications, so be it.

TRUSIX's ACL proposal will be delivered to the IEEE at the July meeting. The proposal is only a guide, and it will not be written in a formal specification language as a favor to the reader.

TRUSIX's audit subgroup is trying to follow both POSIX and X/Open efforts in this area. Their subgroup is focusing on pre-selection, in contrast to the 1003.6 focus on post-selection, and they will review a token-based scheme at their next meeting.

6. At the previous meeting, a common descriptive top-level specification language (DTLS) was proposed. For the moment, this language will form an appendix to the draft, and will be used as an internal tool to let the

group define unambiguous security interfaces. Every subgroup of 1003.6 will provide descriptions of interfaces in both English and DTLS. Steve Sutton will be the chair of the DTLS team, and will work in conjunction with the technical editor of the draft.

The Security Working group is split into separate groups for audit, discretionary access control (DAC), mandatory access control (MAC), and privileges. Each subgroup gave a summary report at the end of the week and some were able to give a first-cut delivery schedule. The following is a short summary of each group's efforts.

Audit

The scope of the audit group encompasses audit definition, auditable events, audit trail contents, and audit trail access and control. The group will also define a portable audit trail data representation and focus on post-processing event classes.

Audit records will include process identification, audit id, effective user id, effective group id, media addresses, MAC labels, and privilege information. In San Jose, the audit group will try to identify all token types, define the audit id, propose some changes to the "seek" function, pursue event classes, and review and merge the DTLS interface descriptions with the English sections.

DAC

The DAC group is almost done with its rationale section. One question this time around was how to pass access mechanisms based on DAC across the network. Currently, file ownership is the first access check; on networked systems, this can lead to spoofing, particularly when root tries to access files on other systems.

Another hot issue was access functions. The consensus is that an access function to an opaque DAC (i.e., one that prevents knowledge of the structure) should replace the use of *stat*(), *chmod*(), *stat*() or locking mechanisms for controlled file access. The function will not replace *chmod*(), *stat*() or permission bits; however it will define operations that will allow applications to continue to work correctly in the face of ACLs.

MAC

Issues addressed here come from the MAC requirement that all system objects be labeled with security levels (e.g., CONFIDENTIAL, SECRET, TOP-SECRET). Two proposals were on the table – one from Addamax, the other from Olin Sibert – but no strong consensus was reached. Miscellaneous comments on the issues discussed:

1. Downgrading (of security levels)
   - How should it be done?
   - Must the old label dominate the new?
   - Does downgrading need to be strictly controlled?
   - What about upgrading?

2. Directory labels.
mkdir should be allowed to label directories on creation, to permit portable, level-hierarchy-dependent applications.

3. File locking.
The standard should address locks and may consider them as objects.

4. "Write-up" appends.
Writing to a file at a level above you is known as "write-up." Processes can write to files that they can't read. At first blush, this seems analogous to standard UNIX, which allows files with permissions `--w--w--w-`. What MAC adds is the prohibition that the process even know if the write succeeds. Because appending to such a file provides no way to assure that the write succeeded, the question of whether to allow such write-ups was raised and discussed.

5. Change of file level with open file connections.
UNIX does not expect open connections to break. (An exception is */dev/tty** on 4.3 BSD, which can be checked for open connection breaks.) Since */dev/tty** are special files and 1003.1 doesn't address special files it was argued that 1003.6 need not either, but this issue will be discussed further in San Jose.

6. Open tranquillity.
The tranquillity property states that a resource should not be in active use during changes to its attributes. (See also issue #5 above.) It was stressed that POSIX should be defining states and mechanisms that are as safe as

possible, obvious to implement, deterministic, and clear. Only privileged processes should be able to change the MAC label of a file object.

7. Replication or Recalculation?
Replication means copying current properties across from one label to another. Recalculation means re-evaluating the situation, then assigning properties or attributes needed for each file to work as labeled. The consensus was that recalculation is needed in the standard, but there was no consensus on how either recalculation or replication should occur.

8. Multilevel directories
A "multilevel directory" is a directory with files at different levels (e.g., both TOP SECRET and CONFIDENTIAL). Should a multilevel directory feature be available for general use? Should it be part of the standards? If so, operations on multilevel directories would be restricted and functions to be able to create, check for existence, and query for directory name would be required. These directories would inherit their DAC from their parent.

The directory that stores files the user can see at the current time, as determined by the label at request time, is the "access hidden directory." An open question is whether access to such a directory should be controlled by process privilege or the pathname syntax.

9. Text Format
Two proposals were put forward on text format, but only one was discussed because of time constraints. Despite this, the group resolved that naming should be site-specific, but names should be unique and order-independent. Furthermore, a label should be interpretable and unique. One major problem was that the characters suggested for hidden directories were outside the constrained character set provided by 1003.1 – `[a-z][A-Z][0-9]` and a very limited set of punctuation characters.

10. System High/Low?
This government concept is used a lot in discussions of secure systems. It was put on the agenda for the July, San Jose meeting.

11. Other Issues
Should the standard assure a non-decreasing directory hierarchy? In other words, should subdirectories always have at least as high a level as the parent? Should the standard define level ranges such as system high? Should the standard define a process clearance range? (Clearance only defines how to specify an error return that the system is allowed to give.)

Privileges

The group reviewed interface functions defined at the previous meeting, and agreed on all of them except *exec*(), which poses unresolved problems about inheritance of privileges. The group expects to finish this in July.

Some of the functions defined so far are:

is_effective(p)
make_effective(p)
make_ineffective(p)
is_inheritable(p)
make_inheritable(p)
make_not_inheritable(p)
is_permitted(p)
relinquish(p)
make_effective_if_inherited(p)
make_all_ineffective(p)

all related to querying the process privilege state.

Old goals were revised and new goals added, including: support for old binaries, support for new binaries implementing true least privileges, acquisition of effective privilege following *exec*(), prevention of some programs from inheriting privileges, and unsetting of privileges on exit from signal handlers.

Other issues included:

1. Privilege inheritance
When is it needed?

2. Forbidden privilege
Should a flag be available to forbid a process to gain a privilege?

3. Privilege System Variable
Should the standard define a system variable to set privileges at installation time?

IEEE 1003.6: Security Extensions Update

Ana Mariá de Alvaré <anamaria@lll-lcc.llnl.gov> reports on the July 10-14, 1989 meeting, in San Jose, California:

P1003.6 (security) is split into four main groups: privileges, mandatory access control (MAC), audit, and discretionary access control (DAC). In addition, there is a definitions group, whose charter is to define terms and to ensure that definitions used by 1003.6 do not clash with definitions in other 1003 groups.

Definitions

The definitions group reviewed all definitions new to draft two. The majority were from the audit group and were approved. Amusingly, the lone exception was the definition of "audit," which included an interpretation of an audit record; the definition group considered this to be outside the audit group's goals.

