

# :login:

September / October 1992 Vol. 17, No. 5

## From the Editor's Desk

My trip to BizMart yesterday (for FAX paper) was rudely interrupted by a headline screaming off the pages of the September, 1992 BYTE magazine: "Is UNIX Dead?"

It seems that experts now believe that Windows-NT will kill the UNIX operating system dead in short order.

I admit to being a bit prejudiced but it seems to me that this journalistic tomfoolery is getting old. The "death of UNIX" has been predicted for over a decade. We now have an operating system that's available on basically every processor with memory management (or even pseudo-memory-management) from Cray supercomputers to the lowly desktop (though I benchmark the 486/50's at over 25 MIPS). Can you imagine running Windows-NT on your IBM mainframe? I can't. (I have been known to have little vision in this area, though.) I hope that my editorial compatriots will realize that the desktop market is a large one - but it is not 100% of the computer market.

This issue includes terrific articles and columns. You may wish to check out the summary of the File System Workshop written by Drew Perkins. I found that it brought out the flavor of the workshop and its papers - and that I was able to learn basically what went on with only 15 minutes of reading. I hope to be able to publish more summaries of workshops in future issues.

The San Diego program has been set and is included here. I think it has something for everyone. It, too, includes summaries of workshops. San Diego is a great venue and the hotel rates are the lowest in a long time - come on out for what looks to be a great conference.

How about those presidential candidates? Isn't it amazing? Whichever way you lean, please do make sure that you exercise your right to vote in the November elections. If we all vote, we can all complain afterward.

RK

The closing date for submissions to the next issue of *:login:* is October 23, 1992.

## Association News

USENIX Online Index .....	3
File Systems Workshop Report .....	4
<i>Drew Perkins</i>	
Campus Reps Sought .....	7
Attendee Survey Results .....	8
Call for Tutorials .....	10
USENIX Committees.....	11

## SAGE News

SAGE Membership .....	12
SAGE Working Groups .....	13
ABCs of UNIX .....	14
SAGE Views .....	15
SAGE Book Review.....	18

*Steve Simmons*

SAGE Report on World Conference .....	19
---------------------------------------	----

## Features

What's Out There? .....	20
<i>Jeff Kellem</i>	
Millionaires .....	23
<i>Peter Collinson</i>	
In Favor of Student Grants .....	24
<i>George Neville-Neill</i>	
Autoloading in Perl .....	25
<i>Tom Christiansen</i>	
Using GraphTab .....	28
<i>Mark Horton</i>	
An Update on UNIX-Related Standards Activities..	32
<i>Stephen Walli</i>	

Obtaining GNU Software .....	41
------------------------------	----

## Book Reviews

The Bookworm .....	42
<i>Peter H. Salus</i>	
Advanced Programming in the UNIX Environment.....	44
<i>Peter Collinson</i>	
PrimeTime Freeware .....	45
The Whole Internet Guide and Catalog .....	46
<i>Billy Barron</i>	

## Announcements

Winter 1993 Conference .....	48
UNIX Applications Development Symposium.....	51
Mach Symposium .....	53
SEDMS IV Preliminary Call .....	55
LISA Groups .....	56
Publications Order Forms.....	57
Local User Groups .....	60
Computing Systems Coming Attractions.....	62
Calendar of Events.....	63



# General Information

*;login:* is the official newsletter of the USENIX Association.

*;login:* (ISSN 1044-6397) Volume 17, Number 5 (September/October 1992) is published bimonthly by the USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710. \$24 of each member's annual dues is for an annual subscription to *;login:*. Subscriptions for non-members are \$50 per year. Second-class postage paid at Berkeley, CA and additional offices. POSTMASTER: Send address changes to *;login:*, USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.

## Contributions Solicited

You are encouraged to contribute articles, book reviews, and announcements to *;login:*. Send them via email to *login@usenix.org* or through the postal system to the Association office. Send SAGE material to *bigmac@erg.sri.com*. The Association reserves the right to edit submitted material. Any reproduction of this newsletter in its entirety or in part requires the permission of the Association and the author(s).

## Editorial Staff

Rob Kolstad, Editor <*kolstad@usenix.org*>  
Ellie Young, Staff Editor <*ellie@usenix.org*>  
Carolyn S. Carr, Managing Editor and Typesetter  
<*carolyn@usenix.org*>  
Stephe Walli, Standards Report Editor  
<*stephe@mks.com*>  
Bryan MacDonald, SAGE Editor  
<*bigmac@erg.sri.com*>  
Michelle Dominijanni, Copy Editor

## Membership and Publications

USENIX Association  
2560 Ninth Street, Suite 215  
Berkeley, CA 94710  
Telephone: 510/528-8649  
FAX: 510/548-5738  
Email: <*office@usenix.org*>

Copyright © 1992 USENIX Association. The Association acknowledges all trade references made herein.

*;login:* was produced on a SUN 3/50 with Framemaker 3.0 software and printed on recycled paper. ® Thanks to mt Xinu for technical assistance and laser output from their Apple Laserwriter IIg.

## Conferences, Workshops & Symposia

Judith F. DesHarnais, Conference Coordinator  
USENIX Conference Office  
22672 Lambert Street, Suite 613  
El Toro, CA 92630  
Telephone: 714/588-8649  
FAX: 714/588-9706  
<*conference@usenix.org*>

## Tutorials

Daniel V. Klein, Tutorial Coordinator  
Telephone: 412/421-2332  
<*dvk@usenix.org*>

## Supporting Members

Digital Equipment Corporation  
Frame Technology, Inc.  
Matsushita Graphic Communication Systems  
mt Xinu  
Open Software Foundation  
Quality Micro Systems  
Rational  
Sun Microsystems, Inc.  
Sybase, Inc.  
UNIX System Laboratories, Inc.  
UUNET Technologies, Inc.

## USENIX Association Board of Directors

If a member wishes to communicate directly with the USENIX Board of Directors, he or she may do so by writing to *board@usenix.org*.

### President:

Stephen C. Johnson <*scj@usenix.org*>

### Vice President:

Michael D. O'Dell <*mo@usenix.org*>

### Secretary:

Evi Nemeth <*evi@usenix.org*>

### Treasurer:

Rick Adams <*rick@usenix.org*>

### Directors:

Eric Allman <*eric@usenix.org*>  
Tom Christiansen <*tchrist@usenix.org*>  
Lori Grob <*grob@usenix.org*>  
Barry Shein <*bzs@usenix.org*>

## USENIX Association

Executive Director  
Ellie Young <*ellie@usenix.org*>

# USENIX Online Library and 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, and contains information about papers published in USENIX conference and workshop proceedings, newsletters, journal, and the like.

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 as mail, send mail to *library@uunet.uu.net* with "send bibliography" as the contents of your message.

To get it via ftp from *ftp.uu.net*, login as "anonymous" with your email address as the password. Then:

```
ftp> cd library
250 CWD command successful.
ftp> get bibliography
```

This help file can be retrieved with "send help" or as the ftp file "help.bibliography".  
(There is no person associated with the library address and it will never be read by human eyes.)

## How to Access Information

To build the indices so you can easily access information, run "idxbib" on the bibliography: *idxbib file.name*. You can then pull information from the file by running "lookbib". You can either build refer files or run lookbib interactively.

For example, the following command would put all entries which refer to Smith into a file called "stuff":

```
echo smith | lookbib bibliography > stuff.
```

Or you could interact with the index by saying:  
lookbib bibliography

It will ask you if you want instructions when it starts, answer yes. Then at the prompt, for example: >smith

## To Get an Online Paper

As of this date, we have not yet set up the online papers. When this capability is provided, we will announce it on the net and these instructions will be updated with retrieval information.

## Publications Indexed

### USENIX:

Conference proceedings, workshop and symposia proceedings, *Computing Systems* journal, newsletter

EurOpen (formerly EUUG - European Unix Users Group):

Conference proceedings, newsletter (1982-1989)

Other sources are being continually evaluated and will be included as deemed suitable.

## Fields Used in the Index

The standard bib/refer formats are used. These include:

A	Author (may be multiple entries)
T	Title of article
P	Page number(s)
W	Primary author's institution
I	Issuer/publisher
B	Conference proceedings or book title
J	Name of newsletter or journal
D	Date of publication or conference
C	Location of conference
V	Volume number
N	Nbr within volume
O	Other comments (e.g., "Abstract only")

These fields may be extended to include other information such as identifier for retrieval, keywords, online format of paper (PostScript, troff, etc.), language (if other than English), etc.

## More Information

For additional information about the online index and library, and /or instructions for donating papers, contact: *index@usenix.org*

Or write to:

USENIX Association  
2560 Ninth St., Suite 215  
Berkeley CA 94710

# File Systems Workshop Report

## Workshop on File Systems Ann Arbor, Michigan, May 21 - 22, 1992

by Drew D. Perkins  
*<ddp@andrew.cmu.edu>*

Approximately 160 people attended the File Systems workshop on the campus of the University of Michigan in Ann Arbor, Michigan on Thursday, May 21 and Friday, May 22, 1992. The workshop was sponsored by the USENIX Association and the University of Michigan's Center for Information Technology Integration (CITI).

Peter Honeyman from CITI served as the Program Chair and all-around workshop host. Mike Kazar from Transarc, Larry McVoy from Sun Microsystems, Mendel Rosenblum from Stanford University, and Liuba Shrira from MIT also served on the workshop Program Committee. Local arrangements were coordinated by Carol Kamm from CITI and Judy DesHarnais from USENIX.

### You Name It!

Thursday morning was divided into two sessions: You Name It!, a session of technical talks; and We Name It, a panel discussion session.

The first speaker, Vincent Cate from Carnegie Mellon University, described *Alex – A Global Filesystem*. This generic name does not do justice to its true usefulness. Alex is an NFS to Anonymous-FTP gateway. With Alex, users no longer need to be Internet experts to find and use information and files published on the Internet. Especially when combined with the now legendary archie system, files from around the world can be accessed with standard UNIX programs like *ls*, *cat* and *grep*, as simply as if they were in your local file system.

Alex is implemented as a UNIX server process and provides a standard NFS server interface. This allows it to be mounted from any existing NFS client. Alex also acts as a standard FTP client, allowing it to communicate with any existing FTP server. When Alex receives NFS requests, it translates these into their equivalent set of FTP operations. Since NFS is a stateless connectionless protocol, and FTP a stateful connection-oriented one, Alex caches a number of objects to achieve

good performance. These objects include host names, open logged-in FTP connections, directory information, and whole files.

Alex is more than just a good idea. It is already up and operational. Cate intends to distribute source code some time during the summer of 1992. Get the README file from *alex.sp.cs.cmu.edu* for availability. Alex is highly recommended for any Internet site.

The next talk was *The Prospero File System* by B. Clifford Neuman from the USC Information Sciences Institute. Neuman argued that the size of large global and inter-organizational file systems makes information very difficult to organize and find. For instance, how is a user supposed to know that *alex* can be found on *alex.sp.cs.cmu.edu*? Prospero attempts to solve this problem by allowing users to create and manage their own virtual systems. These virtual systems allow files and directories to be organized and cross-indexed by topic, rather than by location. So that every user is not required to organize their own world, virtual systems, or pieces of them, may be shared between users. Prospero is currently enjoying widespread use for remote access to the archie database.

### We Name It

Following a coffee break, the remainder of Thursday morning was spent discussing file naming and flaming about file attribute handling in UNIX. The panel consisted of Brent Welch from Xerox PARC, Rob Pike from AT&T Bell Laboratories, Jeff Mogul from Digital Equipment Corporation, and Cliff Neuman from the USC Information Sciences Institute.

Pike enthralled everyone with his humorous argument for local name spaces as implemented by Plan 9. He believes in referring to everything as it relates to himself: "me", "my office", "my computer", "my /bin", and "my /dev/tty". He doesn't seem to need to name "you", "your office", or "your /dev/tty".

Mogul argued that file systems need first class attribute handling. UNIX has "crummy attribute handling" with a very limited standard set (i.e., those stored in a file's inode). Other attributes end up being hidden in a file's name (\*.c, \*.txt, etc.) or in a file's data (magic numbers).

Neuman again blasted global name spaces. In his view, current names spaces are managed by either bureaucracy or anarchy. The solution, of course, is to be able to organize name spaces yourself.

### Abstractions

Following lunch, Thursday afternoon was divided into two more sessions: Abstractions, another Technical Session; and a discussion of Works in Progress. This was followed by a buffet reception in the Michigan League Garden.

Brent Welch, now with Xerox PARC but previously from the University of California, Berkeley, spoke first about *A Comparison of the Vnode and Sprite File System Architectures*. He argued for a clean separation of naming and I/O in file systems. Although this split is taken for granted in classical distributed systems literature, implementations derived from the UNIX family, such as Sun's vnode architecture, have mixed the two, decreasing potential functionality.

Noemi Paciorek, Center for High Performance Computing, Worcester Polytechnic Institute, spoke next about their work on *An Object Oriented, File System Independent, Distributed File Server*. Paciorek described how her work uses Mach IPC ports (objects) for file system communication, rather than the vfs/vnode interface. Unfortunately, it was unclear what problem she was trying to solve and why anyone would want to use this system.

### Works In Progress

Thursday afternoon was rounded out with a number of short talks on works in progress. After hearing a description of the FICUS Stackable Vnode System, work going on at UCLA, the audience was awed by Matt Hecht, IBM Federal Systems Division, who described the mind-boggling computing needs of the IRS.

The IRS scans in all tax return forms, storing the images in a hierarchical storage system, and throws away the paper (hopefully to be recycled!). The system supports manual annotation of images along with Intelligent Character Recognition (ICR).

There are 10 IRS service centers, each of which processes an average of 30 million tax returns. Returns average 16 images, each 50-100 KB after Group IV fax compression. Each image is stored in a separate file, yielding 480 million files per year per service center. Personal tax returns are kept for 7 years; business returns for 75 years.

Forms progress through a 28 step pipeline involving 2000 employees and 1000 workstations per center. For obvious reasons, an input spike occurs around April 15: 350,000 forms/day/center.

Accessing the first image in a form for the current year takes approximately 10 seconds. Access to additional images in the same form takes only 1/2 second. To access a form from the previous year takes 10 minutes; older forms require up to 16 hours.

The next speaker, Rob Pike, discussed Plan 9 from Bell Labs. Plan 9 contains a number of innovations. All objects look like file systems, including networks, devices, processes, the console, and the environment. Most UNIX system calls are changed into ASCII files: `/dev/time`, `/dev/user`, `/dev/pid`, etc. System management programs are simplified by the `/proc` file system. `ps` is implemented by `cat /proc/*status`. A process may be stopped by `echo stop > /proc/27/ctl`. The backup system is integrated with the file system. Once a day, the file system is dumped to a WORM drive that may thereafter be used for access to old data. Finally, Plan 9 allows the name space to be unique in every process.

Next, Brian Pawlowski, Sun Microsystems, described LADDIS, an NFS file server benchmark making its way through the SPEC committee. He discussed how LADDIS evolved from *nhfsstone*, the problems it fixed, and many of the problems that remain.

Jim O'Toole described the Semantic File System, a combination of a file system with a database. SFS utilizes transducers to examine data as it is stored in a file system, and to enter files in multiple directories based on the attributes of the data. In one sense it seems to be an organizational tool similar to Prospero.

Carl Staelin, Hewlett-Packard Laboratories, introduced *Coconut*, which extends 4.4BSD's port of LFS to tertiary storage such as robotic tape handlers. *Coconut*'s primary interest is in high-performance access to write mostly, large, sequential archival storage and multi-gigabyte scientific data. File system blocks are addressed uniformly over all disks and tapes. *Coconut* caches segments of tape on hard disks, and migrates new data from disk to tape with a background archive process.

Thursday ended with a short talk by Michael McClennen, University of Michigan, on *Multi-Structured Naming*. McClennen believes that most files are characterized by a five component name. Organization of files is a difficult problem, and

files are often lost. Instead, files should be identified as a sequence of attributes. His implementation also utilizes a user-level NFS server, a common theme at the workshop.

### High Performance

Friday morning was divided into two technical sessions, one on High Performance and a second on Caching.

The first talk was *DataMesh Research Project, Phase I*, by John Wilkes, Concurrent Systems Project, Hewlett-Packard Labs. The DataMesh hardware prototype utilizes an 8-node transputer system network. SCSI disks are attached to seven of the transputers, the eighth provides a SCSI connection to a host workstation. The DataMesh software prototype, called Jungle, employs a couple of interesting techniques to improve performance. The UFS disk scheduling algorithm sorts requests by cylinder because it was designed in a day when seek times outweighed rotational latency. With today's much faster disks, rotational latency is far more costly. DataMesh takes rotational position into account with a new 2-dimensional scheduling algorithm. An interesting insight the DataMesh group has had is "Why ever have spare empty blocks on your disk?" DataMesh reduces file read times by replicating frequently accessed data in otherwise free disk blocks. When a block is needed, the closest replica is selected. Replicates may be made when the disk is otherwise idle.

John H. Hartman, University of California, Berkeley, discussed *Zebra: A Striped Network File System*. Zebra tries to extend the work on RAID and LFS to network file systems in order to provide scalable performance, high availability, and balanced server load. This is achieved by striping files across multiple network file servers in order to achieve performance greater than a single server can deliver. Parity information is sent to an additional file server to allow file reconstruction in the event of a single failure. Zebra does not attempt to address network partitioning problems. Hartman also seemed to hedge when asked questions about network performance.

The next speaker, Sanjeev Setia, University of Maryland, presented his research into the *Optimal Write Batch Size in Log-Structured File Systems*. A goal of LFS is to achieve write performance approaching the transfer rate of a disk. This is done by batching write requests into large segments that can be written sequentially on the disk. Setia points out that for the purpose of minimizing read response time, this is precisely the wrong strategy. While most write requests are

asynchronous and non-blocking, read requests are more often synchronous blocking operations. If a read request is received immediately after a large write segment has begun, it will have to wait a potentially long time. Hence, a compromise segment size must be found that balances these two goals. Setia has developed an analytic model for read response time that allows the optimal write batch size to be computed given disk characteristics.

Jeff Mogul, DEC WRL, described *A Recovery Protocol for Spritley NFS*. In 1989, Mogul proposed adding very simple Sprite-like enhancements to NFS to improve performance and guarantee consistency. While it was interesting research, this proposal was not suitable for production use because it lacked a crash recovery mechanism. In this talk, Mogul described a simple mechanism to correct this problem. He proposes a scheme similar to that used in Sprite whereby servers rebuild their state by communicating with each of their clients. To do this, servers record which clients are active in a small amount of non-volatile storage such as NVRAM or even on disk. Mogul introduced the idea of "embargoed" clients, which require special handling. These are those clients with which the server could not communicate when it booted. He also described a problem and its solution with space allocation during close operations.

Carl D. Tait, Columbia University, entertained the audience with his description of *An Efficient, Variable-Consistency, Replicated File Service*. His research is focusing on the file service needs of mobile clients such as portable computers connected with wireless networks. A system was described that allows clients to specify when they require strong consistency and when weak consistency suffices. Servers replicate files using a lazy asynchronous update scheme. He also described the recovery procedures for failures including conflicting updates.

The "processor" File System in UNIX SVR4.2 was described by Ashok V. Nadkami, UNIX System Laboratories. This enhancement to SVR4 was developed to allow management of multiprocessor systems. Modeled after the /proc file system, the new /system/processor file system contains statistics files for each processor along with a control file to modify their state.

Sedat Akyurek, University of Maryland, discussed his work on *Placing Replicated Data to Reduce Seek Delays*. As previously mentioned in the DataMesh talk, seek times can be reduced by replicating frequently accessed data in multiple

locations on the disk. Akyurek described his simulations showing the potential benefit of this technique when applied to a standard SunOS 3.2 system.

*Issues in Massive Scale Distributed File Systems* was presented by Matt Blaze, Princeton University. While client caching has allowed distributed file systems with hundreds and perhaps thousands of nodes to be constructed, no existing system can support 100,000 clients. Previous research has shown that one attractive idea, hierarchical caching, will not work. Blaze is researching a solution involving dynamic hierarchies that shows promise. Rather than building static hierarchies of primary and intermediate file servers, clients arrange themselves into hierarchies dynamically, based on actual file usage. If only one client uses a file, it communicates directly with the file server. As more clients begin using the same file, they divide themselves into a tunable hierarchy.

*Faster AFS*, presented by Michael Stolarchuk, CITI, University of Michigan seemed like a commercial to attract people to the AFS User's Group meeting. Stolarchuk spent just a few moments showing that a performance problem exists with

the current stacked vnode implementation of AFS. When accessing a cached file in AFS one would normally expect to get performance equivalent to UFS. This seems not to be the case; performance is much worse. He promised to give the answers why during his AFS User's Group meeting talk.

*The Delta File System*, from the University of Illinois, Chicago, proposes some interesting ideas. Delta uses a "threaded" log to achieve good synchronous performance for applications such as NFS. A log of meta-data modifications is maintained as a linked list on the disk. Because at least 10% of a UFS disk is normally free, it is almost guaranteed that a free block can be found under one of the disk heads at any time without incurring a disk seek or rotation. Delta usually utilizes one of these empty blocks for its log.

Aside from the annoying timing of the workshop – it was held the same week as InterOp's Spring Conference – the workshop was quite enjoyable and educational. It provided a very good sampler of current file systems research. For those interested in obtaining additional information, workshop proceedings are available from the Association.

---

## Campus Reps Sought

---

The Association invites staffpersons at colleges and universities to become USENIX representatives on their campus. The benefits for the student community are an introduction to the premier practical technical organization in computing. The benefits for the Association are the involvement of future practitioners in our field. The benefits to you, as a campus rep, are the opportunity to contribute to USENIX and a receive package of benefits including many printed materials, fee-waived conference admittance, and the like.

Please contact this committee for further information by emailing to [students@usenix.org](mailto:students@usenix.org).

For the University Outreach Committee:

Dan Geer, *Geer Zolot Associates* (Chairman)  
Rob Kolstad, *BSDI*  
Evi Nemeth, *University of Colorado*  
Sonya Neufer, *Canstar*  
Peg Schafer, *BBN*  
Melinda Shore, *Cornell University*  
Pat Wilson, *Dartmouth College*

The following is a list of our current campus representatives:

Ted Hanss, *University of Michigan*, CITI  
Stephen Henderson, *Auburn University*  
William Hogue, *University of Wisconsin*  
Jeff Kellem, *Boston University*  
David Kotz, *Dartmouth College*  
Carol Miller, *North Carolina State University*  
Richard Ord, *UC San Diego*  
John Ousterhout, *UC Berkeley*  
Ben Pratt, *University of Utah* –  
    *Supercomputing Institute*  
Peter Roden, *MIT – Project Athena*  
Michael Stumm, *University of Toronto*

# Attendee Survey Results

Summer Technical Conference  
San Antonio, Texas  
June 8-12, 1992

Total number of attendees: 1059

Total number of surveys tabulated: 284

## LOCATION & COST

	Very Important	Important	Not Important
How important is the location of a technical event in your decision to attend?	42 15%	180 64%	60 21%
How important is the Cost of Travel/ Hotel in your decision to attend?	67 25%	152 57%	47 18%

## JOB TITLE - Are you:

Company Management	General	15	5%
	Financial	0	0%
	Administrative	2	1%
	Marketing	3	1%
Computer Systems Operations	Management	56	20%
	Staff	80	28%
Engineering/Manufacturing	Management	12	4%
	Staff	91	32%
	Consultant	9	3%
	Educator	6	2%
	Student	8	3%

## Are you a member of USENIX?

Yes	233	84%
No	45	16%

## Did you receive information about Summer 1992 Conference in the mail?

Yes	215	77%
No	66	23%

How many technical events will you attend in 1992? 615 100%

How many of these will be USENIX events? 385 63%

Average Events Per Respondent 2.17 events

## Has this number increased/decreased over 1991?

Increased	119	42%
Decreased	158	56%

## Did you attend:

Winter '92 (S.F.)	87	31%
Summer '91 (Nashville)	88	31%

## SITE & HOTEL

### Is San Antonio a pleasant & appropriate site for a USENIX Conference?

Yes	271	99%
No	4	1%

**How about the Marriott Rivercenter?**

<b>Yes</b>	<b>267</b>	<b>98%</b>
<b>No</b>	<b>6</b>	<b>2%</b>

**Did you stay in a hotel in San Antonio?**

No	26	9%
Rivercenter	168	59%
Riverwalk	52	18%
Menger	6	2%
Alamo Travelodge	5	2%
Downtowner	5	2%
Motel-6	5	2%
La Quinta	4	1%
Holiday Inn	4	1%
Crockett Best Western	4	1%
Courtyard	3	1%
Days Inn	1	0%
Hilton	1	0%
Total	284	100%

**Length of Stay**

Total/nights	922
Average Stay/nights	3.57

**Roomshare**

Yes	67	30%
No	158	70%

**EXHIBIT**

**Is it important that a product exhibit accompany a USENIX conference?**

	<b>Very Important</b>	<b>Important</b>	<b>Not Important</b>
	88	120	68
	32%	43%	25%

**How many UNIX relevant product exhibits will you attend in 1992?**

UniForum	20	22%
Interop	19	21%
Sun World Expo	16	18%
USENIX	14	15%
UNIX Software Symposium	8	9%
Xhibition	5	5%
Siggraph	5	5%
Open Systems	2	2%
EurOpen	1	1%
NeXt World	1	1%
Total	91	100%

**Electronic Mail - Attendee List**

	<b>Yes</b>	<b>%</b>	<b>No</b>	<b>%</b>
Do you have access to electronic mail?	274	96%	10	4%
Would you like electronic attendee lists?	207	78%	57	22%
Would name & e-mail address be sufficient?	158	64%	87	36%
Would electronic version replace printed version?	197	75%	66	25%

# Call for Tutorials

## Call for Tutorial Proposals

by Daniel V. Klein

<dvk@usenix.org>

In an effort to continue to provide the best possible tutorials to the technical community which it serves, the Association is soliciting proposals for future new tutorials at its conferences. The tutorial proposals may cover any subject, ranging from introductory to advanced materials.

The type of tutorial we are most interested in are "introductory or overview tutorials for advanced people." We tend to avoid overly introductory materials (i.e., a proposal on "Introduction to C Programming" would not be appropriate). Previous conferences have included tutorials on such diverse topics as UNIX Network Programming, X Toolkit Intrinsics, Topics in System Administration, Mach Virtual Memory Internals, Kernel Internals, and Software Contracts and Intellectual Property, among many others. Tutorial instructors are paid for their presentations, have their travel and reasonable expenses reimbursed, and receive a complimentary conference registration.

Tutorials usually run for a full day (6 hours of class time plus morning, lunch, and afternoon breaks), although we are currently experimenting with half day (3 hour) tutorials. A proposal should include a statement of what you want to teach, and a coherent outline to your tutorial (not simply a list of what you want to cover, but the order in which you want to cover it, with an estimate on the amount of time for each subject).

Because a tutorial lasts about 6 hours, we need to know that you can comfortably fill that time, but not seriously overfill it (i.e., that you won't suddenly discover at 4:30 that you have another 3 hours of slides left to present). If you have any supplementary materials to distribute (e.g., copies of papers, shell scripts, source code, illustrations, etc.), give an indication of the volume of supplementary material, and a rough count of the number of slides you will be presenting during class. Historically, a typical tutorial has between 75-200 slides, along with up to 200 pages of supplementary material. If possible, include a couple of sample slides (one with text, one with a graphic) with your proposal.

If you have a complete or draft course already done, having a copy of the current materials available would be most useful.

We also need to know if you will be presenting or distributing any source code. If so, is it copyrighted by someone other than you? If you do not hold the copyright, you must be able to demonstrate that you have permission to use this material (this may be dealt with by requiring course attendees to have a source license). Because the USENIX tutorials fall outside of the "fair use" clause of the U.S. copyright law, the same rules apply for supplementary papers or reports.

Finally, your proposal should also include a summary of your previous teaching or lecturing experience, as well as a couple of references (that is, one or two people who have seen you teach that we can contact). These may be your students, supervisors, or colleagues.

Remember, we are looking for a proposal, so nothing you submit will be cast in concrete. You may later decide to change some ordering of materials, or we may suggest some changes. You needn't worry about getting it perfect the first time around. What we are trying to do is get a very solid feel for what you are offering.

The tutorial schedule for all conferences is usually scheduled 4-6 months in advance of the conference – the earlier we receive a proposal, the more conferences it can be considered for. Please send your proposals to <dvk@usenix.org>, or by physical mail to:

Daniel Klein  
USENIX Tutorial Coordinator  
5606 Northumberland  
Pittsburgh, PA  
15217-1238

Be sure to include an electronic and physical address and a phone number. All proposals will be acknowledged upon receipt.

# **USENIX Committees**

The USENIX Board of Directors may from time to time create or terminate standing and ad hoc committees, and may determine the names of such committees and the qualification of its members. It may also, to the extent permitted by law, delegate the powers and duties of the Board to these committees. The Board may elect the members of such committees or may authorize the President and/or any other officer(s) to select members. Below is a listing of the current committees.

## **Nominating**

The chair of this committee is selected by the Board of Directors, no later than 9 months preceding the Annual Meeting in every even-numbered year. The chair then selects the members of this committee. This committee is charged with presenting names of candidates for each Officer and for the Directors to the members for election.

## **Publications**

Deals with long-term and financial issues concerning journal, book, proceedings and newsletter programs.

## **Prizes/Scholarships**

Formed in June '92 to make a proposal about instituting awards to celebrate special achievements in the community, to obtain donations for such awards, and to contemplate other avenues for other scholarships and prizes.

## **Tutorial Review**

Evaluates the performance of offered tutorials and instructors based on attendance figures and student evaluations, and provides feedback to instructors. Decides on the tutorials to be offered at upcoming conferences and workshops, based on submitted proposals, peer and student evaluations of instructors, perceptions of the needs of the community, and tutorial attendance history. The program chair of the most current technical conference is invited to participate and one current Board member also serve.

## **University Outreach**

Established in 1991 to connect the existing base of the Association with its future membership, and to make the benefits and opportunities of USENIX participation accessible to graduate and undergraduate students. Committee members:

1) recruit primary contacts at campuses, who in

turn serve as a distribution channel for USENIX materials, and 2) get feedback from contacts on other ways of enhancing the value and relevance of USENIX to the university community.

## **Invited Talks**

Provides a long range view and continuity of topics offered at the main conferences; encourages participation by recruiting speakers; raises audience interest and participation by trialing different presentation formats, including panels, tutorials, overviews, demonstrations, and formal talks; fosters an informal interactive setting in order to promote exchange of technical ideas and user experiences. The coordinators select and schedule talks for the upcoming conferences; work with staff and speakers on logistical issues and printed materials, as well as recruit new speakers.

## **Conference Site Selection**

Works with Conference Coordinator on site selection issues for the main conferences (with the goal that this will lessen the amount of time the entire Board spends on this subject).

## **Online Library/Index**

Handles technical issues involved in its creation and maintenance; responsible for making sure it is current; decides what publications will be included.

## **Executive**

A subset of the Board that may make Board-level decisions between the meetings (e.g., oversees expenditures of large sums such as computer hardware purchases). The Executive Director consults with this committee in cases of personnel layoff or termination, and hiring of senior staff. Members have signature authority (along with Conference Coordinator and Executive Director) on bank accounts. Usually the president and at least one member who is located near the Executive office serve.

## **Special Technical Groups & Local Technical Groups**

Formed in June '92 to investigate and develop additional proposals for Special Technical Groups and the creation of Local Technical Groups.

## **Special Technical Group Document**

Formed in June '92 to work on refining the STG proposal which was drafted in the Spring of '92 by the USENIX SIC committee, to present recommendations to the respective Boards for approval, and ensure that it is reviewed by the USENIX attorney.

# **SAGE Membership**

USENIX recently launched its first Special Technical Group. The Systems Administrators' Guild (SAGE) is devoted to the advancement of systems administration as a profession. SAGE will recruit talented individuals to the profession, develop guidelines for the education of members of the profession, establish standards of professional excellence and provide recognition for those who attain them, and promote work that advances the state of the art and propagates knowledge of good practice in the profession.

USENIX and SAGE will work jointly to publish technical information and sponsor conferences, workshops, tutorials and local groups in the systems administration field. An interim board has been appointed and elections will be held after this LISA Conference to choose a new board, which will take office in January 1993. If you wish to join SAGE, please return this form to the address below.

Yes, I would like to join the USENIX special technical group SAGE, the Systems Administrators' Guild, as follows:

- I am a current USENIX member. I wish to join SAGE. Enclosed is \$25 to cover dues for the remainder of 1992. My membership number is \_\_\_\_\_.
- I am not a current USENIX member. I wish to join USENIX and SAGE. Enclosed is \$90 (\$65 for a one year individual membership in USENIX; \$25 for SAGE dues for 1992).

\*\*\*\*\*

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_ Country \_\_\_\_\_

Phone \_\_\_\_\_ email address: \_\_\_\_\_

## **PAYMENT OPTIONS**

- Check enclosed payable to USENIX Association or SAGE.  
 Please charge my:  Visa  MasterCard  

Account # \_\_\_\_\_ Exp. Date \_\_\_\_\_

Signature \_\_\_\_\_

Outside the U.S.A.? 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

9/92

## **USENIX Mailing List**

- I do not want my address made available to other members.  
 I do not want my address made available for commercial mailings.

Please mail this form to: USENIX Association  
2560 Ninth Street, Suite 215  
Berkeley, CA 9471

# SAGE Working Groups

## SAGE Working Groups

Each issue will feature reports from the various working groups that SAGE has created to help define its efforts. This will include calls for participation that the working groups issue to complete their chartered tasks. If you need more information about these items, please contact the group chair listed at the end of each report.

## SAGE Publications

The SAGE publications working group is one of the most active of the current working groups. Its initial focus was on both technical publications and software distributions, but it has become rapidly apparent that the software side of the focus was large enough to form its own group. Thus, the Online group was formed.

That done, we immediately took on the task of creating a newsletter that would appear as a section within this newsletter. With the participation of Steve Simmons and Paul Evans, as well as the entire interim board, we assembled the first set of SAGE submissions in record time (that appeared in the last issue.) This second submission is looking good so far, and the venture appears to moving along well.

We have been tasked by the interim SAGE board with building a set of speaker documentation for SAGE representatives world wide. Another proposal we are discussing is the creation of a topic-focused journal, proposed by Elizabeth Zwicky. This journal would target a very specific problem in each issue, and attempt both to collect relevant material from past conferences and other sources as well as to generate new material from the system administrators well versed in the areas in question.

The publications working group has been an active part of SAGE's advancements to date. I hope that we can continue to make these positive contributions for some time to come. If you are interested in participating in this group, please drop me a line, or send email to [listserv@usenix.org](mailto:listserv@usenix.org) with a subscription request for sage-pubs.

Bryan McDonald, Chair  
SAGE Publications Working Group  
SAGE Publications Coordinator  
[bigmac@erg.sri.com](mailto:bigmac@erg.sri.com)

## SAGE Policies

The policies working group is soliciting examples of computer policies and procedures. We would like anything that documents what you do when, and why you do it. These can deal with operations, account creation/access, disk space, file system ownership, mailing list ownership, peripheral access, physical computer access, or anything else that effects the way you run your computer(s). We have the pointer to the archive list of policies and seek to expand the SAGE library from this beginning.

This can cover anything from a single computer to a computer center serving thousands of users. It doesn't have to be UNIX specific, in fact, we'd like it to cover as broad a range as possible.

We will be formulating several suggested policy statements, as well as some possible procedural documents. These can then be used by systems administrators as an aid in establishing their own P&P documents.

Send electronic (ASCII or PostScript preferred) versions of your documents to [<sage-policies-survey@usenix.org>](mailto:sage-policies-survey@usenix.org). If you only have paper versions, please contact me directly to discuss how we can best make it available to the working group.

If you would like to be involved with this policies working group, send email to [<sage-policies-request@usenix.org>](mailto:sage-policies-request@usenix.org) with a single line message saying "subscribe sage-policies". As I understand it, anyone can join the working groups at this time, though that may change in January.

Thank you for any assistance you can provide.

Lee Damon, Chair  
SAGE policies Working Group  
[nomad@watson.ibm.com](mailto:nomad@watson.ibm.com)

## SAGE Public Relations

As a fledgling organization that is looking for world-wide exposure, SAGE is seeking members or potential members interested in representing

the organization at the many conferences and other gatherings of systems administrators held around the world. The current interim board has so far been struggling to fill this need but the scope is beyond the reach of just a dozen people.

The publications working group is putting together a set of speaker or BOF documents to be used in pursuit of this goal, and the USENIX office is working on a new-member package. We would like to make both sets of information and documentation available to representatives worldwide.

If you or someone you know would be interested in participating in this program, please contact me or any member of the interim board, or the USENIX office.

Bryan McDonald  
*bigrmac@erg.sri.com*

#### SAGE Ethics

The SAGE Ethics Working Group is chartered with developing a code of professional ethics for system administrators. Towards that goal, we would like to peruse similar codes developed by other organizations. If you know of such a code, and can provide either the text of it, or a complete reference (well, a usable pointer, at least) to it, please send that information to <[sage-ethics@usenix.org](mailto:sage-ethics@usenix.org)>. Names of organizations that have developed codes of ethics are useful, too, but we'd much prefer real references.

Ed Gould, chair  
SAGE Ethics Working Group  
*ed@pa.dec.com*

---

## *The ABCs of UNIX*

---

By Duane Bailey and John Hagerman

A is for Awk, which runs like a snail, and  
B is for Biff, which reads all your mail.

C is for CC, as hackers recall, while  
D is for DD, the command that does all.

E is for Emacs, which rebinds your keys, and F is  
for Fsck, which rebuilds your trees.

G is for Grep, a clever detective, while  
H is for Halt, which may seem defective.

I is for Indent, which rarely amuses, and  
J is for Join, which nobody uses.

K is for Kill, which makes you the boss, while  
L is for Lex, which is missing from DOS.

M is for More, from which Less was begot, and  
N is for Nice, which it really is not.

O is for Od, which prints out things nice, while  
P is for Passwd, which reads in strings twice.

Q is for Quota, a Berkeley-type fable, and  
R is for Ranlib, for sorting ar [sic] table.

S is for Spell, which attempts to belittle, while  
T is for True, which does very little.

U is for Uniq, which is used after Sort, and  
V is for Vi, which is hard to abort.

W is for Whoami, which tells you your name  
while

X is, well, X, of dubious fame.

Y is for Yes, which makes an impression, and Z is  
for Zcat, which handles compression.

## Whither The Customer? - or for Whom Do We Administer Systems?

by Kevin Smallwood

<kcs@staff.cc.purdue.edu>

*[This is the first of what we hope will be a regular series of articles on issues of a less technical nature, but, nonetheless, very important to system administrators. Free discussion is encouraged and solicited, if you are interested in participating, please contact me. — Bryan, SAGE Ed.]*

What's in a name? Was Shakespeare correct when he wrote, "A rose by any other name still smells as sweet"? If a new variety of rose was developed and called "Putrid Stench," would you be very eager to give it a sniff and do you think you would appreciate the smell? What do you think of when you hear the name skunk cabbage? Yes, the crushed leaves do have a skunklike odor, but when thoroughly dried, the leaves are a tasty addition to soups and stews. I suggest that a name IS important; in many cases it provides a mindset and a frame of reference. Yes, it often allows us human beings to pre-judge and anticipate, but we are only human.

I love to visit Walt Disney World and Disneyland, and I am not alone. One of the features about Walt Disney World that totally fascinates me is how clean the facility is. Once I watched as a person loaded film into their camera and then dropped the empty film box on the ground. Within thirty seconds a Disney "cast member" swept up that litter. You blink your eyes and the park is clean. Why is that so difficult for other theme parks?

How is it that Walt Disney Productions can keep a park so clean? I found the answer to this question in "A Passion For Excellence," coauthored by Tom Peters and Nancy Austin. "At Disneyland and Disney World, every person who comes onto the property (the "set") is called a guest. Moreover, should you ever write the word at Disney, heaven help you if you don't capitalize the G."