The group also chose a global naming convention, *PREFIX*_FUNCTIONNAME, where *PREFIX* represents the security section/topic. Current prefixes are "priv_," "mac_," "aud_," and "acl_" (DAC). The same prefix rule extends to data structures (e.g. "priv_t").

MAC

Several issues were resolved.

- A "write up" standard will be neither restricted nor guaranteed.

- The "upgrade directories" function was dropped, since a "write up" without a read does not guarantee success.

- Change file label/level and change process label operations will be accepted for privileged processes.

- The MAC_PRESENT variable will be added to the sysconf, to indicate that a MAC mechanism is installed in the system. MAC_CONTROLLED and MAC_ALWAYS were also proposed. MAC_CONTROLLED would return the value of a file controlled by a MAC mechanism, and MAC_ALWAYS would indicate that all objects on the system contain associated MAC information.

- A set of six privileges were defined:

  P_upgrade
  P_covertchannel
  P_MAC_READ
  P_MAC_WRITE
  P_LABEL_OBJ
  P_LABEL_SUBJ

The last two might be folded under READ/WRITE privileges, however these two are the most sensitive of all.

The next meeting will see discussions of Sun's multiple-level directories, the recalculation function, and information labels. The group will also review the .6 draft, the MAC common description language interface, and 1003.1/.1a.

Privileges

The privilege group has defined interfaces for file privileges. For example, *priv_fstate_t*() will return whether privilege for the file is required, allowed, or forbidden. A process's privilege can be permitted, effective, or inheritable.

Also, there is now a list of needed privileges, including PRIV_SETUID and PRIV_SETGID (set the uid and gid of a file or process), PRIV_FOWNER (change the owner uid of a file), PRIV_ADMIN (do administrative operations like unlinking a file), PRIV_RESOURCE (set the sticky bit or be able to use memory), and DAC_READ/WRITE (override access search or read and access write).

The process-privilege interface is still an open issue, and will be discussed in October. These three suggestions are on the table:

1. A function pair. *priv_set_priv(id, attr, value)* and *valuet priv_get_priv(id, attr)*. (Something of type "valuet" can take on the values "required," "allowed," or "forbidden.")

2. An interface to set or unset multiple privileges at a time.

3. A requirement that the operating system recalculate privileges for each process every time that process manipulates an object.

Next meeting, the privilege group will focus on developing functional interface descriptions in both English and in Common Descriptive Language (CDL).

DAC

The DAC group decided to describe interfaces using a procedural interface. They defined the minimum set of functions required for access control lists (ACLs) - open, close, write, sort, create_entry, get_entry, dup_entry, delete_entry, set_key, get_key, and add/delete permission - and the minimum set of commands - *getacl* and *setacl*. They also defined the needed privileges and passed their list to the privilege group. The October meeting will focus on polishing the current draft and addressing default ACL interfaces.

Audit

The group discussed portability, especially data portability. Should only privileged processes write to audit logs? (The consensus is, "Yes.") And how much should the record format be standardized?

The October meeting will see a draft review, plus discussions on event identification, classes, style and data representation, and token grammar.

New Group: Network/System Administration

Because interconnectivity is at the heart of many security and administration issues, "interconnectivity" between P1003.6, P1003.7 (system administration), and P1003.8 (networking) had to improve. A joint evening meeting of the three groups set this in motion, and five members of 1003.6 have signed up to review drafts from the other two groups. They intend to begin working on this area formally in October.

IEEE 1003.7: System Administration Update

Steven J. McDowall <sjm@mca.mn.org> reports on the July 10-14, 1989 meeting, in San Jose, California:

War and Remembrance - How I survived a POSIX Meeting

Listen closely to this tale of wonder and bewilderment and hope that you shall never have to face such horrors as I. Yes, I was there when, in a flurry of activity, the 1003.7 committee elected Steven Carter to the chair. To show he was a good choice, Carter immediately sat on the chair to which he'd been elected. This was swiftly followed by the election of Vice-chairs Martin Kirk and Dave Hinnant (though I shall speculate not on what vices they may have perpetrated on those chairs); Mark Colburn, Secretary (owing to a proven ability to take dictation lying on a pool-side sun bed); and their honors Bob Bauman and Shoshana O'Brien, Technical Editors.

You may sense that I feel few exciting things happened in San Jose. Correct. I wish this group would get into some real fights, like other groups. Interoperability may prove our only hope. Still, progress is progress, however uncontentious. Here's what else seemed to me to be important.

1. Language Independence

The group voted, nearly unanimously, that the country of Language should be independent. We were uncertain about where, precisely, it might be, but tentatively put it near Borneo.

We chose to use ASN.1 ("Abstract Syntax Notation - 1") as our internal notation for data structures. The group also appointed me representative to the 1003.1 language-bindings group to watch what those pursuers of knowledge are doing in this area.

2. Interoperability

X/Open continues to push this into the foreground. Luckily for us, they also continue to help us understand what it entails. Group consensus holds that interoperability is within the purview of 1003.7. What we're still uncertain of is how far down we should standardize; only through the application layer? down to the packet layer?

For example, a standard application-layer protocol ensuring interoperability might require that certain Application Program Interface (API) calls be available, with given arguments and results, but say nothing about how those calls are made. In contrast, a transport-level protocol might require that the information be fed into the API will be in a pseudo-ASN.1 format to help in non-homogeneous networks. A still lower level protocol might detail the exact packet structure, including ASN.1 format for the object data, to prevent foreign machines in a non-homogeneous network from throwing out otherwise unrecognizable packets.

Most committee members have strong, idiosyncratic ideas about this subject and the issue is certain to resurface in Brussels. We need input on this from the community at large. Where do YOU think a standards organization like the IEEE should draw the line in ensuring interoperability?

[Editor's note – This is not a rhetorical question. Things you do in the future may be affected by decisions P1003.7 makes in this arena. If you have an opinion on this subject, speak up.]

As an aside, the current X/Open representative, Jim Oldroyd of the Instruction Set, Ltd., who has really helped the group a great deal in this area, may not attend the next 1003.7 meeting. We think this would be a real loss, and hope that X/Open and his employer find a way to arrange for him to go.

3. Misc.

Some progress was made in doing the ASN.1 syntax for a few of the basic objects the committee decided on for phase I of the standard. Everyone is discovering that defining such objects (File Systems, Devices, Spools, etc.) in a non-ambiguous way using a meta-language like ASN.1 might not be as easy as we first thought. Live and learn, eh?