Is this tripe? I suggest that it is not. Let's look at a contrasting example. Tom Peters describes, "On an all-night flight to Denver our plane stops briefly in Salt Lake City. It is on the ground for only about nine minutes, and then the Salt Lake passengers begin to board. As the new people

begin to come down the ramp, the head stewardess turns to her associate and says, 'Here come the animals.'" Think about this: there are a lot of things we do to "animals" that we would not consider appropriate for "Guests." We all know how bad airline service can be.

In his book, "It's Not My Department!", Peter Glen tells the following story:

I had landed late at a hub city and missed my connecting flight. There was another one in four hours, so I decided to try my luck on a commuter airline that had a flight leaving for my destination in less than an hour.

All of the check-in counters for this airline were located in one area, so I waited in line in front of a sign that listed the city of my destination. After about 10 minutes I was told that this particular flight was an earlier one that had already left and that the passengers for the flight I wanted were being checked in at another counter. After another 10 to 15 minutes at that counter, the agent told me my flight check-in had been moved to another counter. I went to the new counter.

I handed my ticket to the agent, and she said, "Sir, you should check in at the next counter." Now since the next counter was part of her counter and she could actually reach over and touch the computer, I said, "Look lady, I am a Customer! This is the third line I've waited in and I'm not waiting in another one. I am a Customer! You take two steps to the left and check me in."

She was a little startled, but she did what I requested. When she handed me my boarding pass, she said, "Sir, we try not to use the 'C' word around here."

They don't try to use the 'C' word because they don't try to treat their Customers as Customers, and it shows.

I would imagine that many of you have similar horror stories. I doubt that many of you would be happy in the above situation; you are a Customer, and you want to be treated as one — your money is being used to employ this person. Yet, how many systems administrators make their "users" perform equally demeaning tasks and jumping through hoops?

What do you require from a "user" to restore a file? A complete dossier neatly typed (on the department's only manual typewriter) in triplicate, of course, hoping that the sheer complexity of the request will dissuade this "user" from ever requesting a file restoration again?

How do you handle notification of system downtime? Do you often say to yourself, "Oh, heck, those stupid users don't know how to use this computer, so they won't even miss it for this fifteen minute reboot. Let 'er rip!"

And, of course, we all know what is best for our "users" in all cases, don't we? We all know that a researcher in biology doing X-Ray Crystallography needs the latest version of "bison" instead of an optimizing FORTRAN compiler. I mean, why would anyone be so stupid as to program in FORTRAN? How many of you have said that once or twice?

So, again, I suggest that a name is important. For this reason, I implore you to refer to the people for whom you administer systems not as "users," but as "Customers", "Clients", or "Patrons".

Why? What image do you conjure up when you think of a "user"? Maybe my high school drug education class left a lasting impression on me, but I have a very vivid image of a junkie huddled in a corner with a needle sticking out of his arm.

There are some sites proud of the fact that they call their Customers "lusers"; they even print it in their newsletters. And, who hasn't heard of "Stupid User Tricks"? With all of this imagery and mindset, it is difficult to display much common (or should I say uncommon) courtesy toward those who justify our jobs and often indirectly pay our salaries. Tom Peters refers to this as "thinly disguised contempt for the Customer." In many cases, I don't think it is all that thinly disguised, either. It is easy to deny maintenance on a FORTRAN compiler for a "user" (or "luser"). Yet, how would we act if we all got paid on commission? Satisfying that Customer would mean a whole different thing, wouldn't it?

How secure are you and your systems administration staff in your positions? Treating the people you administer systems for as valued Customers isn't really necessary, or is it? I know of a couple organizations where the systems administration staff was so smug to think that their "users" couldn't live without their talents, that they treated the "users" as antagonists rather than Customers or even equal partners. This was a clear battle of ego; there were well educated, talented people on both sides of the issue. The "users" of those organizations won in the long run. With enough properly placed complaints about work-stopping road-blocks put in the way of the "users," management was forced to look

for alternatives. In both cases, management brought in an outside organization that realized that they had to provide both good technical solutions and perceived quality service to the Customers. Initially, it cost the organizations a little more, but in the long-run, the Customers were pleased with the level of service they received, finished the projects, and the organizations were flourishing in the businesses they were in.

Don't think that can happen to you? I know that the two systems administration organizations felt the same way right up to the day the pink slips were handed out. As more and more independent systems administration organizations that know the value of the "Customer" come into being, the higher the chances of the same thing happening to you if you continue to show "thinly disguised contempt" for your Customers.

Now, I don't mean to imply that simply calling someone your "Customer" is a quick fix. If you don't believe that as a systems administrator you provide a technical service and that those people you administer systems for are your Customers, then you will still exhibit that "thinly disguised contempt for the Customer." However, getting into the correct mindset is a good first step. Furthermore, if you expect to provide service excellence, you must treat your employees the way you want them to treat the Customers. If you don't have employees under you, share this article with your boss. Expert after expert agree that the Customer will be treated no better than the employee is treated.

Would you try a little experiment for me? For just one day, while you are at work, tell yourself over and over "Customer, Customer, Customer." Don't let the word "user" enter your mind. Use "Customer" in your writing, speaking and thinking. At the end of the day, look back and see if you treated people differently. Then, ask yourself how would you want to be treated: as a Customer or an animal?

In future articles, I hope to expand on some of the issues I only touched upon in this article. Many of you know the technical knowledge to be competent systems administrators, but lack an equally important skill: providing quality Customer service. Also in future articles, I hope to respond to your comments, criticism, suggestions, and questions. So, please write.

## Counterpoint

by Rob Kolstad

<kolstad@bsdi.com>

Kevin Smallwood points up a fine method of improving the image of system administrators in his article *Whither the Customer?*. I get the idea while reading the article, though, that there's a bit of a power-play or adversary relationship that we, as system administrators, are fighting.

I think there's no denying that a power-based relationship ("I can put your job to the bottom of the printer queue" or "I'm going to tell your supervisor that you didn't get my workstation installed") is probably the least productive kind of interaction that a system administrator can have.

I fear, though, that the notion of user-as-customer, with its subtle implication that "the customer is (always) right" and the (too often one-way) deference to the customer by the "clerk" (a.k.a. salesperson, administrator) pushes the pendulum too far in the other direction.

I have spent the majority of my professional life in industrial computer centers (as opposed to academic ones). I believe that one particularly good tone for user-administrator relationships in the industrial setting revolves more around teamwork than around one co-worker serving the other. When people believe they are part of a team, particularly a team with a common goal, it is amazing how smoothly things can proceed.

Bob Paluck, President and CEO of Convex Computer Corporation, is a master at focusing engineering teams on a common goal. He built a 30 person organization that built a mini-supercomputer from scratch in 18 months. The project included then-revolutionary 20,000-gate gate-arrays, circuits and cabinets, and software (including both an operating system and vectorizing compilers). His key strength was the sharpening of the group's focus. The group responded by working as a team – with members using their strengths to help other members who needed the help. To me, this same kind of teamwork exemplifies the zenith in user-administrator relations.

It is important, though, to avoid going too far in this 'help each other' motif. In the extreme example, team members line up outside some particularly competent person's office, each waiting their turn for the talented person to solve the problem assigned to them as their primary work-task. This situation signifies a talent mis-match inside the working group that requires remedy.

In summary, I think that peer-peer working relationships (typified by the teamwork approach) may have even more benefits than relationships which might appear to be based upon power or presumed superiority.

**Managing NFS and NIS by Hal Stern. O'Reilly & Associates, ISBN 0-937175-75-7, 410pp. Soft-cover.**

Reviewed by Steve Simmons  
*<scs@lokkur.dexter.mi.us>*

Four or five years ago you could count the number of UNIX system administration books on one hand and have a majority of fingers left over. The past two years have seen an explosion of books on the topic. In general this is good, but there are some downsides.

Almost every one of the recently published books attempts to cover the topic very broadly. While this is a benefit for the new administrator, it often means that subtle system-to-system differences are not well documented. It also means that certain specialized subsystems are not covered with the depth or detail that one might like.

This situation is similar to that of the general UNIX market ten years ago – a relatively small number of very broad books, and lots of grubbing through the manuals (or source) for specific answers. O'Reilly Associates noticed this gap and filled it with their Nutshell Handbooks. I'm pleased to note that they're still doing the job with a new series of in-depth single-topic books intended for the system administrator.

*Managing NFS and NIS* by Hal Stern opens with its weakest chapter, a very quick overview of networking fundamentals. The treatment is so cursory that it is of little help to the new or network-naïve administrator, and the experienced administrator does not need it at all. Fortunately this is the book's low point.

Following the overview of networking Stern covers NIS in detail. Among the topics presented are such useful items as a table of standard NIS maps,

defining and using netgroups, a flow diagram of how data is accessed from the maps, the relationships between the various daemons and servers, how to make and use custom NIS maps, and a number of other useful topics. Some of this information is available by diligent reading of the manual pages, but Stern pulls the various pieces together into a reasonably coherent description of the system as a whole.

While the sections on NIS are useful, for most uses NIS works fine "out of the box." NFS is far more problematic, especially as networks grow and change. Stern spends the most of the book on NFS, and it is here that the network administrator will find the most help (this is doubly true for the Sun administrator).

Stern covers details of workstation booting which are not covered in any standard Sun manual I have found, including a complete description of the boot process for diskless workstations. Other useful topics include NFS mount types and their pluses and minuses, distributed mail spools, diagnostic and administrative tools, PC-NFS, and a host of others. The material is covered well enough that when I have a problem, I grab this book before bothering with the Sun manual. More often than not, the Sun manual is not needed at all.

The book is not without its weaknesses. The treatments of security and performance tuning are reasonable but necessarily somewhat sketchy. While these are reasonable topics, both deserve a much wider treatment. (Note that O'Reilly has separate volumes which address them.) They could be dropped from the book and replaced by pointers to books from O'Reilly and others with little loss.

# SAGE Report on World Conference

## Report on World Conference on System Administration and Security

by Bryan McDonald  
*<bigmac@erg.sri.com>*

Washington, D.C. was the site of the World Conference on System Administration and Security (WCSAS), sponsored by the Open Systems Conference Board. Alan Paller, chairman of the four day conference (July 20-23, 1992), described this conference as a source of practical solutions to today's system administration and security problems.

Jon Gossels, Area Manager for DCE and DME at the Open Software Foundation (OSF), gave the keynote: *The Future of Systems Management and Security in Distributed Computing*. He presented an overview of the OSF DME architecture that should be available to OSF members for further development next year. A lively panel discussion followed the keynote and featured Gossels, Frank Moss of Tivoli, Dean Thompson of Hewlett-Packard, and Rob Kolstad of BSDI.

The topic of the keynote and panel, the OSF Distributed Management Environment (DME) and its industry-wide acceptance, while presented with enthusiasm, was greeted with skepticism by many of those present, as evidenced by the questions posed during the panel session. One attendee commented afterwards that he felt that the keynote was just a sales pitch, but that the panel discussion presented a more balanced picture.

Multiple audience participation sessions were well received by organizers and attendees, providing useful contacts for many system administrators with a problem. The technical tracks were organized well, if at the last minute, and covered a very wide range of topics, from security tips to perl examples to a discussion of using your university dining hall system to provide network keys.

Birds of a Feather sessions were small, organized by the vendors, and lightly attended, but the number of attendees leaving at 4:30 to beat the commute traffic influenced attendance figures for all events that stretched into the evenings.

As the conference closed, the chairman informed me that he was satisfied with the event as a whole, and that he plans to hold another one in the coming spring. Official reports placed total registrations at 250 or better, but reports from attendees place the average daily attendance at 120 to 150 people, with a very large percentage of these attendees being local. The Hilton Hotel, while potentially a nice setting, was less than ideal as many areas were under construction, but the surrounding neighborhood and easy access to the likes of the Smithsonian proved an enticing lure to the adventurous.

# What's Out There?

Volume 1, Number 2

by Jeff Kellem

Beyond Dreams

<composer@Beyond.Dreams.ORG>

## Overview

Well, hi again! I'm back, with an issue of this newsletter already in between issues of this column. Apologies if I haven't returned your mail regarding this column. Online things are still moving to my new location. But, your messages should be here; I just need to sift through all my mail.

Speaking of online things, let's talk about online texts. There's a bunch of things going on to make texts -- whether they be books, papers, articles (such as these), or the transcripts of presidential candidates' speeches -- available online. I'll mention a couple of the projects attempting to make online texts available, leaning towards the online libraries in the next century.

There's also mention of the USENIX bibliography, the Cygnus Solaris-2 Development kit, the Davenport Group, and an archive devoted to Handicap related materials.

## Project Gutenberg

Project Gutenberg was started by Michael Hart about 20 years ago, in early 1971, at the University of Illinois. Its primary purpose is "to encourage the creation and distribution of English language electronic texts." Project Gutenberg hopes to help create the electronic library of the 21st century, with a goal to bring 10,000 books online by the end of the year 2001. Part of this hope is to make it easy to have an accessible online library at your fingertips (at home).

They are currently making texts available in pure ASCII format in an attempt to make them as widely available and accessible as possible. The idea is that you don't really need any special hardware and software (beyond a basic computer :-) to read, index, or search these electronic texts (or 'Etexts' as Project Gutenberg refers to them).

Michael Hart started out Project Gutenberg by typing in the United States' Declaration of Independence, followed by the United States' Constitution. Their latest releases include *Aesop's Fables*, *Paradise Lost*, *Alice in Wonderland*, *The Scarlet Letter*, *Data from the 1990 U.S. Census*, *Roget's Thesau-*

*rus*, *The War of the Worlds*, and *Zen & the Art of the Internet*.

Every etext Project Gutenberg releases is expected to be in the public domain, whether originally placed there or after expiration of its copyright. Another aspect of Project Gutenberg is the hopeful formation of "The Public Domain Register". This would be similar to "The U.S. Copyright Register", but as a central repository that could be searched for works that were in the public domain.

Too often it is hard to tell whether a work is in the public domain, since someone (a publisher, for instance) can come along and put the work into their own edition and copyright that edition. My understanding is that the actual work is still in the public domain, though that particular edition of the work is not. It would be extremely useful to be able to go to "The Public Domain Register" (after its formation) to find the copy that was put into the public domain. Project Gutenberg hopes to get "The Public Domain Register" going in 1997 -- which happens to be the 100th anniversary of "The U.S. Copyright Register".

Another goal of Project Gutenberg is to make available a simple, childlike guide to the Internet titled "A Child's Garden of the Internet". Each section is geared towards the absolute novice and is expected to be basically a "Ten Minute Tutorial" (which should be able to stand on its own) on some subject related to the Internet. This will be published in hardcopy, with the etext also available.

To take a look at the current etexts available from Project Gutenberg, anonymous ftp to:

mrcnext.cso.uiuc.edu

and look in the /etext directory. The etext92 subdirectory contains the current 1992 releases, while the 'articles' subdirectory has articles and newsletters from Project Gutenberg.

The Project Gutenberg repository is also searchable via WAIS using the 'proj-gutenberg.sr'. Search the directory-of-servers.sr for "gutenberg" to obtain the 'proj-gutenberg.sr'.

For more information on Project Gutenberg, contact via postal mail:

David Turner, Project Gutenberg

Illinois Benedictine College

5700 College Road

Lisle, IL 60532-0900

via e-mail: <Michael Hart <hart@vmid.cso.uiuc.edu>>

You can also subscribe to the online Project Gutenberg newsletter/mailing list. If you receive

the bit.all USENET newsgroup hierarchy, then you can just read *bit.listserv.gutnberg*

To subscribe to the newsletter/mailing list via e-mail, send a message containing no subject with a body consisting of:

```
sub gutnberg Your Real Name  
to listserv@vmd.cso.uiuc.edu
```

### The Online Book Initiative

The Online Book Initiative (OBI) was started back in 1990 by Barry Shein and a small group of people. Its goals are similar to Project Gutenberg's. The basic idea (to make texts freely available and redistributable online) is the same, though not with the stated [Project Gutenberg] goal of working towards the library of the 21st century.

OBI plans to make online texts of all sorts freely available and redistributable. These include both public domain works and works with copyrights that allow redistribution. The current OBI archives consist of over 150MB of material (some of which is compressed), ranging from RFCs to Moby Dick to various poetry to technical papers to transcripts of U.S. Presidential Candidate Bill Clinton's speeches.<sup>1</sup>

Access the OBI archives via anonymous ftp from:

*obi.std.com* in the /obi directory.

Two OBI mailing lists exist, an announcements list and a discussion list. To subscribe to either list, send a note to:

*obi-request@world.std.com*

For further information on OBI, send a note to:

*obi@world.std.com*<sup>2</sup>

### What You Can Do

Both Project Gutenberg and the Online Book Initiative could use extra assistance, whether by donating materials to be included in their repositories, helping with scanning, proofreading, and/or indexing documents, or donating funds to help further their causes. Be sure to contact each group before donating, so as not to duplicate

1. If anyone knows if the transcripts of Bush's speeches are available, I'm sure the OBI would be interested in them.

2. The World (*world.std.com*), a public access unix system, acts as the home of the OBI. For more information on The World, contact: The World, Software Tool & Die, 1330 Beacon Street, Suite 215, Brookline, MA 02146, 617-739-0202 (voice), 617-739-WRLD [-9753] (data), *info@world.std.com*

efforts and to verify any correct procedures.

### Other Things Available Online

Of course, there are many other projects, public and private, around the globe that are working towards getting texts online – too many to mention here, unfortunately. They tend to be attached to various universities, but not always. Some time ago, I had archived a list of various text archive projects; if I can dig it off tape, I'll try to make it available on *ftp.dreams.org*.

### USENIX Bibliography Online

The USENIX Association over the years has attempted to make indices of all their publications available online. [*See p. 3 in this newsletter for more information. – Ed.*] For one reason or another, this has not always been a steady thing. Recently, the project regained momentum and will be kept up-to-date in the future. [The index is currently up-to-date.]

There has also been discussion by some folk to try to make USENIX papers available online (with the author's permission, of course). The current index of USENIX related material can be found via anonymous ftp from *ftp.uu.net* in the: /published/usenix/bibliography directory.

### This Column and "beyond"

The text of this column is also available online via anonymous ftp from: *ftp.dreams.org* in the /pub/whats-out-there directory. Organization of the anonymous ftp area for Beyond Dreams is just starting, so locations of things may change. But, I'll try to keep the above pointing to the appropriate area.

Other Beyond Dreams related material will also be made available there, including the slides from the Invited Talk I gave at the Summer 1992 USENIX Conference in San Antonio titled *Self-Help on the Net*. That's in the /pub/self-help directory on *ftp.dreams.org*.

### A Standard to Come?

The Davenport Group, a "group of open-systems vendors, software vendors, and book publishers" which started meeting in May 1991, is attempting to "serve as a catalyst for emerging standards and to promote an open-systems approach to on-line documentation and electronic publication."<sup>3</sup>

The Group was founded by Dale Dougherty of O'Reilly & Associates. They are currently work-

3. Quoted from the file 'davenport.intro' available via anonymous ftp from *ftp.ora.com* in the '/pub/davenport' directory

ing on establishing a common interchange format for online documents, basing it upon SGML and HyTime.<sup>4</sup>

There are various working groups within the Davenport Group, some of which include:

- The SGML Query Language working group, which plans to look into standardizing a SGML-oriented query language for searching documents that conform to whatever format the Davenport Group comes up with, and
- The Committee for Common Man (CFCM) working group, which is developing a standard form of online documentation along the lines of man pages.

For futher information regarding the Davenport group, `ftp://ftp.ora.com` and look in the `/pub/davenport` directory. The file `davenport.intro` will be useful to read.

#### Miscellaneous Unrelated Tidbits

Recently, an anonymous ftp archive dedicated to making available programs and information related to the handicapped was set up at `handicap.shel.isc-br.com`. Their archives duplicate those of the Handicap News BBS.<sup>5</sup>

There are plans to include all 2800+ issues of the Handicap Digest, in the future. Here's a sample of things included in the archive:

National Federation of the Blind  
Muscular Dystrophy  
Multiple Sclerosis  
Chronic Fatigue Syndrome  
Related Educational Software  
Programs for UNIX and the handicapped  
Americans with Disabilities Act (ADA)  
Calendars of Conferences and other Events

The archive maintainer is Bill McGarry  
`<wtm@bunker.shel.isc-br.com>`.