IEEE 1003.8: Networking (IPC) Update

Steve Head <smh@hpda.hp.com> reports on the July 10-14, 1989 meeting, in San Jose, California:

Overview

P1003.8 is the IEEE POSIX committee working on network standard interface definitions for POSIX. The committee is divided into several subcommittees, including transparent file access, remote procedure call, network IPC, and MAP. This report summarizes recent activity in the network IPC subcommittee, which is currently working on two potential interfaces: a "detailed" network interface (DNI) and a "simple" network interface (SNI). DNI is roughly (though not exclusively) at the transport level. SNI is intended to be somewhat simpler to use than DNI, but at roughly the same level.

At this meeting, a draft of DNI was begun, which included a scope, a chapter-by-chapter outline of the document specifying functionality included in each chapter, and the beginning of a rationale, which discusses goals. For SNI, goals, objects, and functionality were discussed, but without a full resolution.

Also, a schedule was adopted which forecasts the activities of the committee towards mock ballot and full ballot of DNI and SNI through January 1993.

Several joint meetings with P1003.6 (security) and P1003.4 (real time) were held on the subjects of network security and real-time IPC.

Plans were made to make P1003.8 a steering committee and to elevate each P1003.8 subcommittee (including P1003.8/2) to full POSIX committee level.

At this meeting, the main topics of discussion were:

DNI draft

A draft of DNI was begun. The draft now includes a scope, plus skeleton chapters on connection setup and tear down (including naming), data transfer, async event management, option management, POSIX 1003.1 extensions, OSI transport protocol family options, and Internet protocol family options. Appendices include related standards, a rationale, and comparisons with X/Open's XTI and BSD's sockets. Each chapter is currently language-independent, specifying functionality only, not C routines.

So far, DNI is a functional superset of XTI and sockets, although this has not been formally adopted as an explicit goal by the group.

SNI goals, objects, and functionality

The group discussed SNI goals, objects, and functionality. Some progress was made. SNI's proponents now envision it as being capable of complex operations, such as async events. Users will be able to intermix SNI and DNI routine calls as needed.

SNI may adopt some of the characteristics of UNIX standard I/O, specially tailored for networking, but the exact relationship to the UNIX standard I/O package has not yet been addressed.

Schedule

A tentative schedule was adopted for DNI and SNI.

Summer 1989 meeting
SNI goals/functionality; SNI/DNI outline
Fall 1989 meeting
SNI/DNI connection setup/teardown
Winter 1990 meeting
SNI/DNI data transfer
Spring 1990 meeting
SNI/DNI event management
Summer 1990 meeting
SNI/DNI POSIX 1003.1 extensions
Fall 1990 meeting
SNI/DNI protocol-independent options
Winter 1991 meeting
SNI/DNI miscellaneous functionality DNI protocol-dependent (ISO, ARPA, etc.) options
Spring 1991 meeting
SNI/DNI definitions
Summer 1991 meeting
SNI/DNI review drafts
Fall 1991 meeting
SNI/DNI approve drafts for mock ballot
Oct. 1991
SNI/DNI mock ballot
Winter 1992 meeting
SNI/DNI resolve mock ballot objections
Spring 1992 meeting
SNI/DNI review drafts
Summer 1992 meeting
SNI/DNI approve drafts for full use ballot
Aug. 1992
SNI/DNI full use ballot
Fall 1992 meeting
SNI/DNI resolve full ballot objections
Winter 1993 meeting
SNI/DNI resolve full ballot objections
Feb. 1993
SNI/DNI submit approved drafts to IEEE stds. board
Spring 1993
data representation network interface goals ...

Security

We held two joint meetings with the POSIX security committee (P1003.6).

P1003.6 more or less views its role as describing necessary high-level security features and requirements, and would like to leave the job of filling in specific interfaces to P1003.8. This is agreeable to P1003.8, but both groups need to work to ensure that this division of labor leaves no holes.

Paul Melmon, of Hewlett-Packard, also made a presentation on Internet protocol address family security. The presentation covered a special interest topic, B1 security for TCP-IP networks. For this level of security, security labels are usually automatically inserted into the IP header by the system, on behalf of the process. The label content is normally determined by the security level of the process. At the receiving end, packets are rejected for reception by another process unless deemed appropriate by the system, which compares the label with the label appropriate to the receiving process. Privileged daemons such as inetd, which need to be able to handle incoming connection requests or data from processes at arbitrary levels, are an exception to this scheme. For such processes, label options need to be associated with connections and datagrams. The presentation was favorably received by the group, but no clear consensus emerged on exactly how the POSIX networking interface(s) would be impacted.

An issue emerged with respect to security and existing transport interfaces – in particular, XTI and sockets. XTI specifies Internet address family security label options based on MIL-STD 1777 version dated September 1983. 4.3 BSD allows a user to specify a choice of security label through the IPOPTIONS *setsocketopt*() request. However, MIL-STD 1777 has been updated via RFC 1038 ("Draft Revised IP Security Option," M. St. Johns, IETF, January 1988). An even later RFC is scheduled to be released in the near future with further changes in this area. The specifications are driven primarily by needs within U.S. government agencies.

The new (RFC 1038) protocol format specification is incompatible with the old. In addition, many vendors require a new, more extensible, IP security option for the commercial market; a consortium of vendors, including Sun, HP, Unisys, and others (at the moment, this group is called simply "the Consortium"), is addressing this need.

Also, neither XTI nor sockets specifies any restrictions on the use of label options. This may be a security hole: unrestricted users can "spoof" a higher level of security than they actually possess. For example, an "unrestricted" (low-level) process could specify that outbound data it writes to a network endpoint object be accompanied by a "classified" label, implying (to the remote system) that the data was sent by a process with a higher security level.

Finally, neither XTI nor sockets provides the ability to retrieve a label associated with an incoming UDP datagram in an atomic manner. XTI has no provisions for UDP labels at all.

In the light of these issues and recent developments DNI and SNI may need to track the standards governing security as they evolve, possibly offering a standard (and secure) interface to such features.

IPC

For historical reasons, both P1003.4 (real-time) and 1003.8 now find themselves working, independently, on IPC. We held a joint meeting with P1003.4 on IPC. The general concern was the divergent directions of the interfaces, given the overlapping user needs. There were specific differences in areas such as name resolution, options, and performance characteristics.

"Real time" IPC has two variants: one, an event-based version which simply allows passing a pointer to shared memory from one process to another; the other, a message-based version which allows data messages between sending and receiving processes. Both versions use the UNIX file system name space for rendezvousing; both versions use queues and allow various manipulations on the queues. The message-based version requires timestamps, has provisions for user-process-defined priorities and sender identification, and has several options to optimize data transfer. In contrast, DNI and SNI are both based on a simpler, data-stream paradigm, with no queue manipulations, timestamps, filesystem rendezvousing, user-defined priorities, or sender identification, and few options for data transfer optimization. DNI and SNI may include options for message boundary delimitation, and will use a more general rendezvousing mechanism (aka name server interface) than the UNIX file system.