For those of you with Sun workstations and moving to Solaris, Cygnus Support has completed release 1.0 of their Solaris-2 Development Kit. This was the outcome of a project they started called "GNU C for Solaris". It's available via anonymous ftp from `ftp.uu.net` in the `/vendor/cygnus` directory. Cygnus Support sells professional support for various freely available

4. SGML stands for Standard Generalized Markup Language (ISO/IEC 8879). HyTime is the Hypermedia/Time-based Structure Language standard (ISO/IEC DIS 10744).

5. Handicap News BBS can be reached via +1 203 337 1607. It supports speeds of 300/1200 /2400bps, 24 hours a day.

packages, such as the GNU Project's suite of programs.<sup>6</sup>

#### Corrections

In the previous issue of *What's Out There?* (Vol. 1, Issue 1) in the May/June 1992 issue of this newsletter, I mentioned the Geographic Name Server. It's maintained by Tom Libert `<libert@eecs.umich.edu>`, NOT by Merit, Inc. Apologies for the misinformation.

I also mentioned Charlie. At the time of writing the article, Charlie was still available. Unfortunately, for various reasons, Charlie had to be taken out of service. Luckily, various people have the data that was in Charlie and work is being done to make use of it.

At the Summer 1992 USENIX Conference, a group of us got together at the Archivists Birds Of a Feather session to discuss keeping track of packages on the Net. From this discussion (and ideas from the past several years), there was general agreement that a "package registry" of sorts should be set up.<sup>7</sup> This registry would contain standard forms for various packages on the Net. Plans are afoot to standardize the registry form, set up this registry, and develop mirroring software that uses this information. The idea is to have Certified FTP Mirror sites (CFMs) and UUCP Servers (CUSs) which would guarantee to have registered, accurate, and up-to-date packages. This would be automated and checked by standardized software.

For more information (or to join a mailing list discussing this project), send e-mail to:

`registry-request@cfcl1.com`

Also, if you're interested in similar archivist topics, you should read and participate in the USENET newsgroup `comp.archives.admin`.

#### What's Next?

In future issues, I expect to include such topics as gopher, astronomy related archives, directory services, along with various other information related services.

Till next time, I'll see you on the net.

*Jeff Kellem is the founder of Beyond Dreams, an organization which promotes information exchange and communication. He can be reached via `composer@Beyond.Dreams.ORG`*

6. For further information on the kit or Cygnus Support: Cygnus Support, 814 University Avenue, Palo Alto, CA 94301, USA, +1 415 322 3811

7. Charlie was a type of package registry.

# Millionaires

by Peter Collinson  
[pc@hillside.co.uk](mailto:pc@hillside.co.uk)

It's a new experience for me being a vendor. A vendor in a computer start-up too. I sit in the back room of my house selling software. I describe my activities to my uninitiated friends as being a "trainee millionaire." Hope springs eternal, I guess. We all know what millionaires do – don't we? I thought I did.

Trainee millionaires don't need brains – they just need to know how to put things into envelopes. You must be an expert envelope stuffer. I cheat a little – my envelope contents all come from my computer, individually and lovingly printed for each recipient.

My computer printer puts addresses into the little boxes so that they show through the window in the envelope. This saves on the phase of addressing the envelopes. I can master the skill of putting things into envelopes, addressing them is beyond me.

To send a letter, I get the several bits of paper from the laser printer and flip through to get to the personal point where I sign it. Trainee millionaires must use a swanky fountain pen – so I have to wave the sheets of paper in the air to get the ink to dry. Only fountain pen users have noticed that modern paper is no longer designed to work with ink. The ink doesn't soak into the paper like it used to. If you're not careful to let it dry, it runs off the paper and down the page. I digress. I need to digress a bit to let the ink dry.

Once the ink has become squiggles on the paper rather than some running stream, I can locate the tiny fold marks that I have carefully printed at the side of the page. I make two folds in the paper, ending up with a concertina that slips into the envelope and instantly springs apart to make it look satisfactorily full.

When I am done I have a pile of envelopes that make a low sighing noise when you pick them up. The concertinas contract under hand pressure and the air slithers out through the tiny holes at the sides of the flap. I am surprised that no-one's thought of talking envelopes yet, special speaking attachments that can be powered by the air whooshing out as the envelope is gripped. Product announcements by post.

At this point, it's a good idea to look in the little window at the front to make sure you can see that address that was so lovingly printed. It takes real skill to ensure that the paper got into the envelope the right way round.

There are now two more things to do, and mortals use their tongues to do them. One is stick down the flap of the envelope (dirty minds were probably thinking about something else). The other is placing the stamp on the envelope in the correct place.

Trainee millionaires can't use their tongue because it's needed to do other things, so they will use something else that is wet. Being a start-up, it is appropriate to use the sponge that is normally used for cleaning the car – you can find that easily, it's lying around the garage.

Don't just put the sponge under the tap and place it on the table. You'll end up with a mess. Make sure that you put the sponge into a dish. You'll find that this is best done when the family is not around. The family associate the dish with nice meals and know that the car sponge has been with the by-products of nice meals that birds just ate.

You'll want to have a nice meal after all this intense envelope work so make sure the family is not around, wet the sponge and put it on that dish. The wet sponge is a dual purpose tool. You can use it to wet the envelopes AND the stamps too.

Using a sponge with stamps involves a lot of skill and determination. Mortals will take one stamp, lick it (ahem, place it on the sponge) and stick it onto the envelope. Trainee millionaires don't do this. They take a whole strip of stamps, and wet them all. You then have to stick them onto the envelopes as fast as you can before the glue stops working.

There is also a special art in making sure that the stamp is placed on the envelope intact. Or you can play 'fool the Post Office' by trying to leave a little bit of each stamp on the previous envelope. It takes real skill and enormous practice to do this consistently all the way through a strip of stamps.

Finally, your task is done. The envelopes are stuffed. The stamps are mounted. The envelopes sealed. You look at the fruits of your labours while sipping delicately on that bottle of fizzy water (champagne comes later, you're only a trainee millionaire – right?). Anyway, the sponge has ensured that the liquid doesn't taste of glue. Remind me to clear the sponge away before the family return.

In the end, all this is wasted training and that's sad. It makes enough money so the trainee millionaire can hire other people to put things into envelopes. The fully trained envelope stuffer

sneaks off for expensive lunches, trips to the golf course, learning to fly aeroplanes etc., etc. I can hope, I suppose.

## In Favor of Student Grants

By George V. Neville-Neil  
*<ggn@pa.dec.com>*

The funding of student grants to attend USENIX conferences and workshops is an important service that the Association provides to the community and students. As a recipient and now active member, I would like to acknowledge the importance of this program.

I first got involved with USENIX by reading an article in *Computing Systems*. Although I had been involved with Unix at school, I had never heard of USENIX. After reading that issue of *Computing Systems* I joined as a student member. While discussing USENIX with others, I found out that one of advantages of being a member was going to the semi-annual conferences. But as a student I figured it would be unaffordable to go.

A few months later, I read about the student grants program, and applied for one. With that grant I attended the Winter 1990 Conference in Washington DC. I was able to meet a lot of new people and, to use that horribly '80s phrase, network. The connections I made at the conference helped me to get a job after graduation, and have enabled me to keep up with technical trends in the industry.

"So what?", you might ask. At the Winter 1992 conference you may have seen my name listed as a volunteer. This was my direct response to USENIX helping me out. I felt I owed a debt to USENIX, and should try to help the Association as much as possible. Turn about is, after all, fair play. Did USENIX buy my loyalty? Not really, it's just that I'm more than willing to help an organization that was willing to support someone they'd never even met.

You can all put your handkerchiefs away now, and I'll get down to the real reasons to have a student grants program. At the Winter '92 conference's "Meet the Candidates Night," a lot of people talked about the need to increase membership by reaching out to people. Students are excellent candidates for this type of recruitment. These are the people who, in a few years, will be "the industry". Students contribute a lot to the Association. According to the Executive Director, "...for the last few conferences, we've had an average of 3 student papers that have been accepted .... and at two recent conferences students won the best overall paper, as well as best student paper!" The contribution of new ideas by students keeps people interested in USENIX. If we can get more students to attend our conferences, it is likely that more of them will contribute their work as well.

The final point to ponder is just how much we are spending on this program. According to the Association, \$10,000 is awarded for stipends for the two technical conferences, and \$15,000 is allocated to all workshops and symposia. Most students that register for a conference also become members.

In conclusion, the student grant program gets us new members, encourages submissions from student authors, is good public relations, and doesn't cost that much. With a return on investment like this, I believe it is a wise expenditure of Association resources that cultivates its future membership.

# Autoloading in Perl

by Tom Christiansen  
*<tchrist@convex.com>*

*Editor's Note: Advanced Perl; only trained experts should try these stunts.*

## Abstract

Interpreted systems like Lisp have for a long time used a technique known as autoloading to speed up execution. Until now, however, autoloading has not been available for Perl programmers. This article presents two techniques for autoloading in Perl.

## Introduction

Interpreted programs must be parsed by their interpreter before they can execute. In large programs, particularly large interactive programs, this can produce a noticeable and potentially irritating delay in start-up time. As an interpreted language, Perl also suffers from this problem. Even if a program is several thousands of lines long, users will always expect a prompt as soon as they hit the carriage return to their shell.

One way to minimize start-up time is to parse the program in advance and save the parse tree on disk. In Perl terms, this amounts to using the `dump` built-in to produce a new executable. Unfortunately, because the entire Perl binary will be included in this image, the file will range in size from large to enormous.

Autoloading is a mechanism to delay the interpreting of a function until it is really needed. This can exhibit a real reduction in start-up time for large programs, especially ones that do not make use of all their functions on each run. Unlike programs compiled into machine language, where a function that is never called may never be faulted in, an interpreted program is not usually smart enough to avoid loading everything. In a sense, autoloading is to an interpreted system what a demand-page VM system is to a compiled one. You can also think of it as delayed compilation.

Autoloading can also facilitate debugging large programs by reloading functions that have been modified – while the mainline program continues to run.

The basic idea behind autoloading is the creation of a stub function for each real function whose compilation is to be delayed. When called, the stub function replaces its own definition with that of the real function, and then transparently calls the newly loaded version. For each function to autoload, a stub is generated this way, using an `&autoload` function (defined later):

```
&autoload("myfunc")
```

From then on, if the `&myfunc` function is ever called, the stub will load the real `&myfunc` definition from disk, overwriting the previous stub definition of `&myfunc`, and run the new one. But if it is never called, then no compilation costs are incurred beyond that of calling `&autoload`, which is considerably less than loading all but the smallest functions.

Consider this code:

```
if (-p $file || &myfunc("$file.bak")) {  
    ...  
}
```

The `&myfunc` function is loaded only if the second part of the `||` conditional is taken because `"-p $file"` is false. If `&myfunc` had 1200 lines to it (or even 120), these savings add up quickly.

The principal hurdle to surmount before we can write an autoload function in Perl is the restriction that a function may not redefine itself, although it is perfectly free to redefine another function. Because Perl gives the applications programmer access to its internal symbol table, a little finesse and shameless disregard for immaculate code enable us to squirrel away the old function definition so the new one can be loaded.

Figure 1 shows what a basic `&autoload` function might look like. This version assumes that a function lives in a stand-alone file of the same name as that function with the customary `.pl` extension tacked onto the end of it.

```
1 sub autoload {  
2     local($fun) = $_[0];  
3     eval <<"EOCODE";  
4     sub $fun {  
5         warn "autoloading $fun from $fun.  
6         pl...\n";  
7         &swapload("$fun", "./$fun.pl");  
8         &$fun; # call new version of $fun  
9     }  
10    EOCODE  
11    die "autoload failure creating stub  
12    for $fun: $@" if $@;  
13 }
```

Figure 1: `autoload`

The `eval` statement at line 3 creates (but does not execute) the stub function, using the familiar '`<<`' notation from the shell as its argument, which means it will compile (`eval`) everything down to line 9. The `warn` statement at line 5 will be executed only when the stub is called, not when it's compiled by `eval`: it is there for debugging purposes only and should either be removed it or guarded with an "if `$debug`" for production code. Line 7 finally calls the newly loaded function. Because no parentheses were used, the autoloaded function will inherit the same argument list as we passed the stub ("`$file.bak`" in the example call to `&myfunc` above).

Now for the tricky part: coding the `&swapnload` function. The way this works is certainly implementation dependent: not only does it assume 4-byte pointers, it also knows that the 8th one in the symbol table is the pointer to the function. Under normal circumstances, code like this should be strenuously avoided. In this case, however, heroic measures are called for.

```

1 sub swapnload {
2 local($fun,$file) = @_;
3 local($tmp) = substr($_stub{$fun}, 7*4,
4);
4 substr($_stub{$fun}, 7*4, 4) =
5 substr($_main{$fun}, 7*4, 4);
5 substr($_main{$fun}, 7*4, 4) = $tmp;
6 require $file;
7 }

```

**Figure 2: swapnload**

The `$_stub{$fun}` entry is the actual symbol-table pointer to all objects of name `$fun` (like `$fun`, `@fun`, `%fun`, and `&fun`, to name just a few) in the stub package. (Packages are for statically-scoped, protected namespaces; there is no significance to the choice of the name `stub`. The choice of `main`, on the other hand, is significant — it assumes the function should exist within the `main` symbol table, a reasonable assumption for most programs.) Now index into that object skipping over 7 four-byte words to pull out a 4-byte substring, the real C pointer to the function definition. Once that step makes sense, it becomes more obvious that this code is simply exchanging two pointers, using the `$tmp` variable to hold the old one while the swap is done.

The `require` statement at line 6 will produce a fatal error if the file can't be loaded properly. This may seem harsh, but imagine if the VM system can't fault in part of a compiled program: the process should die because it can't be run. If this is truly undesirable behavior, either use the `do` form of file inclusion, or protect the `require` with an

`eval`, as in this code:

```
eval 'require $file'; warn $@ if $@;
```

One bit of awkwardness involved with this approach is that it assumes all functions to be autoloaded reside in separate files out on disk somewhere. For user-defined functions determined at run time, this is probably a good thing. For those that are just normal parts of the main program, though, it is inconvenient. It is easier to distribute a program as one self-contained file instead of scattering its pieces all across the disk.

To do self-contained autoloading, we make use of Perl's ability to process data supplied in the same file as program source. In this case, the data will be the code to be autoloaded. The special token `__END__` tells Perl to stop compiling the program as though it had hit end of file, and makes whatever comes after that available through the pre-opened filehandle `DATA`. Instead of autoloading from an external file, the program will seek into its own data area to find the function to load.

```

1 sub main'foo { &dataload; }
2 sub main'bar { &dataload; }
3 __END__
4 sub main'foo {
5   print "i am foo\n";
6 }
7 sub main'bar {
8   print "i am bar\n";
9 }

```

**Figure 3: dataload example**

Figure 3 is a snapshot of the set-up used for internal autoloading from the data area. For each function to autoload from `DATA`, declare a stub function as shown on lines 1 and 2. Beneath the `__END__` mark, place the real definitions for those functions. You could also write a function that created the stubs on the fly as we did in `&autoload` above.

```

1 sub fetch_function_addrs {
2 local($_);
3 local($start) = tell(DATA);
4
5 for (local($pos) = $start; <DATA>; $pos
6   = tell(DATA)) {
7   if (/^\s*sub\s+((\ew+)?(\w+))/) {
8     $Func_Addrs{$1} = $pos;
9   }
10 }

```

**Figure 4: fetch function addrs**

Before they can be autoloaded, we need to know where each function in the program resides. The function in Figure 4 computes this. It works by

scanning through the `DATA` handle in search of function definitions. When one is found by the pattern match in line 6, line 7 stores the previous line's seek address in an associative array indexed by the name of the function. Thus, the seek address for the function `&myfunc` would be found in `$Func_Addrs{ 'myfunc' }`. If a program wanted to find the locations of all its functions in the code instead of just the ones in the `DATA` portion, it could seek to 0 on `DATA` before starting the `for` loop.

The `&dataload` function itself is shown in Figure 5.

```

1 sub dataload {
2     local($fun) = (caller(1))[3];
3     local($addr) = $Func_Addrs{$fun};
4     local($code, $_, $tmp);
5
6     die("No address for $fun")
7     unless defined $addr;
8     seek(DATA,$addr,0);
9     while (<DATA>) {
10         $code .= $_;
11         last if /^\}/;
12     }
13     die "No \"sub $fun\" in code at
$addr:\n$code\n/" if !defined $code;
14     unless $code =~ /^sub\es+$fun/;
15
16     $tmp = substr($_, 7*4, 4);
17     substr($_, 7*4, 4) =
        substr($main{$fun}, 7*4, 4);
18     substr($main{$fun}, 7*4, 4) = $tmp;
19
20     eval $code;
21     die $@ if $@;
22     &$fun;
23 }
```

**Figure 5: dataload**

Line 2 uses the `caller` built-in to find out the name of the function that called it. This is the function whose stub definition got us here, and which we want to replace. Line 3 pulls out the address of the function, and line 6 makes sure that we really have a valid seek address, which we seek to in line 7. To read in the function, the loop starting at line 8 goes through `DATA` building up the function in `$code` until it finds a line beginning with a single '`}`'. (Note that this does assume a certain rigor in indentation style.) Lines 13 and 14 assure that a real function was found.

Lines 17 and 18 pull the swap-and-load trick previously described. Finally, it `evals` the code extracted from the data area, (line 20), checks to make sure it compiled correctly (line 21), and then calls the new function (line 22).

When doing either external or internal autoloading, some care must be taken in selecting which functions will be compiled after execution begins. Certainly the `&autoload` and `&dataautoload` functions must themselves be initially resident, but so also must be any functions they themselves should call, either directly or indirectly. Otherwise you have the sort of problem seen by confused operating systems that mistakenly page out their page-in routines.

In practice, it is better to avoid parsing of the data area every time just to load the function addresses. These can be cached to disk the first time the program is run, and loaded from the cache afterwards. Because Perl allows an associative array to be transparently mapped to a DBM file on disk, this can be both convenient and quick. Just be careful about regenerating the cache if the program is ever modified, or the seek addresses will be wrong. A good way to check this is to compare the modify times of the program and the cache, and perhaps store a checksum of the whole script within the cache as shown in Figure 6.

```

sub getdatasum {
    local($pos) = tell(DATA);
    seek(DATA, 0, 0);
    local($/) = undef;
    $Func_Addrs{'CHECKSUM'} =
        unpack("%16C*", <DATA>);
    seek(DATA, $pos, 0);
}
```

**Figure 6: getdatasum**

Another useful application of autoloading is reloading modified functions during debugging sessions. For short programs, this is unnecessary, but for long ones, you may want to change just a few functions in the running program instead of reloading it from scratch. To do this, just keep each function in its own file, and create a `Makefile` to manage them. When a function is out of date, `make` can be directed to signal the running program to direct it to autoload just those functions needing reloading.

## *Using GraphTab*

## **Using GraphTab for Medium Resolution Graphics on Dumb Terminals**

by Mark R. Horton

*<stargate!mark@cis.ohio-state.edu>*

## Introduction

Text and graphics are two ways to display information to users. Graphics applications can display pictures, logos, and different typefaces, but aren't as portable as text. Alphanumeric displays are universally available, but limited to text.

There are times when it would be useful to display a logo, graphic, banner message, or picture, but it has to go into a text file for display on an unknown terminal or printer. For example, electronic mail is generally restricted to alphanumeric contents. Tools have been written to draw pictures made up of ordinary printing text. For example, the banner command shows big letters made up of identical characters.