Unresolved issues include these:

1. Whether it is desirable to rely on a UNIX filesystem name space for general-purpose internetwork IPC rendezvous, both because machines may be far apart, and because mounting each machine's filesystems from all others is impractical in a large network.
2. How timestamp information can be kept accurate over a network.
3. How to encourage more interaction between X/Open XNET, and other concerned parties, and P1003.8. (This should require only an education process, since these groups are already interacting with P1003.4.)
4. What direction to take on the interaction of IPC and networking. The P1003.4 IPC group seems to favor generalizing the IPC mechanism for networking. This currently clashes with the networking group on transparent file access, which is currently focusing on an NFS-supportable subset of P1003.1 file semantics, and has never adopted support of P1003.4 file semantics as a formal goal.
5. Whether it is feasible, given timing and balloting considerations, to form a joint group or offload IPC onto a networking group.

It seemed generally agreed that there should be closer relations between the real-time and networking groups in this area, and that needless differences should be minimized.

One feature from real-time IPC was adopted which should allow faster performance in DNI than in either XTI or sockets: "tear-away writes." These let a user process specify that it does not need to access a write/send buffer after a write/send operation, freeing the system to unmap the buffer from user space and schedule the buffer for DMA, thus avoiding the need for a buffer copy operation.

Naming

A name service interface working group was created at this meeting, and attracted a lot of attention, both in and out of P1003.8. We described specific needs of the DNI and SNI interfaces to the new working group at a joint meeting: simple name resolution, name registry (SNI only), and the ability to get path information for a given service. We also clarified our position that at least the simple name resolution was needed at or before the DNI/SNI full-use ballot, to avoid dependency and usability problems.

P1003.8/2 -> full POSIX committee

P1003.8/2, along with other P1003.8/x groups, is in the process of becoming a full POSIX committee (P1003.y). The P1003.8 structure will evolve to become a POSIX networking "steering committee," overseeing the efforts of each P1003.8/x group.

Steering committees are sometimes used in IEEE standards committees to structure related subgroups and join their forces whenever a concerted effort is needed to address a problem. They help ensure that redundant standards are not created and that each subgroup has a clear and unique focus. POSIX has no steering committees yet, and a minor precedent will be set if this new organization becomes formally adopted. (Other such steering committees, such as one for languages, are being contemplated and may appear in the near future.)

Language independence

The P1003 steering committee has decided that new POSIX standards (with a few exceptions) will be specified in a language-independent manner, with at least one specific language binding. (Typically, one expects this will be C.)

P1003.8 is, thus, required to comply with the P1003 steering committee decision in this regard, and the P1003.8/x networking standards will be issued in a form that includes a language-independent specification.

Bytes versus octets

Neither POSIX nor the C standard specifies the number of bits in a byte. The number is system-dependent and accessible to a user process as CHAR_BIT, which according to the C language standard has a minimum value of 8. In networking this specification is insufficient to guarantee complete and formal interoperability, since (if an interface is specified in bytes) one system's notion of a byte may differ from another's – at least, in principle. Thus, most formal networking standards avoid the use of the term byte in favor of octet, implying an ordered set of eight bits.

POSIX data-transfer operations are defined exclusively in terms of bytes, not octets. For POSIX to be interoperable in the networking sense, either POSIX must change to octets, some relatively ugly solutions must be adopted, or some simplifying assumptions should be made whenever networking may be involved. The issue probably affects network IPC, and seems like it could also affect other areas – the most likely candidates being transparent file access and data archival.

The problem has been noted by P1003.8 at large, but not yet specifically addressed. Informal polls conducted at POSIX meetings indicate that most, and perhaps all, current vendors use eight-bit bytes. The ultimate solution may be to use weasel-wording equivalent to the assumption that interoperating systems will all use eight-bit bytes.

IEEE 1003.11: Application Transaction Processing Update

Bob Snead <bobs@ico.isc.com> reports on the July 10-14, 1989 meeting in San Jose, California:

1003.11 (application transaction processing, or TP) is one of two recently approved working groups – the other being P1003.10 (supercomputing) – whose charter is to write an application environment profile (AEP). A profile is simply a list of pointers to existing standards within the POSIX OSE (Open System

Environment). Where the group finds functionality missing from this set of standards, the group may either commission its definition by some other POSIX group or write a new PAR to request that IEEE create a standard in the area.

This was our first meeting as 1003.11; the previous three meetings were as a study group. This study group was formed last year at the Ft. Lauderdale meeting to investigate the feasibility of extending POSIX into transaction processing. In those first three meetings there was consensus that POSIX should address transaction processing.

At this point, the TP group is reviewing existing standards in detail to find out what's already been done. To this end, they have split into two subgroups, one to review models, the other to search out and review other relevant standards. There seems to be some consensus that once we understand what is available, there will still be new interfaces to define.

TP under UNIX is currently sort of a funny domain. Database vendors believe that transaction processing is theirs. They build TP primitives into their products that let application developers define transactions over modifications to data. More and more UNIX application developers want, instead, to write applications that bind a group of modifications to data managed by assorted vendors' products, including multiple databases, screen managers, and file systems. Sensing this need, X/Open boldly chartered a group to define such services. In addition, ISO, some time ago, recognized the need for services to define transactions which span heterogeneous open systems, and began a group to define such services. ISO also has groups defining CCR (Commitment, Concurrency, and Recovery) and RDA (Remote Data Access), each of which is an essential part of TP, especially distributed TP.

Both efforts are pretty far along. X/Open has defined a model and a set of interfaces but, since they are not a real standards body, referencing their work may present some problems for P1003.11. The ISO group recently resolved all outstanding objections to their model, services, and protocols. What remains for us then is to place the relevant portions of their work into a POSIX framework, filling in the holes.

Dear Reader,

We would like to know if you found the foregoing reports informative. Should we continue to publish these in *;login:*? Please send your suggestions to the editor, *ellie@usenix.org*.

Out-of-Print USENIX Proceedings Now Available

The Association has photocopied, bound sets of most of its past workshop and conference proceedings available for purchase.