\$ banner UNIX # (R) of USL

Programs to plot functions on Teletype printers or to draw Snoopy calendars using block letters were very popular in the 1970s. Suns and BSD systems have programs for very large signs. However, such displays are often too large to fit on a screen, requiring the user to print the message on paper and hang it up on the wall sideways. This is great for signs, but not so good for signature files and logo displays when software starts up. It turns out it is possible to do considerably better than the display of large block letters, using only plain mixed case text. On an ordinary CRT screen, it is possible to display 98x160 graphics with a resolution good enough for many applications. If you have a Sun running SunOS 4.1 (or an earlier version) try typing:

```
$ /usr/lib/vfontinfo -m cm.i.11 UNIX
```

wqggw:	wggw	wqgg.	:wgg:	wggw	:wggp:	wqggw
j	.M'	j PM# .	.M'	.M	!M   ..dP"	
,M	dI	, P ' M#	dI	dP	!M   /'	
dP	P	dI 'ML, P		MI	, PM	
M	.dI	P	MdI	, P	, /F vM,	
'MxxpP'	x+   x	qP	xx+   x	xd+Fx	xM#xx	

There are many fonts available in different styles and sizes. In place of cm.r.12 try any font from except those ending in "r". The number in the font name represents the "point size" which is larger for bigger text. Here are some examples, showing the versatility of different fonts:

```
$ /usr/lib/vfontinfo -m B.14 Kludge
```

```
--gg-- : =gr---gg
MM . gp' MM
MM . dL' MM ~xx - ,xx . ,xx,MM . ,xx, . ,x . ,x,x
MMdMMb MM MM [MM . +P' /'MM . +P' "M" . +P" "
MM' 'M. MM MM [MM MM' MM [M] MM MM
MM . dL, MM MM [MM MM dfMM, +P" MM,
MM vM. MM YM. x+MM "M" . dMM M. . "M" . x
MM . dMM M. . "M" . x
v. .JM
```

Larger fonts have more bits and so they tend to look somewhat better.

```
$ /usr/lib/vfontinfo -m bocklin.28 Hack
```

Smaller fonts can be used to create legible text using less space than the banner command  
\$ /usr/lib/vfontinfo -m nonie.r.10 HELLO

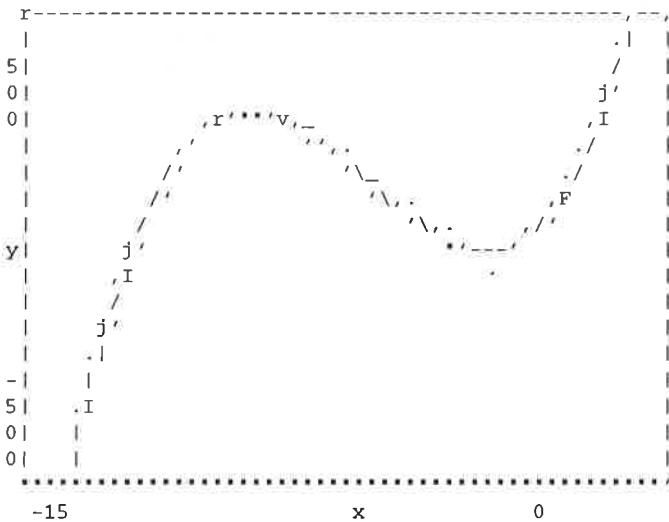
gl	jg	gpwww:	gl	gl	.xpwb,
M	M	M	M	M	jP' Y#
M+www+M		M+www	M	M	M
M	M	M	M	M	,M
M	M	M xxxxx,	M xxxxx	M xxxxx	'bpxP'

GraphTab

It is possible to create graphical displays for ordinary alphanumeric displays. In addition to the obvious use of Xs and blanks to get 24x80 resolution, finer resolutions are possible by making clever use of the shapes of the alphanumeric characters that are available. GraphTab is a technology to create such medium resolution graphical displays using only alphanumeric text. Since most characters are nearly twice as tall as they are

wide, a simple approach is to display a 2x1 rectangle (2 pixels) in a single character position. A blank represents both pixels being empty, a 'I' or 'M' shows both pixels being full, a " " or ' ' shows the upper pixel only, and a " , " or " ." shows the lower pixel only. Even higher resolution is possible. A single character position can represent a 4x2 rectangle, or 8 pixels. Since there are 256 possible patterns and only 95 printing characters, the display will be ambiguous, but usually the overall shape is more important than being able to pick out individual pixels. The GraphTab table was generated by considering each of the 256 possible 4x2 patterns and choosing an alphanumeric character that most closely resembles the shape of the desired pattern. While GraphTab images are not as sharp as images created with true graphics hardware, they can be much better than images drawn with large blocks of X's. The usual problem with low resolution graphics, legibility of the text drawn using graphics, is not necessarily a problem, since text labels can be drawn using real text. For example, consider a plot of a cubic function drawn using GraphTab:

$$y = x^3 + 10x^2 - 20x$$



Another use of GraphTab is to include digitized photos of people. This can be useful, for example, to include with email when it is not possible to meet the other person face-to-face. Some GUI systems, such as the AT&T DMD series, show a 48x48x1 picture of the people who recently sent mail to you. GraphTab can display these images at different resolutions, occupying either 12 or 24 lines on the screen.

**Ken Thompson, at zoom level 2:**

```

d;    7.
|M,_,-,_,-,_,-,xgIM
MM/MM+PF|MPMMMM|MM
MT|';' ;' T? :+MM.
|M;|';gg+MMgMMMM
jMM/MMMMMMMMMMMMM.
jMMMMMMMM#d+MMMMMM#,
TMMMMMMMMGM+MMMMMM+I
,+HMM#HMM#HMM#HMM#HMM#I,
|[ MM#MM#MM#MM#MM#MM#MM#M]>
/+#MM#MM#MM#MM#MM#MM#M[F

```

## Dennis Ritchie, at zoom level 1

## Using the vfontinfo Command

The `/usr/lib/vfontinfo` command, present in 4.2BSD and SunOS 4.1, incorporates the Graph-Tab technology to display the characters from bit-mapped fonts. (This is perhaps not the best use of GraphTab, but I am describing it here because it is widely available today.) Of course, Suns have real graphics hardware, but this command can be used from any ASCII terminal, over the network, and included in email.

You may find it in other systems derived from 4.2BSD. Not all Suns have vfontinfo ; it is optionally loaded from the standard distribution as part of the Versatec package, and your administrator may or may not have installed it when the system was brought up. The command and font library is not present in SunOS 5.0. There are several undocumented options to vfontinfo that can be used to create examples. Without any options, the command displays numeric data about each character in a particular font file from . For example:

```
$ /usr/lib/vfontinfo R.14
Font R.14, raster size 7594, max width 64, max height 40, xextend 24

      ASCII    offset    size   left   right    up   down   width
R.14    41    !    1221     26    -5      8    25      1     12
R.14    42    "    1247     18    -4     17    26   -17     19
R.14    43    #    1265     105   -3     23    25     10     26
.
```

Adding the `-v` (verbose) option causes `vfontinfo` to also show the bits of each character. (We can also add another argument to choose a few characters of interest.) This option uses GraphTab's 2 pixels/character method to show the bits.

```
$ /usr/lib/vfontinfo -v R.14 d|  
tail +5 | sed 's/.*/&& /g'
```

The tail command skips the headers, which may not be interesting. The sed command works around a bug in the command; at zoom level 1 it skips every other column. (A fixed and enhanced version of the vfontinfo source and man page can be found on ftp.uu.net in the directory which can be retrieved from the Internet with anonymous FTP. This version makes the above sed command unnecessary. The output shown here is from the new version.) The vfontinfo program has an undocumented zoom option to control the pixel size. The default zoom level, 1, uses 2 pixels/character. Zoom level 0 uses 2 characters/pixel. Zoom level 2 uses 8 pixels/character. Level 0 produces displays of big pixels. (GraphTab would use the sequence MM for each pixel, but vfontinfo uses to show the individual pixels more clearly for the font application.)

```
$ /usr/lib/vfontinfo -z0 -v R.14 d
```

Zoom level 2 is used to display small pixels. Each character represents an area of the graphics display four pixels high by two pixels wide. The same character fits into a much smaller space while retaining approximately the same shape.

```
$ /usr/lib/vfontinfo -z2 -v R.14 d
```

More usefully, the `vfontinfo` command has a message option to use the small pixel code to display banners.

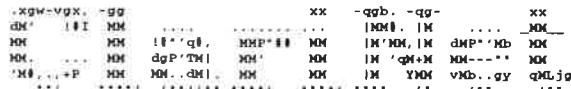
```
$ /usr/lib/vfontinfo -m R.10 GraphTab
```

```

    .w--.d1          -x      p-g--.ql      -x
    dI      !I          M      I      !I      M
    M      "M/'Yb j'``\,  "M/*'\b M r-`Yb   j'``\,  M/*'\b
    M.      -qV.  M'      xr-T'!  M T1  M      M      xr-T'!  M T
    .,+!  M.      ./!  M.,..P  M      M      M.      ./!  M.,..P
    .,.,.  .,.,.  M      M      M      M      M      M

```

The message option can be useful for creating compact banners in a wide variety of fonts. Inspecting `/usr/lib/vfont` and experimenting with different fonts and type sizes is both fun and useful. (Avoid fonts whose names end in "r"; these fonts have been rotated and cannot be directly displayed.) The ASCII file sent via E-Mail to ClariNet customers includes the graphic:



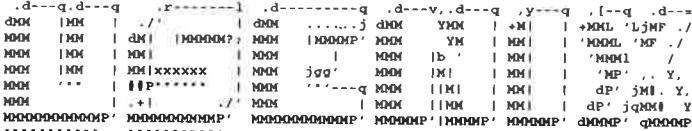
This graph was generated with a command called `vbanner` which is similar to

```
$ /usr/lib/vfontinfo -m B.10 ClariNet
```

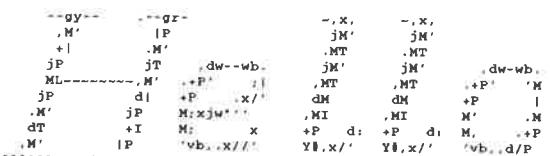
### More Examples

To find create a nice banner, you may need to try several things. Many of the fonts are already quite ragged, and the GraphTab process aggravates the raggedness. Some messages step on bugs in `vfontinfo`; it is necessary to keep the message to within 80 characters of output, and in some cases the last column may be cut off. (This bug has been fixed in the version available on UUNET.) Here are some examples of fonts that look relatively nice.

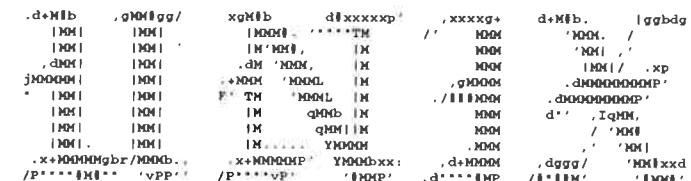
```
$ /usr/lib/vfontinfo -m shadow.16 USENIX
```



```
$ /usr/lib/vfontinfo -m I.18 Hello
```



```
$ /usr/lib/vfontinfo -m mona.24 UNIX .d.
```



```
$ /usr/lib/vfontinfo -m script.18 UNIX
```



A copy of volume 2C of the 4.1BSD or 4.2BSD manual includes the Berkeley Font Catalog, which includes samples of the fonts in this directory. To create your own sampler of 12 point typefaces, try

```
cd /usr/lib/vfont
for i in '/bin/ls' | sed 's/\.[0-9].*/' | uniq'
do
    echo $i
    /usr/lib/vfontinfo -m $i.12 UNIX
done | more
```

### Conclusion

GraphTab is a useful technique for displaying medium resolution graphics in situations that were intended for only alphanumeric text, such as E-Mail and Netnews. One widely available tool that uses GraphTab is the `vfontinfo` command. Beware, however, that SunOS 5.0/Solaris 2.0/System V release 4 does not have the `vfontinfo` command.

If you need the command, you might want to save the program and the fonts for future use, or get the version on UUNET. Other applications using GraphTab can make better use of the technology. A companion paper to this one is in preparation, describing the implementation of GraphTab and illustrating more general use of the technology.

# An Update of UNIX-Related Standards Activities

by Stephen Walli

Report Editor <[stephe@mks.com](mailto:stephe@mks.com)>

USENIX Standards Watchdog Committee

## Report on POSIX.0: The Guide to Open Systems Environments

*Kevin Lewis <[klewis@gucci.enet.dec.com](mailto:klewis@gucci.enet.dec.com)> reports on the July 13-17, 1992 meeting in Chicago, IL:*

First, let me indulge in some self-aggrandizement before going into the details about the POSIX.0 work. A little over a year ago, I projected that the guide document would be going into formal ballot in July 1992. (I am the co-chair of the working group and have a stake in this!) As of the writing of this report, July 16, 1992, the POSIX.0 guide has been sent out for formal ballot by the IEEE. I must add that this would not have been possible without a core of dedicated people within the group, acting as section leaders.

As for the work of this group at the July meeting in Chicago, there were two major areas of activity. The first was the development of the Guide document rationale. The rationale captures the group's memory on those issues it felt were significant and which it felt might surface during the formal ballot process. The audiences for this document will be the section leaders to assist them during ballot resolution, and future members of the working group who might need to understand the thinking of the group on these issues.

This document was completed during the meeting and will be available to the group prior to the October meeting in Utrecht during which the group will be resolving ballots.

The other major area of activity were discussions around the group's coordination with other standards organizations at the ISO level. The group is specifically concerned with WG15, the WG15 Rapporteur Group for Coordination of Profile Activities (RGCNA), and SC22. We have formally requested that the U.S. Technical Advisory Group (TAG) to WG15 seek review and comment of the formal ballot draft by WG15 and SC22. In addition, we asked the TAG to notify the WG15 RGCNA that several members of POSIX.0 would like to participate in their Utrecht meeting in October. The formal ballot draft, along with a cover letter highlighting questions for consideration during the review, was passed to the TAG.

Finally, for those who are interested, the POSIX.0 Ballot Group composition is:

Class	Number	Percentage
Academic	3	3.49%
General Interest	23	26.74%
Producer	24	27.91%
User	36	41.86%
Total	86	100%

We've gotten over the first two hurdles of establishing a balanced ballot group and getting our document out on time. Stay tuned to find out what the response is....

## Report on POSIX.2: Shell and Utilities

*David Rowley <[david@mks.com](mailto:david@mks.com)> reports on the July 13-17 meeting in Chicago, IL:*

### Summary

September the 16th, 1992 – that's the date people have been waiting for since the POSIX.2 working group was formed more than five years ago. It's the date the IEEE Standards Board is due to approve P1003.2 as an IEEE Full Use Standard. The standard includes both the "Dot 2 Classic" and "Dot 2a" components, previously balloted as separate standards. The IEEE Standard (based on the new Draft 12) is identical (at least from a technical standpoint) to ISO/IEC Draft International Standard 9945-2:1992.

NIST continues to work on a new FIPS (Federal Information Processing Standard) for POSIX.2, expected in draft form by early Fall 1992.

POSIX.2b work is progressing well, incorporating symbolic link support within a number of utilities and a new PAX archive format, and addressing a number of international concerns regarding locales.

Test assertion work continues. The POSIX.2 assertions have almost full coverage, and will go to ballot soon, perhaps as early as October. The POSIX.2a test assertion work is going well, though assertions for *vi* have not yet been attempted.

There is talk that the test assertion work will be renamed P2003.2 instead of the current P1003.3.2.

## Background

A brief POSIX.2 project description:

The base utilities of the POSIX.2 standard deal with the basic shell programming language and a set of utilities required for the portability of shell scripts. It excludes most features that might be considered interactive. POSIX.2 also standardizes command line and function interfaces related to certain POSIX.2 utilities (e.g., `popen()`, regular expressions, etc.). This part of POSIX.2 , which was developed first, is sometimes known as "Dot 2 Classic."

The User Portability Utilities Option, or UPUO , is an option in the base standard (previously known as POSIX.2a ). It standardizes commands, such as `vi`, that might not appear in shell scripts, but are important enough that users must learn them on any real system.

Some utilities have both interactive and non-interactive features. In such cases, the UPUO defines extensions from the base POSIX.2 utility. Features used both interactively and in scripts tend to be defined in the base utility.

POSIX.2b is a newly approved project which covers extensions and new requests from other groups, such as a new file format for PAX and extensions for symbolic links. It also includes resolution of items arising from comments by ISO Working Group 15. POSIX.2 is equivalent to the International Standards Organization's ISO DIS 9945-2 – the second volume of the proposed ISO three-volume POSIX standard.

## POSIX.2 Status

The ISO Draft International Standard 9945-2 that was approved at the May meeting of Working Group 15 is due to be approved at the IEEE on September 16th. There are no apparent road-blocks to the IEEE Standards Board approving the standard, but of course there are very few sure things in life.

POSIX.2 Draft 12 comprises the standard set of utilities ("Dot 2 Classic") and now includes the User Portability Utilities Option (previously "Dot 2a, User Portability Extensions"). Hal Jespersen has done a fine job integrating the two separately balloted standards into one epic tree-killing tome, coordinating it with the ISO 9945-2:1992 Draft International Standard, POSIX.2 's ISO equivalent. The implementors of the world owe Hal a debt of thanks for ensuring that the ISO and IEEE standards can be technically identical.

## FIPS and Certification

NIST continues to work towards a new FIPS (Federal Information Processing Standard) for POSIX.2 . Verifiable conformance to the standard is now the critical issue. Fortunately, it seems as though good progress is being made within the standards industry on coming up with a well-endorsed solution. X/Open has issued an RFQ (Request for Quotation) for an Integrator to put together a joint POSIX.2 and XPG4 Commands and Utilities verification suite. This work points towards there being a single validation suite for both the POSIX.2 and XPG4 implementations of the shell and utilities, again making life much easier for implementors and users alike. The XPG4 commands and utilities specification comprises a superset of the POSIX.2 utilities. The X/Open suite will allow verification of the XPG4 superset as well as the POSIX.2 subset.

NIST will likely point to this suite, once in place, as the yardstick for gauging conformance to the POSIX.2 FIPS.

The suite will likely be finished towards the end of 1993.

## PAX File Format

The group continued to define the new PAX file format, but are now intent on verifying the sanity of using the ISO 1001 tape format as a base format. A posting to "comp.std.unix" requested feedback and input as to the appropriateness of ISO 1001 , along with a request for alternate proposals. The proposals will be discussed at the Utrecht meeting in October.

The group also modified the proposal for codeset representation of filenames, user names, etc. contained in the archive. The format that will be used is now specified as UTF (UCS Transformation Format). A slight problem with this exists because the UTF description is contained in Annex F of the ISO 10646 Unicode standard, and is only informative rather than normative. The group is therefore (a little) hesitant to point to it, but feels the space savings and the inherent seamless ability to upgrade to the full 32-bit codeset (UCS4) overcomes these objections.

## Working Group 15 Requirements

The group also examined the Working Group 15 (ISO) requirements for the next revision, as outlined in Annex H of the ISO Draft International Standard 9945-2:1992. Most of the issues centered around the definition of locales, specifically codeset issues. A number of specific proposals are pending from the ISO member bodies, includ-

ing something similar to trigraphs for the *sh*, *awk*, etc., extensions to locale character class definitions, reincorporation of the substitute facility, relaxing of the restriction on NUL collating lower than all other characters, support for state-dependent characters sets (such as shift encoding), and a general character translation utility (perhaps X/Open's *iconv*).

These issues will be discussed further at the Utrecht meeting on October 22nd, 23rd (just before the next WG15 meeting).

### Test Methods

POSIX.3.2 Test Method work is progressing well, with almost all of the assertions corresponding to the current draft of POSIX.2. The group expects to go to ballot sometime around October.

Work on the UPUO test methods also progressed, with only a few gaps remaining. The daunting *vi* command still strikes fear in some that would approach it, and has not yet been addressed. This will be worked on at the Utrecht meeting.

### Report on POSIX.5: Ada Bindings to POSIX

*Del Swanson <dswanson@email.sp.unisys.com> reports on the July 13-17, 1992 meeting in Chicago, IL:*

The POSIX.5 group has been working to produce Ada language bindings to POSIX standards. As of June, 1992, the IEEE Standards Committee has approved the Ada binding to POSIX.1 as a standard, designated POSIX.5. It should be published as an IEEE standard by the end of the year. Congratulations all around to the working group, the ballot resolution committee, the ballotters, and all the supporting employers, spouses, lovers, etc.

At this time, it is not expected that this document will become an ISO standard, because of its format and derivation. POSIX.5 is a "thick" binding: it can be read by itself, since it duplicates the descriptions of all the functions, in addition to describing how they relate to the Ada language. And POSIX.5 is derived from the POSIX.1 C binding, since no Language Independent Specification (LIS) yet exists. ISO requires that language bindings be "thin," not duplicating any information present in the base document, and that they be bindings to an LIS.

TCOS-SS (the IEEE committee responsible for all POSIX standards) had previously agreed that POSIX.5 could be approved as an IEEE standard in its current form. It would not be submitted for ISO standardization. A new version of the standard (which will then be submitted for ISO stan-

dardization) will be produced after the LIS is approved, and after the revision of the Ada language, now expected to be finalized in 1994.

Meanwhile, there has been a reaction from the European community, and from members of ISO Working Group 9 (on Ada) that there should be an Ada binding of POSIX officially sanctioned by ISO. At the July POSIX meetings, therefore, we recommended to TCOS-SS that it suggests to ISO Working Group 15 (ISO POSIX) that POSIX.5 be approved as an ISO "Committee Document."

Now that the IEEE standard has been approved, it is incumbent upon the group to resolve interpretation questions. Officially, this involves the formation of an interpretation committee (on which nearly the entire group sits). The intent is to explain interfaces, elaborate semantic descriptions, and define the implications of problematic interface specifications. About ten interpretation requests have been received to this point. The TCOS approach is that this interpretation group adds nothing normative to the standard, even by logical extension. Any such specifications must be done by balloted revisions to the document.

The major current activity of the group is the development of bindings to the Real-Time Extensions standards being developed by the POSIX.4 group. The binding to POSIX.4 will be relatively straightforward. This is especially true since a draft thin binding to POSIX.4 has been prepared by one of our members at Florida State University with financing from the U.S. Army.

This draft has now been updated a couple of times by the group, and is ready to be massaged into IEEE format, with a few changes reflecting the latest POSIX.4 draft. This POSIX.20 draft 1 is planned to be circulated for mock ballot after the October meetings. Our goal is to have POSIX.20 approved as a standard hard on the heels of POSIX.4 LIS.

This schedule is somewhat of a change from our previous assumption that we would produce a unified binding to POSIX.4 and POSIX.4a (threads extensions). Our current direction is to proceed directly with balloting the binding to POSIX.4, and work concurrently on the binding to POSIX.4a. The advantages are that this reflects the document structure of the POSIX.4 group, that this approach will fill the needs of some users sooner, and that the approval of the POSIX.4a standard is likely to be significantly later than POSIX.4.

Meanwhile, we have also agreed to assist in the production of the POSIX.4 LIS. The new technical editor of this document has been a joint member

of the POSIX.4 and the POSIX.5 groups. The members of the POSIX.5 group are committed to help him and the POSIX.4 group to produce the LIS as quickly as possible.

The production of a binding to POSIX.4a is going to be significantly more complex, because of the interplay of two separate modes of intra-process concurrency, Ada tasks and POSIX threads. Complicating the issues is a difference of philosophy among members of the group, which is probably reflective of the community at large.

A key question that differentiates the philosophies: should operating system functions be visible in a binding if the language itself provides parallel functionality? Several other issues ensue. Should functions be visible that, if called directly, may interfere with the assumptions and operations of the language support library? Would it be acceptable to isolate such functions to emphasize their danger? Is it adequate (or acceptable) to assume that Ada compilers will allow calling such functions via language interface conventions?

One of the greatest technical challenges to the POSIX.4a binding is to determine the implications of interactions among processes in a multi-language environment. The feasibility of mapping tasks to threads is being demonstrated in prototype implementations. But some potential conflicts caused by the interactions of the two entities are becoming apparent.

We are assuming that these conflicts must be resolved, since at a minimum Ada programs will want to make use of libraries written in C, such as GUI and DBMS packages. We are starting to catalog such potential conflicts, which revolve around the creation and destruction of threads/tasks, parent-child relations of threads/tasks, and the handling of exceptional conditions. We have barely begun the resolution process.

Meanwhile, members of our group are involved with two efforts that are prototyping implementations of Ada bindings to the Real-Time Extensions (including threads). As it happens, this is not only valuable input to our effort, but a few problems have been found with the base document drafts that have been passed on to POSIX.4.

In preparation for the next meeting, we have volunteers to analyze issues with task/thread interactions, and to propose directions and bindings to synchronization and scheduling functions. We hope for significant progress on these issues, as well as completing preparations for the mock ballot.

## Report on POSIX.7: System Administration

*Bob Robillard <duke@cc.bellcore.com> reports on the July 13-17, 1992 meeting in Chicago, IL:*

### Overview of POSIX.7

Since this is the first snitch report on POSIX.7 in quite some time, I'll start with some background. (If you already know what POSIX.7's been up to for the past year or so, you can skip ahead some). POSIX.7 is one of the three POSIX "Base Standards" (POSIX.1 and POSIX.2 are the other two). It covers the kinds of commands typically found in section (8) of the man pages – things like *fsck* and *init*.

Early on, POSIX.7 decided to address distributed system administration, rather than just single machine administration, in the belief that networked computing is the way things are going. This has caused a great deal of trouble since distributed system administration is relatively new and perhaps less ready to be standardized than stand-alone administration. The hope, however, is that the final standard will be more useful.

In the last year, POSIX.7 broke its work into several pieces. Each area of system administration is getting its own document. The current POSIX.7 "sub-groups" are:

POSIX.7.1 – Printing Administration,  
POSIX.7.2 – Software Installation and Management, and  
POSIX.7.3 – User and Group Administration.  
POSIX.7.1 – Print Administration

The Printing group is probably the furthest along, since they held a mock ballot in June. The base for their document is the Palladium print system which was originally developed as part of MIT's Project Athena. It is now included in the Open Software Foundation's DME project. The document specifies print commands, a programming interface, and a set of managed object definitions. (More on these later.)

Palladium is the reference implementation of the ISO Document Printing Application Standard (DPA), currently in international ballot as a Draft International Standard (DIS) under working group ISO/IEC JTC1/SC18 (the official name of the ISO document is: Information technology – Text and office systems – Document printing application (DPA) – Part 1: Abstract-service definition and procedures, September 1991). It's a client/server distributed system.

One of the reasons for the mock ballot was to determine whether Palladium was an acceptable choice for a base. Since *lpr* and *lp* (both the pre-SVR4 and SVR4 versions) are much more widely used, the group was concerned that their standard would be voted down on "not-existing-practice" grounds.

The people in the mock ballot okayed Palladium. Eleven (11) said Palladium would be okay, nine (9) said it would be okay if some changes were made (changes the POSIX.7.1 group then adopted) and only five (5) were against it. Astute readers will note that this was a small mock ballot group, but it was at least a well-rounded one with 6 University people, 10 from computer or operating system vendors (NeXT, IBM, Sequent, USL, Sun, OSF, Intergraph, Fujitsu), and 4 from user companies (US West, Bellcore, British Telecom, Boeing).

The No votes on Palladium were particularly strong, however, so the group is still concerned. If you have an opinion on this either way, please contact the author.

### POSIX.7.2 – Software Management

The Software Management group is not far behind the printing group. The draft document is stabilizing and should go to mock ballot soon. It includes commands to install and upgrade software packages. It also includes managed object definitions, but no API.

The base of their standard is HP's *swinstall* tools and USL's *pkg* tools. Currently, the commands look more like the HP commands, but that is still in flux.

### POSIX.7.3 – User and Group Management

The User and Group Management subgroup is still in the early stages. They have been gathering submissions for their base documents, and have been trying to determine a course of action.

There has been some debate about whether User/Group Management is a mature enough area to standardize, and the POSIX Project Management Committee (PMC) suggested that this group publish a Guide rather than a standard. You can expect these issues to be cleared up in the near future, and a solid direction to form soon.

### Managed Objects?

All of the POSIX.7 documents are providing descriptions of "managed objects" for their area. I'm not an expert on this, but here's everything I know about it.

Managed objects are hot in distributed management. UI's Atlas, OSF's DME, HP's OpenView, the Object Management Group (OMG) — everyone who's anyone in the field is using them. I think they come from network management, where the "object" being "managed" was a physical thing (like a router, for example.)

The concept is that there's an object out there, and to do something, you send it messages. The Print document, for example, has the concept of a printer object. If you want to know what kind of paper a printer has, you send a message to the printer object and ask for its "media-supported" attribute. There are objects for print jobs, software packages, etc.

The idea is that these "managed objects" work well with distributed systems because you don't have to know where the printer is — the message sending mechanism deals with that. Also, they are an aid to interoperability, since all POSIX compliant software will have to support the same set of objects.

### Road Blocks

Fair warning: I'm now going to get up on a soapbox.

The next step for the POSIX.7.1 document would seem to be to go to ballot. There are, however, two things standing in its way. First, all documents need to have test assertions written before entering ballot.

Test assertions are statements about what a function or command does, written in such a way that someone could easily write a shell script or program to check that an implementation actually does the correct thing. For example: "If *lpr* is given the name of a non-existent file, it returns the following error...." (There are formatting details about test assertions, but that's the basic idea.)

Although having these test assertions is clearly valuable, writing them is a tedious, time consuming process, and it is likely to delay ballot by several meetings. Also, since many details of the commands and functions are likely to change during ballot, many of the assertions will need to be thrown away.

Less clearly valuable is the Language Independent Specification (LIS) of the function calls, which also needs to be written before a draft goes to ballot. The functions have to be abstracted from C to an invented specification format which is free of programming language dependencies.

The idea of this is to remove any parts of the API that are implicitly dependent on C syntax, such as return values from functions, pointer parameters, or the use of structures. Only the functionality should remain.

The group then writes a companion "C thin-binding," which doesn't describe what the functions do, it just talks about how the functionality described in the LIS is implemented in C.

I believe the goal of the LIS is to make it easier for people interested in an Ada or Fortran version of POSIX to write the appropriate language binding for it. Again, this is tedious and time consuming, and will likely eat up several meetings of POSIX.7's limited resources.

### **Report on P1224: X.400 API**

*Steve Trus <trus@duke.ncsl.nist.gov> reports on the July 13-17, 1992 meeting in Chicago, IL :*

#### **Introduction**

The Chicago meeting was productive for the P1224 working group, and we are very near the completion of the standardization of the P1224 and P1224.1 documents.

At the Chicago meeting the group:

- planned the next P1224 ballot resolution meeting,
- reviewed the JTC1 recommendations for the International Standardization of the IEEE X.400, POSIX.17 Directory Services, and Object Management APIs,
- planned future work for the P1224 group,
- presented the status of the IEEE balloting of P1224,
- presented the status of the IEEE balloting of P1224.1,
- resolved the ballot objections and reviewed the ballot comments for the P1224 and P1224.1 documents, and
- planned the recirculation of the P1224 and P1224.1 documents.

#### **P1224 Next Ballot Resolution Meeting**

The group will not meet with the other TCOS groups at the October meeting in Utrecht, NL. We agreed to meet November 16-20 at NIST (Gaithersburg, MD).

#### **International Standardization of the APIs**

JTC1 has recommended that the IEEE P1224 and P1003.17 working groups split each of the X.400, X.500 and Object Management API documents into four separate documents (Language Independent Specification, Test Methods for Language Independent Specification, C Language Binding, and Test Methods for C Language Binding). Additionally, JTC1 has recommended that the IEEE submit the X.400, X.500 and Object Management API documents to JTC1 for fast-track when they are approved IEEE standards, and that members of the P1224 and POSIX.17 working groups solicit international support for these IEEE standards in order to increase the likelihood of a successful fast-track. The P1224 group agreed to follow these JTC1 recommendations.

#### **P1224 Working Group Status and Future Plans**

The first recirculation of the P1224 document began on May 20 and it ended on June 19. The balloting pool consists of 73 members. The balloting for the P1224 document closed with 81% of the ballots returned and 78% of the eligible voters approved the document.

Plans for standardizing future X.400 related APIs were discussed. The X.400 API Association and X/Open will have stable base documents for a P7 and an EDI API by the end of 1992. Tentatively, we would like to begin converting these documents into IEEE standards at the January 1993 meeting.

#### **P1224.1 Balloting Status**

The P1224.1 balloting period began May 6 and it ended June 5. The balloting pool consists of 50 members. The balloting for the P1224.1 document closed with 77% of the ballots returned and 82% of the eligible voters approved the document.

#### **P1224 and P1224.1 Ballot Resolution and Recirculation**

The group spent three days resolving the ballot objections and reviewing the ballot comments for the P1224 and P1224.1 documents. The technical editor will incorporate the changes into the document.

A 10 day recirculation of the P1224 document was scheduled to begin October 4 and end October 14. A 30 day recirculation of the P1224 document was scheduled to begin October 10 and end November 9.

## Summary

The progress of the P1224 working group is very good. We hope to have the P1224 and P1224.1 standards complete early 1993. The primary function of the November meeting will be P1224 and P1224.1 ballot resolution.

## Report on The Proposed ROSE API

*David Cannon <D.Cannon@EXETER.AC.UK>  
reports on the July 13-17, 1992 meeting in Chicago, IL:*

A project authorization request (PAR) has been submitted to the Distributed Systems Steering Committee (DSSC) for review, covering the Transparent Remote Operations Interface (TROI) proposal. The first ROSE meeting presented the details of the proposal, and a second meeting on Friday was a BOF on the technical content.

The presentation was led by J.J. Cinecoe and Dan Shia. J.J. Cinecoe anticipated the obvious question of "why do we need a Remote Operations Service Elements (ROSE) API?" He proposed that the work of the TROI group was essential for two reasons:

- it provides vendor OSI application portability,
- there is a requirement by large corporate users who want to write specialised applications which are needed to be portable across multiple platforms.

ACSE (Association Control Service Element) is too complex for a user-programmable platform; ROSE is perceived to better fill the need, and offers a "true" peer-to-peer relationship with another end-system.

The purpose of the proposed project is to generate an IEEE API for the classes of operations defined by ROSE. It is intended to co-locate meetings of the group with both the OSE Implementors Workshop (OIW) and the IEEE POSIX meetings. If both of these options are used the group would be meeting on average every six weeks. [ed. – *The OSE in OIW is "Open Systems Environment"*]. This is the NIST supported group, which used to meet as the OSI Implementors Workshop, and has recently had its scope expanded.]

A quick overview of ROSE vs. RPC:

- RPCs tend to be very proprietary.

RPCs bundle together:  
interaction semantics  
data transfer

- ROSE unbundles these and provides a variety of synchronous and asynchronous classes of operation.

- transfer of user-defined data streams.

ROSE can provide an equivalent service to RPC across different platforms.

Dan Shia described the Computing Environment on OSI (CEO), of which TROI is one component. The aim is to enable the construction of highly efficient distributed concurrent systems by providing a very thin API over the top of the protocol engine, and is based on OSI ACSE (Association Control Service Element), ROSE, and ASN.1 (Abstract Syntax Notation 1).

The proposal is based on experience gained from an implemented, working testbed.

Someone wanted to know what the difference was between this proposal and the POSIX.12 (Protocol Independent Interfaces) Simple Network Interface proposal. Dan suggested that the main differences were user-defined data presentation and remote operation facilities.

Dan outlined the problems involved in using full ASN.1, which is unparsable, and went on to describe ASN.C. This incorporates a simplified data definition language enabling the automatic creation of data objects, together with the ASN.1 data manipulation language and can be handled, for example, by a C language preprocessor.

The ASN.C specification allows data-object specification through statements which can be mapped on to functions or to extensions of the C language. These could ultimately call XOMcreate to generate the data objects. XOM is being standardized by the IEEE's P1224 working group, and is based on X/Open's Object Management API.

Someone asked whether XOM could do the work of encoding the ASN.1 rules for a particular data-object. Dan said it could for public objects, but wasn't very good at handling user-defined objects.

Dan went on to describe some RPC shortcomings, which include the inability to support:

- all ASN.1 types
- callbacks
- multicast
- peer-to-peer interactions

He also described some limitations of XAP and P1238, (the IEEE's FTAM working group,) including:

- over-complexity for applications writers
- non-integrated naming service
- non-integration of IPC and ITC (Inter-Thread Communication) support.

He concluded that this led to the inherent attraction of the CEO system, which amongst others provides for:

- RPC (blocking) support
- request/reply (non-blocking) support
- multiple underlying protocol stacks
- peer-to-peer operations

The relatively high-level approach offers a number of plusses, for example, a short training period, rapid OSI application development, the ability to port existing applications to the OSI environment, and suitability for development of server applications.

In summary TROI is an attractive proposition; the main problem from an IEEE viewpoint is that much of the elegance is dressed in extensions to languages – ASN.1, C, and any other required language binding – and languages are not the province of TCOS-SS. (TCOS-SS, the Technical Committee on Operating Systems – Standards Subcommittee, is the organizing group within the IEEE for the POSIX related standards efforts.) If they can be shown to be justifiable and useful the APIs could be worked under the TCOS umbrella, but there are some fears that the API work alone may not offer substantially more than P1003.12 and the XOM work.

## Report on ANSI X3B11.1: WORM File Systems

Andrew Hume <[andrew@research.att.com](mailto:andrew@research.att.com)> reports on the May 11-15, 1992 meeting in Pasadena, CA:

### Introduction

X3B11.1 is working on a standard for file interchange on random access optical media: a portable file system for WORMs or rewritable optical disks. TC15 is a committee within ECMA that works on file system standards. This report covers the last three X3B11.1 meetings in Santa Clara, California, Denver, Colorado, and Pasadena, California and two recent TC15 meetings in Denver, Colorado and Reading, England. In brief, we have an ECMA standard!

Pardon my laggardly snitching; I have been snowed under this year. In trying to meet the deadline for the June ECMA General Assembly meeting, I have attended 5 standards meetings in the first six months of 1992 (all but one was a full week) and I redacted new drafts for every one.

### ECMA – 167

Editorially, ECMA-167 is arranged as five separate parts. Semantically, these form four independent standards. (Part 1 contains general references and definitions.)

Parts 1 and 2 describe a general scheme for recognising standards used to record the medium (is it ISO 9660, ECMA-167, or perhaps both?) and for recording boot blocks.

Parts 1 and 3 describe a volume structure standard, which includes support for volume labels, volume sets, volume partitions, and logical volumes (which may span multiple physical volumes).

Parts 1 and 4 describe how to record hierarchical file systems (assuming we have a suitable underlying volume structure scheme). The file system is approximately a POSIX (ISO 9945-1) file system augmented by extended attributes.

Parts 1 and 5 document the arcana of record-structured files. ECMA-167 has to support record-structured files, if only for backward compatibility with ISO 9660, and making it a distinct part allows other standards to easily use the same specification.

An important aspect of each of these parts is their interfaces. The input interface describes what the part needs in order to work. The output interface describes what the part allows you to specify (and perhaps use as input to another part). As an example, Part 5 (record structure) has a single input, the data space of a file, and two outputs, the identification of record formats and record display attributes.

### International Activity

There is a lot of international interest in volume and file structure standards, particularly for removable optical media. That is why our committee has an ISO standard as its main goal, rather than an ANSI standard. That is also why we have bent over backwards to solicit input from, and work with, Europe (ECMA), Japan (JNC), and ISO (SC15).

We reached our first major milestone on June 25 when the ECMA General Assembly accepted our draft as ECMA-167 by a vote of 30 yes, 0 no, and 1 abstention. Regrettably, the General Assembly chose not to forward the standard for fast-track processing within ISO at this time; it will probably do so at its December meeting.