CONFERENCE		COST	FOREIGN POSTAGE
San Diego	1983 Winter	$28.00	$15.00
Toronto	1983 Summer	32.00	20.00
Washington D.C.	1984 Winter	25.00	15.00
Salt Lake City	1984 Summer	29.00	20.00
Dallas	1985 Winter	15.00	10.00
Portland	1985 Summer	45.00	25.00
Denver	1986 Winter	25.00	15.00
Atlanta	1986 Summer	37.00	20.00
Phoenix	1987 Summer	35.00	20.00
Dallas	1988 Winter	26.00	15.00
San Francisco	1988 Summer	29.00	20.00
WORKSHOP			
Systems Admin. I	1987 Philadelphia	4.00	5.00
Systems Admin. II	1988 Monterey	8.00	5.00
Security	1988 Portland	7.00	5.00
Graphics II	1985 Monterey	7.00	5.00
NOT AVAILABLE			
Delaware Conference			
Graphics I Workshop			

Orders to U.S. and Canada are shipped via printed matter. Foreign orders are shipped via air printed matter. Please allow 10-14 days for delivery. Shipment by UPS, first class, or airmail is available upon request. Check, purchase orders, or payments by VISA/MC are accepted. (For charge orders please include card number, expiration date, and your signature.) Prepayment is required. Send orders to:

USENIX Association
2560 Ninth Street
Suite 215
Berkeley, CA 94710

Reprints of individual papers from all proceedings are available for $5.00 each; contact the Association Office.

ORDER FORM

CONFERENCE & WORKSHOP PROCEEDINGS

Qty	Proceeding		Add'l Unit Price	Total	Foreign Postage	Total
___	Graphics Workshop V	Nov. '89	$18	$_______	$10	$_______
___	Distributed and Multiprocessor Workshop	Oct. '89	30	$_______	20	$_______
___	Large Installation Sys. Admin. III Workshop	Sept. '89	13	$_______	9	$_______
___	Baltimore Conference	June '89	20	$_______	15	$_______
___	UNIX Transaction Processing Workshop	May '89	12	$_______	8	$_______
___	Software Management Workshop	Apr. '89	20	$_______	15	$_______
___	San Diego Conference	Feb. '89	30	$_______	20	$_______
___	C++ Conference	Oct. '88	30	$_______	20	$_______
___	UNIX and Supercomputers Workshop	Sept. '88	20	$_______	15	$_______
___	C++ Workshop	Nov. '87	30	$_______	20	$_______
___	Graphics Workshop IV	Oct. '87	10	$_______	15	$_______
___	Washington DC Conference	Jan. '87	10	$_______	20	$_______
___	Graphics Workshop III	Dec. '86	10	$_______	15	$_______

Total price of Proceedings ________

Calif. residents only add applicable sales tax ________

Total foreign postage ________

Total enclosed ________

* Discounts are available for bulk orders. Please inquire.

PAYMENT OPTIONS

□ Check enclosed payable to USENIX Association. □ Purchase order enclosed.

□ Please charge my: □ Visa □ MasterCard

Account # ______________________________ Exp.Date ____________________

Signature ____

Overseas? Please make your payment in U.S. currency by one of the following:

- Charge (Visa, MasterCard, or foreign equivalent)
- International postal money order
- Check - issued by a local branch of a U.S. Bank

10/89

Shipping Information

Orders to U.S. and Canada are shipped via printed matter. Please allow 2 weeks for delivery. Foreign orders are shipped via air printed matter. Please allow 10-14 days for delivery. Shipment by UPS, first class, or airmail is available upon request.

Ship to: ____________________________

____

____

____

Please mail this order form to:

USENIX Association
2560 Ninth Street
Suite 215
Berkeley, CA 94710

2560 Ninth Street • Suite 215 • Berkeley, CA • 94710 • 415/528-8649 • FAX 415/548-5738 • *office@usenix.org*

Final Printing of 4.3BSD Manuals

The 4.3BSD manuals offered by the USENIX Association[†] are now available to everyone.

The 4.3BSD manual sets are significantly different from the 4.2BSD edition. Changes include many additional documents, better quality of reproductions, as well as a new and extensive index. All manuals are printed in a photo-reduced 6″×9″ format with individually colored and labeled plastic "GBC" bindings. All documents and manual pages have been freshly typeset and all manuals have "bleed tabs" and page headers and numbers to aid in the location of individual documents and manual sections.

A new Master Index has been created. It contains cross-references to all documents and manual pages contained within the other six volumes. The index was prepared with the aid of an "intelligent" automated indexing program from Thinking Machines Corp. along with considerable human intervention from Mark Seiden. Key words, phrases and concepts are referenced by abbreviated document name and page number.

While two of the manual sets contain three separate volumes, you may only order complete sets.

The costs shown below do not include applicable taxes or handling and shipping from the printer in New Jersey, which will depend on the quantity ordered and the distance shipped. Those charges will be billed by the printer (Howard Press).

To order, return a completed "4.3BSD Manual Reproduction Authorization and Order Form" to the USENIX office along with a check or purchase order for the cost of the manuals.

Manual	Cost*
User's Manual Set (3 volumes) User's Reference Manual User's Supplementary Documents Master Index	$25.00/set
Programmer's Manual Set (3 volumes) Programmer's Reference Manual Programmer's Supplementary Documents, Volume 1 Programmer's Supplementary Documents, Volume 2	$25.00/set
System Manager's Manual (1 volume)	$10.00

* Not including postage and handling or applicable taxes.

[†] Tom Ferrin of the University of California at San Francisco, a former member of the Board of Directors of the USENIX Association, has overseen the production of the 4.2 and 4.3BSD manuals.

4.3BSD Manual Reproduction Authorization and Order Form

This page may be duplicated for use as an order form

Purchase Order No.: ____________

Date: ____________

Pursuant to the copyright notice as found on the rear of the cover page of the UNIX®/32V Programmer's Manual stating that

> "Holders of a UNIX®/32V software license are permitted to copy this document, or any portion of it, as necessary for licensed use of the software, provided this copyright notice and statement of permission are included,"

I hereby appoint the USENIX Association as my agent, to act on my behalf to duplicate and provide me with such copies of the Berkeley 4.3BSD Manuals as I may request.

Signed: ____

Ship to:

Name: ________________________

____

____

____

____

Phone: ________________________

Billing address, if different:

Name: ________________________

____

____

____

____

Phone: ________________________

The prices below **do not** include shipping and handling charges or state or local taxes. All payments must be in US dollars drawn on a US bank.

4.3BSD User's Manual Set (3 vols.)	______	at $25.00 each	=	$__________
4.3BSD Programmer's Manual Set (3 vols.)	______	at $25.00 each	=	$__________
4.3BSD System Manager's Manual (1 vol.)	______	at $10.00 each	=	$__________
Total	______			$__________

[ ] Purchase order enclosed; invoice required.
(Purchase orders **must** be enclosed with this order form.)

[ ] Check enclosed for the manuals: $__________
(Our printer will send an invoice for the shipping and handling charges and applicable taxes.)

Send your check or purchase order with this order form to:

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710

Local User Groups

The Association will support local user groups by doing a mailing to assist the formation of a new group and publishing information on local groups in *;login:*. At least one member of the group must be a current member of the Association. Send additions and corrections to *login@usenix.org*.