With the exception of France, we do not expect any problems when ISO SC15 processes ECMA-167 as a DIS. France's objections draw mainly from a French company's claim that adopting the standard will have dreadful performance impact on that company's products. We have discussions ongoing about this and other issues but our basic response is twofold:

- Our standard is an interchange standard. There is no intent that integrated applications adopt our format for their internal data format. They might, however, adopt our format for the import and export of data. As a concrete example, we do not anticipate that Epoch's Infinite file server product will change to use our format for their disk format (although they could). However, Epoch might interchange files into and from their server in our format.
- While we have spent considerable effort to minimize the number of seek's to access files and their data, the bottom line is that for good performance, you will have to have some kind of cached database that maps file or directory names to disk addresses. Optical media, particularly 12in media, is just too big and too slow (although a cache helps relatively fast magnetic disks as well). We decided that a portable high-performance cache was a contradiction in terms and too hard to specify in any case, and so we left it to each implementation to decide what, and how, to cache.

### Future Activity

The work in ECMA and TC15 has one single focus, getting the standard into the ISO fast track process. From here on in, the process is purely political. Other than acting as a technical resource, I am pretty much a bystander now.

The process in X3B11.1 is, unfortunately, just as political. Because X3B11.1 is a sub-committee, our parent committee, X3B11, has to approve most of our official activities, and in particular, drafts for processing as ANSI standards and positions for the US TAG to SC15. Ordinarily, this is not a problem but recently, a couple of members of X3B11 starting throwing as many roadblocks in our way as possible. As ANSI has more procedures than probably any other standards organization in the world, this could mean considerable delay in

ANSI processing. As a result of all this hooey, X3B11.1 is changing its focus from technical issues to politics and has now scheduled its meetings concurrently with X3B11 so we can at least argue our own case and cut down on the amount of misrepresentations and falsehoods being made about our committee and its work. (I gave a well-received presentation on our work at the August X3B11 meeting and was present during discussions of X3B11.1's work; this was a real win.)

Just remember, the technical content of a standard is very important, but getting a draft through the standards process is just as important.

### Electronic Distribution of Standards/Drafts

Since I became technical editor of X3B11.1, my drafts have been available electronically by both *ftp* and email (*netlib*) from [research.att.com](http://research.att.com). (For *ftp*, login as *netlib*.) For details, get *index* from *research/memo*.

As far as our standard is concerned, there are three documents:

- The standard itself (121 pages including TOC and index). (Of course, it can't be the actual standard as ECMA owns the copyright on that but rather, the final draft of TC15; ECMA takes this draft and reformats it using a word-processor program and then publishes it.)
- A technical overview (12 pages). This gives a high level overview but has significant technical content.
- A programmer's guide (20 pages). A low level guide through the standard from a C programmer's point of view. It gives you enough details to design an implementation and do most of the implementation.

### Finale

Finally, we have a standard and can now complete our implementations. Although there is considerable procedural work to do, the hard stuff is finished. The technical work has been quite interesting, as has been the role of technical editor. (Mind you, I am scarred for life; I can read standards quite easily now and find myself tsk'-ing at poorly written ones.) Writing a precise description of a nontrivial system is obviously hard, but you never appreciate how hard it is until you do it and then have a whole bunch of folks ballot on it.

If you wish to comment on the standard, get a copy electronically, or contact me or the X3B11 chair (Ed Beshore) for a copy. I will make sure any comments sent to me go to the right folks.

If you would like more details on X3B11.1's work, you should contact either me <[andrew@rsearch.att.com](mailto:andrew@rsearch.att.com)>, 908-582-6262) or the committee chair, Ed Beshore <[edb@hpgrla.hp.com](mailto:edb@hpgrla.hp.com)>, 303-350-4826).

## Obtaining GNU Software

The GNU (GNU's Not UNIX) Project is developing a complete UNIX-compatible software system with freely redistributable source code. The rationale for GNU is explained in the GNU Manifesto (copies are available in the GNU Emacs manual and sources, and by asking [gnu@prep.ai.mit.edu](mailto:gnu@prep.ai.mit.edu)).

You are encouraged to get GNU software from or with others. GNU software is also available by *ftp* on the Internet and by *uucp* from *uunet* and other sites, ask [gnu@prep.ai.mit.edu](mailto:gnu@prep.ai.mit.edu) for details. If you cannot use one of these methods, you can order software directly.

Lately, there have been several additions and changes to the available GNU software. The major ones are detailed below. Almost all of the other older software has been updated to versions that have fewer bugs and support more UNIX systems.

- One new manual has been published documenting Calc, an extensible, advanced desk calculator and mathematical tool that runs under GNU Emacs (the source is on the Emacs tape).
- One new quick reference card has been published for GDB, GNU's Debugger.
- Some programs that used to be on the Emacs tape and the entire contents of the old Compiler tape have been moved to two new tapes: Languages and Utilities.
- New software on the Languages tape includes:
  - *dld*, a dynamic run-time linker;
  - *ae*, a profiling tool used with GCC;
  - and *gmp*, a library for arbitrary precision arithmetic on signed integers & rationals;
- New software on the Utilities tape includes:
  - GNU *ptx*, a permuted index generator, including KWIC;
  - Ghostview, an X11 user interface for the Ghostscript interpreter;

- *elvis*, a clone of the vi/ex UNIX editor;  
- *ms*, MandelSpawn, a parallel Mandelbrot program for the X window system;  
- *cpio*, *patch*, *screen*, *less*, *time*, *tput*, GNU find and GNU m4;  
- and the GNU collections of shell, file, text and font utilities.

- X11R5 is now distributed instead of X11R4.
- There is a new tape containing the freed source from BSD Networking 2 distribution.
- There is a new experimental tape (for the adventurous :-) containing software that is in beta-test. It currently contains GCC 2, GDB 4, the GNU Graphic utilities, as well as the Binary File Descriptor, GNU C and C++ libraries.

Many of the programs in the GNU Software distribution are covered by either the GNU General Public License or the GNU General Library Public License that permits everyone to have and run copies of them at no charge, and to redistribute copies under certain conditions designed to make sure that that all modified versions remain as free as the versions GNU distributes. The Licenses are usually in files named COPYING and COPYING.LIB.

*[The USENIX Association is printing this information as a service to the user community; no endorsement of GNU software is implied. – Ed.]*

# The Bookworm

by Peter H. Salus  
Sun User Group, Inc.  
<peter@sug.org>

## OSI and TCP/IP

Mike O'Dell brought Carl Malamud's *Stacks* to my attention, and I can't recommend it highly enough. Malamud's three volumes on analyzing networks and his many columns are well-known for their readability and his perceptiveness. *Stacks* is a really fine survey of a variety of interoperability topics. As someone who has never liked OSI, I was especially taken by Malamud's revision of the seven layers: Religion, Politics, Finance, Environments, Stacks, Interfaces, Substrates replacing: Application, Presentation, Session, Transport, Network, Data Link, Physical.

My personal feeling is that Bureaucracy should have a place here, but I'll leave that to someone else.

As I've said before, outside of the fact that it was "made in the USA," there was nothing wrong with TCP/IP, except that the big European and Asian PTTs didn't like it. Hence OSI, and the reason why Religion and Politics top Malamud's layers. Well, if you really need to know about it, Craig Hunt's *TCP/IP System Administration* is over 450 pages literally crammed with valuable information. What the protocols are, name service, system configuration, SLIP and PPP, routing, DNS/BIN, are only a few of the topics covered in detail. I found the DNS chapter clear, which I hadn't expected. Once upon a time, I actually looked at my sendmail.cf file. I hope to never have to do so again. I can't imagine wanting to write one. So Hunt's 45 pages on sendmail, which include the remark: "There is rarely any good reason to write a sendmail.cf file from scratch," were of great interest. It's clear that Hunt knows what he's writing about as well as being sensible. This is a useful volume in O'Reilly's new binding, which nearly does lie flat next to your terminal. (And there's a sample sendmail.cf file in Appendix E (pp. 427-438) for masochists.)

## C++: Beginning and Advanced

M. T. Skinner's *The C++ Primer* is the perfect easy introduction for the real beginner. There are a few places where Skinner may assume a little too much, but I found the examples clear and terse.

There's a DOS diskette available, too, but I don't own a DOS machine. On the other end of the scale, entirely, is Markku Sakkinen's doctoral dissertation: *Inheritance and Other Main Principles of C++ and Other Object-oriented Languages*. Chapter 5 (pp. 111-152) appeared in *Computing Systems* 5.1; several of the other chapters are previously published papers, too. But Sakkinen's discussions of what he calls "the darker side of C++" (Chapters 2 and 6) are fascinating; the one on the "Law of Demeter – the law of good style" is valuable beyond C++ and Eiffel. This is probably the sort of volume that a quasi-academic series should pick up.

## IBM's SAA

No, I'm not nuts. I think that you folks really need to know about a book like Michael Killen's *SAA and UNIX*. Killen is an IBM and AT&T consultant who clearly does not like UNIX. The book, which reads like a bunch of trade rag columns hastily taped together, is full of quotes from Peter Cunningham (of UI) and David Tory (President of OSF, to whom ten full pages are devoted), but no mention of the CSRG. There is a mention of "Berkeley 4.2 UNIX" (p. 28), with neither an explanation nor a listing in the index. The section on the Internet worm lays the blame to an unnamed programmer at Berkeley who "left a trapdoor." The description of the worm, etc., will embarrass those who are involved in security. This is a damaging book which will be read by managers who are innocent to its errors and misleading statements. This is a book that never mentions Brian Kernighan and in which Robert Kavner gets more mentions than Ritchie or Thompson. You should know about it, as travelers in the tropics should know about poisonous snakes, insects, and spiders.

## System V and Solaris 2.0

As an antidote, look at the second edition of *UNIX in a Nutshell: A Desktop Quick Reference for System V and Solaris 2.0*. Well put-together, utilitarian, complete. There's little else to say. Check it out.

## OSF's DCE

Another useful little volume is *Introduction to OSF DCE*. Like all the volumes that OSF has issued, this one is overtly authorless. A little detective work revealed that the credit should go to Jennifer Steiner, and she should get a lot. If your manager doesn't understand why a distributed computing environment is a "good thing," he or she should read at least the first few chapters (there are only four). There is also a glossary that

even a vice president should be able to comprehend. If we're working over a network, we need to know what the tools are. This compact volume is a useful one.

#### The Internet, again

First, and perhaps most important to the beginner, is Brendan Kehoe's *Zen and the Art of the Internet*. [In the last issue of this newsletter there was a review of the ftp-able version of this, but here's the book.] This is a small, solid book which will enable the neophyte to find his or her way through addressing, ftp, USENET, telnet, etc. Kehoe -- who has just joined Cygnus Support -- has done a very fine job in a very small volume (112pp.). There are things I am unhappy about in it (SLIP goes undefined, for example; USENIX is called "a group of UNIX [sic] enthusiasts" [p. 30]). But, by and large, this is a well-written, well-presented introduction. However, I must admit that a volume that appears in 1992, with a 1993 copyright date, ought to have eliminated SU (USSR) and DD (German Democratic Republic) from the list of country codes. Russia, for example, is now RU, Slovenia is SI, and Belarus is BY.

Ed Krol's *The Whole Internet Catalog & User's Guide* is nearly triple the size of Kehoe's. It is also written in a far drier style. And it packs a tremendous amount of information. If I match up some sections, Krol's chapters on 'Finding Software,' 'Finding Someone,' and 'Finding Anything,' are more detailed than Kehoe's brief discussions. Kehoe does fine on archie, but Krol does better on gopher and WAIS. I think that I would recommend Kehoe to the beginner and Krol to the more advanced user. I liked Krol's 25-page catalog of Internet resources (274-297; Aeronautics to Zymurgy) a lot.

#### Internationalization (= I18N)

I was looking forward to Dave Taylor's *Global Software*. I'm sorry to have to report that it didn't live up to my expectations. The volume (though it does contain some actual code) seems to be slanted towards the utility of marketing internationally and multiculturally, rather than to the availability and accessibility of hardware and software around the world. Taylor does a good job when he talks about localization vs. internationalization; he also discusses the differences among compile-time, run-time, and link-time

internationalization. The chapter on "International Standards Organizations" is overly superficial and occasionally less-than-accurate. Taylor tends to over generalize, as when he refers to the Middle East, as though there were but one version of Arabic script, as though Hebrew and Arabic posed the identical problems to the programmer, as though Turkish were written in a non-Roman script. Taylor lacks information where writing systems are concerned; he also presents an incomplete view of date/time and numerical representations. Taylor does get my thanks for using 'Hello, cat.' But it may well be the first book on I18N, and as such, get my thanks.

*Stacks*, Carl Malamud (Prentice Hall, 1992; ISBN 0-13-484080-1) \$35

*TCP/IP System Administration*, Craig Hunt (O'Reilly & Associates, 1992; ISBN 0-937175-82-X) \$29.95

*The C++ Primer*, M. T. Skinner (Silicon Press, 1992; ISBN 0-929306-11-2)

*Inheritance and Other Main Principles of C++ and Other Object-oriented Languages*, Markku Sakkinen (University of Jyva"skyla" [Finland], 1992; ISBN 951-680-817-4)

*SAA and UNIX*, Michael Killen (McGraw-Hill, 1992; ISBN 0-07-034607-0) \$39.95

*UNIX in a Nutshell: A Desktop Quick Reference for System V and Solaris 2.0* (O'Reilly & Associates, 1992; ISBN 1-56592-001-5) \$9.95

*OSF DCE* (Prentice Hall, 1992; ISBN 013-490624 1)

*Zen and the Art of the Internet*, Brendan Kehoe (Prentice Hall, 1992; ISBN 0-13-010778-6) \$22.00

*The Whole Internet Catalog & User's Guide*, Ed Krol (O'Reilly & Associates, 1992; ISBN 1-56592-025-2) \$24.95

*Global Software*, Dave Taylor (Springer Verlag, 1992; ISBN 0-387-97706-6) \$34

# **Advanced Programming in the UNIX Environment**

## **Advanced Programming in the UNIX Environment**

**by W. Richard Stevens**

**Addison-Wesley, 1992, ISBN 0-201-56317-7,  
pp. 744, Hardcover, \$50.50**

**Reviewed by Peter Collinson**

Hillside Systems

*<pc@usenix.org>*

If you write programs for a UNIX environment, then you need this book. Every so often someone writes a book that is a "standard", a book that everyone needs open on their desks to help them do their work. This is such a book.

The main idea of this book is to describe the programming interface to the UNIX system. This includes the system call interface and many of the functions provided in the standard C library. Of course, it's reasonable to ask "what is UNIX" in this context. The UNIX interfaces described represent the two main systems of interest at the moment: System V Release 4 and 4.4BSD.

4.4BSD is not available yet, so the author takes information from various BSD-derived systems: first, he uses the current version of BSD available at Berkeley; second, he looks at the BSD/386 system, derived from the Berkeley Net/2 release and finally, SunOS 4.1.[12]. He also doesn't neglect the standardization work that is being done. Often, he describes the POSIX way of doing things and then shows how extant systems differ from that. All examples are written in ANSI/C and he documents the standard routines available from that standard.

Your system documentation is designed to say "how things are." Books like Bach or Leffler, et al. are written to say "how things are implemented." This book sits between the two, saying "how to use what's there."

The book is big, big, big. It's 19 chapters and some appendices, around 750 big pages. To understand why this is, let's pick a chapter at random to see what you get for your money. Chapter 10 is all about signals. It starts with a basic description of what they are and how they are used in the system. It lists the signals that are available on all the systems. A nice touch in all the chapters is the inclusion of historical notes printed in an inset paragraph in small font. This goes a long way to

explain why things are the way that they are.

The Signals chapter continues by examining the normal interfaces to the service and provides an example. The book is stuffed with little example C programs that illustrate what is happening and how things are used. The programs are self-contained and can be typed in, alternatively you can get them using anonymous FTP from [ftp.uu.net/published/books/stevens.advprog.tar.Z](ftp://ftp.uu.net/published/books/stevens.advprog.tar.Z).

The chapter continues with a discussion on all the technical aspects of signal handling, and since this is an area where System V, BSD, and POSIX systems diverge, there is a lot to cover. Also included in the chapter is a discussion of several routines that use signals as the basis for their operation (e.g., *sleep*) or have to worry about signals (e.g., *system*).

The signals chapter is number 10; the chapter numbering is one of the few niggles I have with the book. The chapter number does not appear in the header or footer on each page, so if you are looking for a specific chapter it takes longer than it needs to. Other chapters in the book are:

1. The Introduction: a brisk skim through UNIX facilities;
2. UNIX Standardization and Implementations: this covers the various flavors of the UNIX system and the effect of the standardization efforts on the system;
3. File I/O: UNIX was designed as an experimental file system so nearly all respectable UNIX books have an early chapter on this, this chapter covers the basic I/O system calls;
4. Files and Directories: covers how filesystems work and the various calls that affect the state of files and directories;
5. Standard I/O library: covering the familiar I/O package;
6. System Data Files and Information: covers various files that are used to store system related information;
7. The Environment of a UNIX Process: describes the world of a UNIX process;
8. Process Control: how to make new processes;
9. Process Relationships: how process work together in the system, sessions, and the like;
10. Signals: enough said;
11. Terminal I/O: how to use teletypes and the like;
12. Advanced I/O: like non-blocking I/O and the *select* and *poll* functions;

13. Daemon Processes: what happens in the background and how to write a process to live there;
14. Interprocess Communication: pipes, System V IPC, message queues etc;
15. Advanced Interprocess Communication: more stuff and a big example program showing how things are done.

The remainder of the book consists of several example programs showing how to use some of the advanced facilities:

16. A Database Library: a replacement for dbm ;
17. Communicating with a PostScript Printer;
18. A Modem Dialer;

And finally:

19. Pseudo Terminals: how to use these magic beasts. What's wrong with the book? Not a lot.

The area of networking is completely missing, but you are supposed to go out and buy the other Stevens book for that.

I do sometimes feel that too much emphasis is placed on looking at other written material. For example, as someone who has scant knowledge of the System V way of doing things, I am interested to find out how things are done in the "Consider it Standard" way of doing things. I am often told: "look in the System V manuals." This feels like a Catch 22, I am looking in the book because I don't have the manuals. The text then seems to go on and describe what I needed to know anyway so maybe this feeling is unjustified.

I would not hesitate. Rush to your bookstore and buy one of these, you won't regret it.

## PrimeTime Freeware

---

### Issue 1-2 of Prime Time Freeware (cover date July, 1992)

The next issue of PTF is in production. Here is a summary of the issue; contact us <[ptf@cfcl.com](mailto:ptf@cfcl.com)> if you want more detailed information:

**Format:** two ISO-9660 CD-ROMs, bound into a 50+ page booklet. Each disc contains around 1/2 GB of compressed archives, annotation files, etc. The issue unpacks to around 3 GB (3000 megabytes).

**Content:** PTF is primarily a collection of UNIX-related freeware source code. Binary files and support for non-UNIX platforms are strictly incidental. There isn't room to list everything, but here are some of the bigger items:

*ada.xlib, Andrew, ANU NEWS, Athena, btool, CLM, CLU, CLUE, CLX, CMU Common Lisp comp.sources.{3b1,amiga,games,misc,reviewed,sun,u nix,x}, Condor, COOL, CRISP, dirt, Ezd, Epoch, Franz Lisp, GINA, GNU (prep: /pub/gnu/\*, DJG++, GNUish MSDOS, the Cygnus Solaris-2 and Vintage releases), Go (2D graphics library), Grass, Hyper-News, Icon (several OS's, plus examples), IMAP, INGRES, InterViews, ISODE, Kermit (tapes A-E), LispView, Lucid Emacs, Mach, MAEstro, magic, MH, NCSA Data Analysis Tools, NIHCL, Oaklisp, PARI, PCL, PCLU, Pine, PlaNet, Postie Pat, Q, SCHEME*

(asst. versions), Scorpion, Serpent, SR, SRC Modula-3, T, Tcl (Tk, expect, etc.), TIFF, TXL, UnixTeX, URT, UIT, VOGL, VOGLE, VOPL, VORT, wframe, WINTERP, WRL Modula-2, X11R5p13, XView

Because of the current legal hassle, we did \*not\* include either 386BSD or NET. We hope to include them on future issue, once the dust has settled a bit.

**Price:** \$60, plus shipping, handling, and applicable taxes.

USENIX members may purchase the issue at a discount for \$50. Here is a pricing summary; contact us for unusual cases, quantity discounts, more information and ask about the PTF Buying Plan.