CA - Fresno: the Central California UNIX Users Group consists of a *uucp*-based electronic mailing list to which members may post questions or information. For connection information:

Educational and governmental institutions:
Brent Auernheimer (209) 294-4373
brent@CSUFresno.edu or csufres!brent

Commercial institutions or individuals:
Gordon Crumal (209) 875-8755
csufres!gordon (209) 298-8393

CO - Boulder: the Front Range UNIX Users Group meets monthly at different sites.

Steve Gaede (303) 447-8586
636 Arapahoe Ave., #10
Boulder, CO 80302

FL - Coral Springs:

S. Shaw McQuinn (305) 344-8686
8557 W. Sample Road
Coral Springs, FL 33065

FL - Fort Lauderdale/Miami: The South Florida UNIX Users Group meets the 2nd Tuesday of each month.

Tony Vincent, John McLaughlin (305) 776-7770
{sun,novavax,gould}!sunvice!tony
jmclaughlin@sun.COM

John O'Brien (305) 475-7633
gatech!uflorida!novavax!john

Don Joslyn (305) 476-6415
gatech!uflorida!novavax!rm1!don

FL - Jacksonville/Northeast: UNIX Users of Jacksonville (uujax) meets the 2nd Thursday of each month.

Tom Blakely (904) 646-2820
uflorida!unf7!tfb

Emilie Olsen (904) 390-3621

FL - Melbourne: the Space Coast UNIX Users Group meets at 8pm on the 3rd Wednesday of each month at the Florida Institute of Technology.

Bill Davis (407) 242-4449
bill@ccd.harris.com

FL - Orlando: the Central Florida UNIX Users Group meets the 3rd Thursday of each month.

Mike Geldner (305) 862-0949
codas!sunfla!mike

Ben Goldfarb (305) 275-2790
goldfarb@hcx9.ucf.edu

Mikel Manitius (305) 869-2462
{codas,attmail}!mikel

FL - Tampa Bay: the Tampa UNIX Users Group meets the 1st Thursday of each month in Largo.

Bill Hargen (813) 530-8655
uunet!pdn!hargen

George W. Leach (813) 530-2376
uunet!pdn!reggie

GA - Atlanta: meets on the 1st Monday of each month in White Hall, Emory University.

Atlanta UNIX Users Group
P.O. Box 12241
Atlanta, GA 30355-2241

Marc Merlin (404) 442-4772
Mark Landry (404) 365-8108

MI - Detroit/Ann Arbor: The SouthEastern Michigan Sun Local Users Group meets jointly with the Nameless UNIX Group on the 2nd Thursday of each month in Ann Arbor.

Steve Simmons home: (313) 426-8981
scs@lokkur.dexter.mi.us office: (313) 769-4086

K. Richard McGill
rich@sendai.ann-arbor.mi.us

Bill Bulley
web@applga.uucp

MI - Detroit/Ann Arbor: dinner meetings the 1st Wednesday of each month.

Linda Mason (313) 855-4220
michigan!/usr/group
P.O. Box 189602
Farmington Hills, MI 48018-9602

MN - Minneapolis/St. Paul: meets the 1st Wednesday of each month.

UNIX Users of Minnesota
17130 Jordan Court
Lakeville, MN 55044

Robert A. Monio (612) 895-7007
pnessutt@nis.mn.org

MO - St. Louis:

St. Louis UNIX Users Group
Plus Five Computer Services
765 Westwood, 10A
Clayton, MO 63105

Eric Kiebler (314) 725-9492
plus5!sluug

NE - Omaha: meets the 2nd Thursday of each month.

/usr/group nebraska
P.O. Box 44112
Omaha, NE 68144

Kent Landfield (402) 291-8300
kent@ugn.uucp

New England - Northern: meets monthly at different sites.

Peter Schmitt (603) 646-2999
Kiewit Computation Center
Dartmouth College
Hanover, NH 03755

decvax!dartvax!nneuug-contact

NJ - Princeton: the Princeton UNIX Users Group meets monthly.

Pat Parseghian (609) 452-6261
Dept. of Computer Science
Princeton University
Princeton, NJ 08544

pep@Princeton.EDU

NY - New York City:

Unigroup of New York
G.P.O. Box 1931
New York, NY 10116

Ed Taylor (212) 513-7777
{attunix,philabs}!pencom!taylor

New Zealand:

New Zealand UNIX Systems User Group
P.O. Box 13056
University of Waikato
Hamilton, New Zealand

OK - Tulsa:

Pete Rourke
$USR
7340 East 25th Place
Tulsa, OK 74129

PA - Philadelphia: the UNIX SIG of the Philadelphia Area Computer Society (PACS) meets the morning of the 3rd Saturday of each month.

G. Baun, UNIX SIG
c/o PACS
Box 312
La Salle University
Philadelphia, PA 19141

rutgers!{bpa,cbmvax}!
temvax!pacsbb!{gbaun,whutchi}

TX - Dallas/Fort Worth:

Dallas/Fort Worth UNIX Users Group
Seny Systems, Inc.
5327 N. Central, #320
Dallas, TX 75205

Jim Hummel (214) 522-2324

TX - Houston: the Houston UNIX Users Group (Hounix) meets the 3rd Tuesday of each month.

Hounix answering machine (713) 684-6590

Bob Marcum, president (713) 270-8124

Chuck Bentley, vice-president (713) 789-8928
chuckb@hounix.uucp

TX - San Antonio: the San Antonio UNIX Users (SATUU) meets the 3rd Thursday of each month.

Jeff Mason (512) 494-9336
Hewlett Packard
14100 San Pedro
San Antonio, TX 78232

gatech!petro!hpsatb!jeff

WA - Seattle: meets monthly.

Bill Campbell (206) 232-4164
Seattle UNIX Group Membership Information
6641 East Mercer Way
Mercer Island, WA 98040

uw-beaver!tikal!camco!bill

Washington, D.C.: meets the 1st Tuesday of each month.

Washington Area UNIX Users Group
2070 Chain Bridge Road, Suite 333
Vienna, VA 22180

Samuel Samalin (703) 448-1908

Management Committee Meeting

8th August, 1989

MINUTES

The meeting opened at 3:15pm with all committee members present, namely: President Greg Rose (GR), Secretary Tim Roper (TR), Treasurer Michael Tuke (MT), Pat Duffy (PD), Peter Barnes (PB) and John Carey (JC). Also present was the new AUUGN Editor David Purdue (DP) and retiring committee members Rich Burridge (RB) and Frank Crawford (FC). Wael Foda (WF) attended for some of the time.

1. Apologies

From Chris Maltby who was giving a tutorial, Peter Barnes who was occupied with AUUG89 programme matters, Tim Segall who was unable to be present.

2. Minutes of last meeting (10th May, 1989)

Moved JC, seconded RB **That the minutes be accepted.** Carried.

3. Business arising from Minutes

Re 5(e), Vol 2 No 1 had been mailed, Vol 2 No 2 was in the pipeline. We have paid for 85 copies of Vol 2 Nos 2, 3 and 4.

Re 5(g), the results had been published in AUUGN Vol 10 No 3.

Re 8, Direct mailing been done at a cost of $9343.40: 143 responses had been received as of 4/8/89. The AUUG stand was going ahead.

Re 9, TR had written to ACMS who had requested a mailing list of Institutional members which was to be supplied asap.

Re 10, no progress on membership cards.

Re 15, Forms with new fees had been printed in the old style at a cost of $302.40.

8. 1989 Winter Conference and Exhibition

GR reported that he had been contact at 1:30pm today by an AUUG89 author with a request that certain parts of their paper be removed from the proceedings. GR had pointed out that copies of the proceedings were already out but that he woould discuss the matter with the committee. At the time of the meeting about 50 copies had been circulated. Moved TR, seconded PD **That GR advise the author that AUUG is not prepared to compromise the quality of the conference and proceedings by mutilating the proceedings especially given that it has already been widely distributed.** Carried unanimously. PB noted that the author had previously mentioned that he would need clearance and was looking after it.

WF was invited to report on arrangements for AUUG89. There were 375 advance registrations compared to 280 for AUUG88. The exhibiton was fully occupied. There was good sponsoring and advertising support.

4. President's Report
   GR thanked everyone responsible for AUUG89.

   GR had met with Al Nemeth, President of USENIX. Both were keen to see closer relations between USENIX and AUUG. USENIX would like information on services such as TNT Mailfast for expedient delivery of ;login: to Australian subscribers.

   Moved JC/PB **That the President's report be accepted.** Carried.

5. Secretary's Report
   TR reported on the following.

   (a) The Inauuugral Software Distribution
   Approximately thirty-six orders had now been processed and we still were not accepting any more.

   (b) USENIX Proceedings
   24 copies of San Diego had been sold.
   30 copies of Baltimore, cost $US550, 2 sold at $A30.

   (c) UniForum Product Directories
   101 copies ordered at a cost of $US1792.75 plus $US1288.70 shipping for distribution to Institutional members. Any spares would be sold to members at cost.

   (d) Membership
   As of 6/8/89 there were 0 lifers, 313 ordinaries, 10 students, 80 institutionals and 18 subscribers.

   (e) Direct Mail Campaigns
   Approximately 1060 past members and attendees had been mailed. Attributable responses were 2 changes of address, 9 new institutional members, 18 new ordinary members and 68 "left address".

   (f) Stationery
   Business cards had been printed for PB, PD, JC and DP.

   (g) AUUG Stand at AUUG89
   Proceeding well. One person had been hired to staff it during exhibition hours at an approximate cost of $20 per hour. There would be back issues of AUUGN and USENIX San Diego and Baltimore proceedings for sale. Also membership information and forms. Printing had cost $121.20 for 1000 price lists and 750 information sheets. Volunteers were required to assist with staffing the stand during breaks in the conference programme. A roster was drawn up. Face saver equipment had been arranged by James Ashton who had been given complimentary registration in return for setting it up and training staff in its operation.

   Prentice Hall had asked to sell books from the AUUG stand. Instead they had been invited to supply samples and order forms which they had accepted and were to offer a 20% discount. They had not taken up an offer of limited time for sales.

Moved JC/PD **That the Secretary's report be accepted.** Carried.

6. Treasurer's Report

MT reported that the books had been audited at a cost of $1500.00 which was greater than the approved expenditure. This report was to be presented at the AGM. Moved TR, seconded PD **That this expenditure be authorised.** Carried.

Move MT, seconded PD **That the individual signing limit be increased to $500; that the signatories be changed to GR, MT, TR and JC; that an encashment authority be established at the Mordialloc branch of the Commonwealth Bank suitably endorsed to cover overseas payments only.** Carried.

Moved TR/JC **That the Treasurer's report be accepted.** Carried.

7. Retiring AUUGN Editor's Report

JC thanked PB for his efforts in producing the AUUG89 proceedings issue. JC tabled the 240 page issue reporting that 500 copies had been shipped to the conference and 250 were being kept in Melbourne for mailing.

DP had looked at the new Australia Post regulations on registered publications. It appeared that AUUG's situtation was unchanged except possibly with respect to library subscribers.

JC stated that a formal release form was need for all articles published in AUUGN.

GR acclaimed very highly the diligence and service of the retiring editor (JC).

9. 1990 Summer Meetings

General discussion. The 2nd and 3rd weeks of February were generally agreed on. Immediate need was for a coordinator.

10. 1990 Winter Conference and Exhibition

The following responsibilities were assigned:

CM	guest speakers
JC	programme
?	tutorials
PD	publicity
?	convenor

A target attendance of 800 was agreed on.

11. Membership Periods

Moved MT, seconded CM **That memberships which currently end on a day of the month other than the first be made current to the first of the following month.** Carried.

Move MT, seconded JC **That payments for membership be accepted for up to three years at a time.** Carried.

It was mentioned that the forms should specify the validity period of the fees since a lot of applications are received on out of date forms accompanied by the old fees.

The President adjourned the meeting at 7pm until 5pm on 11th October, 1989 in the same place.

The meeting reconvened at 5:30pm on 11th October, 1989. Present were DP, FC, CM, JC, PB, GR, MT, TR, PD and WF.

8. 1989 Winter Conference and Exhibition (cont.)
There had been 440 delegates plus 855 exhibition visitors. All but one exhibitor had been happy. It was agreed that this was a good result. It was mentioned that AUUG needed a larger stand particularly with the face saver equipment.

10. 1990 Winter Conference and Exhibition (cont.)
WF reported that 15 potential exhibitors had not been accomodated at AUUG89. The exhibition space at the Southern Cross for 1990 was already 50% taken. WF proposed the World Congress Centre being built in Melbourne along side the World Trade Centre. He had made a tentative booking of 4th - 7th September, 1990. A subcommittee of whoever could make it was appointed to inspect and recommend on this site. A breakeven attendance of 600 was suggested by WF.

It was agreed that the brochure had to be out befre June 1990.

12. Promotion of AUUG
Deferred.

13. Secretarial Assistance
Deferred.

14. Constitutional Changes
Deferred.

15. Other Business

(a) 1991 Conference and Exhibition
The AGM had requested that alternative proposal be made for 1991, one for Sydney and one for somewhere other than Sydney or Melbourne. WF agreed to do this.

GR, TR, CM, PB and WF had inspected Darling Harbour this week. The exhibition halls were hugely large and undesirable. There was space for about 100 stands in the banquet hall. WF to make tentative booking for 12, 13, 14 and 15th [of what month?].

(b) TR tabled a petition signed by 18 members wishing to form a chapter entitled SESSPOOLE. Moved TR, seconded CM **That the petition be accepted.** Carried.

(c) TR noted that he had not tabled correspondance in his Secretary's report and in view of the hour offered to detail it in the minutes. This was accepted.

16. Next Meeting
It was resolved that the next meeting should be held in Melbourne at a time and place to be decided by the Secretary.

AUUGN Back Issues

Here are the details of back issues of which we still hold copies. All prices are in Australian dollars and include surface mail within Australia. For overseas surface mail add $2 per copy and for overseas airmail add $10 per copy.

pre 1984	Vol 1-4	various	$10 per copy
1984	Vol 5	Nos. 2,3,5,6	$10 per copy
		Nos. 1,4	unavailable
1985	Vol 6	Nos. 2,3,4,6	$10 per copy
		No. 1	unavailable
1986	Vol 7	Nos. 1,4-5,6	$10 per copy
		Nos. 2-3	unavailable
			(Note 2-3 and 4-5 are combined issues)
1987	Vol 8	Nos. 1-2,3-4	unavailable
		Nos. 5,6	$10 per copy
1988	Vol 9	Nos. 1,2,3	$10 per copy
		Nos. 4,5,6	$15 per copy
1989	Vol 10	Nos. 1-6	$15 per copy

Please note that we do not accept purchase orders for back issues except from Institutional members. Orders enclosing payment in Australian dollars should be sent to:

AUUG Inc.
Back Issues Department
PO Box 366
Kensington NSW
Australia 2033

AUUG

Membership Categories

Once again a reminder for all ‘‘members’’ of AUUG to check that you are, in fact, a member, and that you still will be for the next two months.

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

The membership categories are:

Institutional Member
Ordinary Member
Student Member
Honorary Life Member

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

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

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

Are **you** an AUUG member?

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

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

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

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

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

Check that your membership isn't about to expire (or worse, hasn't expired already). Ask your colleagues if they received this issue of AUUGN, tell them that if not, it probably means that their membership has lapsed, or perhaps, they were never a member at all! Feel free to copy the membership forms, give one to everyone that you know.

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

As a service to members, AUUG has arranged to accept payments via credit card. You can use your Bankcard (within Australia only), or your Visa or Mastercard by simply completing the authorisation on the application form.

THIS PAGE INTENTIONALLY LEFT BLANK

AUUG

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

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

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

AUUG Membership Secretary
P O Box 366
Kensington NSW 2033
Australia

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

I, ... do hereby apply for

□ Renewal/New*	Membership of the AUUG	$78.00	
□ Renewal/New*	Student Membership	$45.00	(note certification on other side)
□ International Surface Mail		$20.00	
□ International Air Mail		$60.00	(note local zone rate available)
Total remitted		AUD$______	
		(cheque, money order, credit card)	

* Delete one.

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

Date: __/__/__ Signed: ____________________

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

For our mailing database - please type or print clearly:

Name: .... Phone: .... (bh)

Address: .... (ah)

.... Net Address: ....

....

.... *Write "Unchanged" if details have not*

.... *altered and this is a renewal.*

Please charge $_____ to my □ Bankcard □ Visa □ Mastercard.

Account number: _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _. Expiry date: __/__.

Name on card: ______________________________ Signed: ____________________

Office use only:

Chq: bank ________ *bsb* ___-___ *a/c* ______________ *#* ______________

Date: __/__/__ *$* *CC type* ___ *V#* ___________

Who: _____________ *Member#* ________

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

I, ... certify that

.... *(name)*

is a full time student at .... *(institution)*

and is expected to graduate approximately ___/___/___.

Title: ______________________________ Signature: __________________

AUUG

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

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

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

AUUG Membership Secretary
P O Box 366
Kensington NSW 2033
Australia

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

.... does hereby apply for

□ New/Renewal* Institutional Membership of AUUG	$325.00
□ International Surface Mail	$ 40.00
□ International Air Mail	$120.00
Total remitted	AUD$______
	(cheque, money order, credit card)

* Delete one.

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

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

Date: __/__/__ Signed: ____________________

Title: ____________________

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

For our mailing database - please type or print clearly:

Administrative contact, and formal representative:

Name: .... Phone: .... (bh)

Address: .... (ah)

.... Net Address: ....

....

.... *Write "Unchanged" if details have not*

.... *altered and this is a renewal.*

Please charge $_____ to my/our □ Bankcard □ Visa □ Mastercard.

Account number: _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _. Expiry date: __/__.

Name on card: ______________________________ Signed: ____________________

Office use only: **Please complete the other side.**

Chq: bank ________ *bsb* ___-___ *a/c* ____________ *#* ____________

Date: __/__/__ *$* *CC type* ___ *V#* __________

Who: ____________ *Member#* ________

Please send newsletters to the following addresses:

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

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

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

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

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

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

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

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

□ System V.3 source
□ System V.2 source
□ System V source
□ System III source
□ 4.2 or 4.3 BSD source
□ 4.1 BSD source
□ V7 source
□ Other *(Indicate which)* ....

□ System V.3 binary
□ System V.2 binary
□ System V binary
□ System III binary

AUUG

Application for Newsletter Subscription
Australian UNIX* systems Users' Group.

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

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

AUUG Membership Secretary
P O Box 366
Kensington NSW 2033
Australia

- Please don't send purchase orders — perhaps your purchasing department will consider this form to be an invoice.
- Foreign applicants please send a bank draft drawn on an Australian bank, or credit card authorisation, and remember to select either surface or air mail.
- Use multiple copies of this form if copies of AUUGN are to be dispatched to differing addresses.

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

Name: .... Phone: .... (bh)

Address: .... (ah)

.... Net Address: ....

....

.... *Write "Unchanged" if address has*

.... *not altered and this is a renewal.*

For each copy requested, I enclose:

□ Subscription to AUUGN	$ 90.00
□ International Surface Mail	$ 20.00
□ International Air Mail	$ 60.00

Copies requested (to above address) ________

Total remitted AUD$______

(cheque, money order, credit card)

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

Please charge $_____ to my □ Bankcard □ Visa □ Mastercard.

Account number: _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _. Expiry date: __/__.

Name on card: ______________________ Signed: ______________________

Office use only:

Chq: bank ________ *bsb* ___-___ *a/c* ____________ *#* ____________

Date: __/__/__ *$* *CC type* __ *V#* __________

Who: ____________ *Subscr#* ________

AUUG

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

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

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

AUUG Membership Secretary
P O Box 366
Kensington NSW 2033
Australia

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

Old address (or attach a mailing label)

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

New address (leave unaltered details blank)

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

Office use only:

Date: ___/___/___

Who: ______________ *Memb#* _________