The issue (two discs and a booklet) may be ordered from:

Prime Time Freeware  
+1 408-738-4832 (Voice), -2050 (FAX)  
415-112 N. Mary Ave., Suite 50  
Sunnyvale, CA 94086 USA  
<[ptf@cfcl.com](mailto:ptf@cfcl.com)>

# The Whole Internet User's Guide and Catalog

**The Whole Internet User's Guide and Catalog**  
by Ed Krol (O'Reilly and Associates, 1992,  
ISBN 1-56592-025-2) \$24.95

Reviewed by Billy Barron

On September 8th, O'Reilly and Associates will publish *The Whole Internet User's Guide and Catalog* by Ed Krol. Krol is best known for RFC 1118, *The HitchHiker's Guide to the Internet*. O'Reilly has said "that this is will be the biggest trade title O'Reilly has ever published" and they "believe it will be one of the more popular holiday gift-giving books for computer enthusiasts."

In the introduction, Krol says that this book is for professionals, but not computer professionals. I think he is missing a potentially large market by this statement. I know many computer professionals especially in the microcomputer arena who do not know anything about the Internet and IP networking, but need to learn about them in the next few months. Also, Krol said, "if I did my job in writing this book, the rest of what you need you should learn along the way." Let's see if he did.

I should start off by saying that I am reviewing a manuscript copy of the book. The manuscript has numerous typos, formatting problems, the resource guide is not complete, and the glossary is missing. For the remainder of this review, I will assume that these will be fixed by the final release as O'Reilly has told me they will be.

The introduction of the book says that you do not have to be a UNIX user to use this book, but almost all of the examples in the book are UNIX based. The mail section depends heavily on Berkeley mail and the news section on *nn*. When I read these sections, I feel like I am reading "Introduction to Berkeley Mail" and "Introduction to NN" chapters. My personal feeling is that the examples will cause more confusion to the novice Internet user who is not using these packages than it does good. If the user is using these packages, then these chapters will be excellent.

Related to this, the book seems to have a definite UNIX basis; the UNIX information tends to be accurate and complete for the most part. I found several errors, overgeneralizations, and oversights in the non-UNIX material. I pointed some of the bigger ones out to O'Reilly. They will

attempt to fix them before the public release of the book. O'Reilly is also considering making the book more oriented to the non-UNIX person in the second edition. However, I feel the first edition is lacking in its coverage of non-UNIX systems.

A few places in the book make me feel like I am reading about the perfect Internet. Krol breezes over numeric IP addresses, which is a good thing since many books go into it much too deeply (what end user actually cares what a Class B network is?), but then he goes on to say, "Don't worry; you don't need to remember numbers like these to use the network." I wish things were this way, but I know they are not. Many old, yucky TCP/IP packages support only the host file and not Domain Name Service (DNS). I know this software is out there in use because any time I mention my FTP site without also giving the numeric address I get several mail messages begging me for it from people without DNS. Also, many sites (e.g., my girlfriend's university) are not registered in DNS and hence numeric IP addresses are the only way to get there. Another example is, "Gopher does not allow you to access anything you couldn't get to already." While I agree this is the way it should be, it is not. I know of quite a bit of information that is only available via Gopher (e.g., UNT NewMan newsletter - yes, I am as guilty as the next person).

Stylistically, the book depends heavily on analogies. Therefore, if you learn well with analogies, this book will be good for you. The book contains many funny inside jokes if you know where to look especially for you PDQ Bach fans.

One of the strengths of the book in my opinion is when it talks about social issues on the Net. I cheered when I read Krol's comments on USENET censorship. The etiquette sections were also good.

The book suffers, however, from inconsistency in spots. Why is USENET News mentioned as being available under World Wide Web and not Gopher? (Note: USENET News in Gopher has been around since February or March.) Why is Knowbot mentioned and NetFind, which I think is an infinitely more useful tool, not? Why is something as basic as the incompatibilities between new and old talk not mentioned, but fancy FTP 'get' commands involving pipes mentioned? I was not even aware of these 'get' commands until I read the book.

The Gopher, WWW, WAIS, and resource catalog sections were a bold move. This information is a fast moving target. Fortunately, while I see minor spots that are dated in these sections, the chapters are still fairly current and useful from the end user's point of view. The section on dealing with WAIS searches that have gone awry is the best discussion on this topic I have seen. Actually, the WAIS and WWW chapters are among the best discussions of this material that I have seen.

As a keeper of an Internet resource catalog, I know that resource information goes out of date fast. This somewhat worries me about having a resource catalog in print. O'Reilly has said that they will be updating the book frequently, but the question remains how many people will buy the updates at \$24.95. On a more positive note, even I, with an incomplete version of the catalog section, found useful resources that I have not seen before. I think users will enjoy this section and find it useful as long as the information does not become obsolete. There is a vague note in the information sheet about the Resource Catalog being available on the Internet at a later date, but I do not know any of the details. If implemented correctly, it would vanquish my obsolete information fear.

While the "Dealing with Problems" has much good information, parts of it will make network administrators lose sleep. The book talks about fixing Ethernets, but the note saying that you (the user) may want to check with your network administrator before doing anything comes at the end of the chapter. I fear that many users will read up to the repairing part and not get to the note before attempting to do something.

To compare this book with the first/free edition of *Zen and the Art of Internet* I recently reviewed in this newsletter (I have not yet seen the second edition), I would have to say that this book is much longer and more complete. The other side to this is that *Zen* explains topics in a clear, compact format whereas Krol's is much more verbose. The most noticeable organizational difference is that *Zen* does not have a resource catalog or chapters on some of the newer Internet utilities: WAIS, WWW, or Gopher.

As you have guessed by now, I am lukewarm on this book. I would suggest buying it if you want a unique and useful resource guide, if you need information on WAIS, WWW, or Gopher, or if you are using Berkeley mail or 'nn' on a Unix box. Otherwise, give the *The Whole Internet User's Guide and Catalog* the same looking over that you give *Zen and the Art of the Internet* and some of the soon-to-be released books on the same topic. I would also advise you computer professionals out there to ignore the audience section of the book but take a look at it anyway if you need to get up to speed on the Internet. Finally, the second edition of this book should be worth reviewing again if it fixes the problems of the first edition.

#### Availability

*The Whole Internet User's Guide and Catalog* by Ed Krol will be available on September 8th. The price is \$24.95 and will be available direct from O'Reilly and Associates, Inc. or through their distributors. The ISBN is 1-56592-025-2.

# Winter 1993 Conference

San Diego, California  
January 25 - 29, 1993

## Tutorials

Monday, January 25

New! Essential Unix Programming,  
*Richard Stevens*, Consultant

New! Osfus Distributed Computing Environment, *David Chappell*, Chappell and Associates

Using, Managing, And Implementing Nfs,  
*Ralph Droms*, Bucknell U.

OSF1 Internals, *Thomas W. Doeppner, Jr.*, Brown U.

Programming With The X Window System,  
*Oliver Jones*, PictureTel Corporation

Symmetric Multiprocessing And Caching In  
UNIX Kernels, *Curt Schimmel*, Silicon Graphics,  
Inc.

System V Release 4.0 Internals Part 1- The VFS  
And Process Subsystems, *George Bittner and Steve Rago*, ProLogic Corporation

Topics In UNIX System Security,  
*Matt Bishop*, Dartmouth College

Essentials Of Practical Perl Programming,  
*Tom Christiansen*, CONVEX Computer Corp.

New! Topics In Advanced System Administration,  
1993 *Trent Hein*, XOR Computer Systems,  
*Rob Kolstad*, Berkeley Software Design, Inc., and  
*Evi Nemeth*, U. of Colorado, Boulder

Tuesday, January 26

UNIX Network Programming, *Richard Stevens*,  
Consultant

New! Osfus Distributed Management Environment *David Chappell*, Chappell and Associates

Distributed File System Administration With  
DCE/DFS, *Phil Hirsch*, Transarc Corporation

New! Tcl And Tk: A New Approach To X11 And  
Gui Programming, *John Ousterhout*, UC Berkeley

New! Micro-kernel Technology, *Lori Grob and Marc Rozier*, Chorus systèmes

System V Release 4.0 Internals Part 2 – The VM  
And I/O Subsystems, *Steve Rago and George Bittner*, ProLogic Corporation

Network Security: The Kerberos Approach, *Dan Geer*, Geer Zolot Associates, *Jon A. Rochlis*, MIT

Introduction To Threads And Threads Programming, *Nawaf Bitar*, Kubota Pacific Computer

**1/2 day: 9 am – 12:30 pm (includes lunch at 12:30 pm)**

More Topics In Advanced System Administration, *Trent Hein*, XOR Computer Systems, *Rob Kolstad*, Berkeley Software Design, Inc., and *Evi Nemeth*, U. of Colorado, Boulder

**1/2 day: 1:30 pm - 5 pm (includes lunch at 12:30 pm)**

Managing The Domain Name System, *William LeFebvre*, Northwestern U.

## Technical Sessions

Wednesday, January 27

**9:00 am - 10:20 Keynote Address**  
*Robert Carr*, Go Corporation

**10:45 am - 12:05 pm [track 1] Libraries and Links**  
Session Chair: *Tom Christiansen*, CONVEX Computer Corp.

Improved Libraries for Dictionaries and Abstract  
Graphs, *Stephen C. North, Kiem-Phong Vo*, AT&T Bell Labs

Linking Shared Segments, *W. E. Garrett, M. L. Scott, R. G. Bianchini, L. I. Kontothanassis, R. A. McCallum, J. A. Thomas, R. Wisniewskii, S. Luk*, University of Rochester

A Library Implementation of POSIX Threads  
under UNIX, *Frank Mueller*, Florida State U.

**10:45 am - 12:05 pm [track 2] New Views**  
Session Chair: *Peter Honeyman*, CITI, U. Michigan

Hello World... *Rob Pike, Ken Thompson*, AT&T Bell Labs

Es: A Shell with Higher Order Functions,  
*Paul Haahr, Adobe Systems & Byron Rakitzis*, Network Appliance Corporation

Jgraph: A Filter for Plotting Graphs in PostScript,  
*James S. Plank*, Princeton U.

**10:45 am - 12:05 pm [track 3] Invited Talk**  
Molecular Visualization, *Thomas Ferrin*, UC San Francisco

**1:30 pm - 2:50 pm [track 1] Tuning**

Session Chair: *Dinah McNutt*, Tivoli Systems, Inc.

Faster AFS, *Michael T. Stolarzuk*, U. of Michigan

The AutoCacher: A File Cache Which Operates at the NFS Level, *Ron Minnich*, Supercomputing Research Center

Pitfalls in Multithreading SVR4 STREAMS and Other Weightless Processes, *Sunil Saxena, J. Kent Peacock, Vijaya Verma, Mohan Krishnan*, Intel Multiprocessor Consortium

**1:30 pm - 2:50 pm [track 2] Tools**

Session Chair: *Saul G. Wold*, Sun Microsystems

Warlock- A Static Data Race Analysis Tool, *Nicholas Sterling*, SunSoft, Inc.

DUEL – A Very High Level Debugging Language *Michael Golan, David R. Hanson*, Princeton U.

The San Diego "Zoo": A Multicomputer Test Suite, *Chris Peak*, Locus Computing Corporation

**1:30 pm - 2:50 pm [track 3] Invited Talk**

Internationalization *Jeff Haemer*, Canary Software

**3:30 pm - 5:00 pm [track 1] Communications**

Session Chair: *Dave Taylor*, Sun World Magazine

PhoneStation, Moving the Telephone onto the Virtual Desktop, *Stephen A. Uhler*, Bellcore

Glish: A User-level Software Bus For Loosely Coupled Distributed Systems, *Vern Paxson, Chris Saltmarsh*, Lawrence Berkeley Lab

UNIX Services for Multilevel Storage and Communications over a Secure LAN, *Bruno d'Ausbourg, Christel Calas*, CERT-ONERA

**3:30 pm - 5:00 pm [track 2] Invited Talk**

Highlights from the USENIX MicroKernel Workshop and Other Kernel Architectures, April 1992 Chaired by *Lori Grob*, Chorus systèmes

**3:30 pm - 5:00 pm [track 3] Invited Talk**

TCP/IP System Administration Adnib, *Craig Hunt*, National Institute of Standards and Technology

## Thursday, January 28

**9:00 am - 10:20 am [track 1] Xbits**

Session Chair: *Mary Seabrook*, Open Systems Solutions, Inc.

A Smart Frame Buffer, *Joel McCormack, Bob McNamara*, Digital Equipment Corporation

Wafe - An X Toolkit Based Frontend for Application Programs in Various Programming Languages, *Gustaf Neumann, Stefan Nusser*, Vienna U. of Economics & Business Administration

Design and Implementation of a Multi-Threaded Xlib, *Carl Schmidtmann*, Digital Equipment Corporation consultant, *Michael Tao, Xerox, & Steven Watt*, Xerox consultant

**9:00 am - 10:20 am [track 2 Filesystems, I]**

Session Chair: *Dan Geer*, Geer Zolot Associates

The Design and Implementation of the Inversion File System, *Michael Olson*, UC Berkeley

Operating System Support for Portable Filesystem Extensions, *Neil Webber*, Epoch Systems

File Systems in User Space, *Paul Eggert, Twin Sun, D. Stott Parker*, UCLA

**9:00 am - 10:20 am [track 3] Invited Talk**

Resource Discovery and Network Measurement in the Global Internet, *Mike Schwartz*, U. of Colorado, Boulder

**10:45 am - 12:05 pm [track 1] Overhead**

Session Chair: *Rob Kolstad*, Berkeley Software Design, Inc.

UNIX Kernel Support for OLTP Performance, *Tom Rogers, Hyuck Yoo*, Sun Microsystems, Inc.

A Study of Overheads in DECstation Network Software, *Jonathon Kay, Joseph Pasquale*, UC San Diego

The BSD Packet Filter: A New Architecture for User-level Packet Capture, *Steve McCanne, Van Jacobson*, Lawrence Berkeley Laboratory

**10:45 am - 12:05 pm [track 2] I/O**

Session Chair: *Jeff Schwab*, Purdue University

The Organization of Networks in Plan 9, *Dave Presotto, Phil Winterbottom*, AT&T Bell Labs

Handling Removable Media in Solaris, *Howard Alt*, SunSoft, Inc.

An Advanced Tape Cataloging System for UNIX, Systems, *Christopher J. Calabrese*, AT&T Bell Labs

**10:45 am - 12:05 pm [track 3] Invited Talk**

From Blazon to PostScript, *Daniel V. Klein*, Lone-Wolf Systems

**1:30 pm - 2:50 pm [track 1] Kernel Improvements**

Session Chair: *J. Kent Peacock*, Intel Multiprocessor Consortium

Efficient Kernel Memory Allocation on Shared-Memory Multiprocessors, *Paul E. McKenney, Jack Slingwine*, Sequent Computer Systems, Inc.

An Implementation of a Log-Structured File System for UNIX, *Margo Seltzer, Marshall Kirk McKU-*

sick, UC Berkeley, *Keith Bostic, BSDI, & Carl Staelin*, Hewlett-Packard

Exploiting In-Kernel Data Paths to Improve I/O Throughput and CPU Availability, *Kevin Fall, Joseph Pasquale*, UC San Diego

**1:30 pm - 2:50 pm [track 2] Invited Talk**

A History of UNIX, *Greg Rose*, Australian Computing & Communications Institute

**1:30 pm - 2:50 pm [track 3] Invited Talk**

The Odin System, *Geoff Clemm*, Bellcore

**3:30 pm - 5:00 pm Presentation Followed by Panel Discussion:** Intellectual Property Participants include *William Ryan*, General Attorney, Intellectual Property Law, AT&T Bell Labs

## Friday, January 29

**9:00 am - 10:20 am [track 1] Information Discovery**

Session Chair: *Jim Duncan*, Pennsylvania State U.

Fremont: A System for Discovering Network Characteristics and Problems, *David C. M. Wood, Michael F. Schwartz*, U. of Colorado, Boulder

The Enterprise Distributed User Directory Service, *C. Mic Bowman, Chanda Dharap*, Pennsylvania State University

Essence: A Resource Discovery System Based on Semantic File Indexing, *Darren Hardy, Michael F. Schwartz*, U. of Colorado, Boulder

**9:00 am - 10:20 am [track 2] Monitoring**

Session Chair: *Dick Dunn*, eklektix

Hardware Profiling of Kernels, *Andrew McRae*, Megadata Corporation

A Randomized Sampling Clock For CPU Utilization Estimation and Code Profiling, *Steven McCanne, Chris Torek*, Lawrence Berkeley Lab

Fault Interpretation: Fine-grain Monitoring of Page Access, *Daniel R. Edelson*, INRIA

**9:00 am - 10:20 am [track 3] Invited Talk**

Highlights from the USENIX Filesystems Workshop, May 1992 Chaired by *Peter Honeyman*, CITI, U. of Michigan

**10:45 am - 12:05 pm [track 1] Filesystems, II**

Session Chair: *Matthew Blaze*, AT&T Bell Labs

UNIX Disk Access Patterns, *Chris Ruemmler, John Wilkes*, Hewlett-Packard

An Analysis of File Migration in a UNIX Supercomputing Environment, *Ethan L. Miller, Randy H. Katz*, UC Berkeley

HighLight: Using a Log-structured File System for Tertiary Storage Management, *John Kohl*, UC Berkeley & Digital Equipment Corporation, *Carl Staelin*, Hewlett-Packard, & *Michael Stonebraker*, UC Berkeley

**10:45 am - 12:05 pm [track 2] O/S Implementations**

Session Chair: *Steve McDowell*, Exlog, Inc.

An OSF/1 UNIX for MPP Systems, *David Black, Paulo Guedes, John LoVerso, Durriya Netterwala, Faramarz Rabii, Paul Roy*, OSF, *Michael Barnett, Brad Kemp, Mike Leibensperger, Chris Peak, Roman Zajciew*, Locus Computing Corporation

An Implementation of UNIX on an Object Oriented Operating System, *Yousef A. Khalidi, Michael N. Nelson*, Sun Microsystems, Inc.

The Nachos Instructional Operating System *Wayne A. Christopher, Steven J. Procter, Thomas E. Anderson* UC Berkeley

**10:45 am - 12:05 pm [track 3] Invited Talk**

MIME & Metamail: Moving Multimedia Mail into the Mainstream? *Nathaniel S. Borenstein*, Bellcore

**1:30 pm - 2:50 pm [track 1] Cache and Carry**

Session Chair: *David S. H. Rosenthal*, SunSoft, Inc.

The Design and Implementation of a Mobile Internetworking Architecture, *John Ioannidis*, Columbia U.

Mobile Computing Environment Based on Internet Packet Forwarding, *Hiromi Wada, Takashi Yozawa, Tatsuya Ohnishi, Yasunori Tanaka, Matsushita*

The Compression Cache: Using Online Compression to Extend Physical Memory, *Fred Douglis, Matsushita*

**1:30 pm - 2:50 pm [track 2] Invited Talk**

Highlights from the USENIX System Administration Conference (LISA VI), October 1992 Chaired by *Rob Kolstad*, Berkeley Software Design, Inc.

**1:30 pm - 2:50 pm [track 3] Invited Talk**

Object Databases: More Than a Shift in Modeling Paradigm *David J. Jordan*, AT&T Bell Labs

**3:30 pm - 4:00 pm Closing Remarks**

For registration and additional information please contact the USENIX Conference Office.

# **UNIX Applications Development Symposium**

## **Call for Participation: Toronto, Ontario, Canada March 29 - April 1, 1993**

*Co-sponsored by the USENIX Association and UniForum Canada.*

One of the major uses of UNIX today is the support, development, and execution of applications ultimately used in achieving end users' business goals. The current trends in large end-user organizations of downsizing major applications from older mainframes to less expensive, more powerful, and simpler, modern networked, machines lend UNIX a serious position in the commercial marketplace. Consequently, more and more computing and information systems professionals are encountering UNIX when developing and maintaining applications.

The purpose of this symposium is to expose the challenges of building and maintaining applications on UNIX platforms, to discuss solutions and experiences, and to explore existing practice and techniques.

This symposium will feature papers, invited talks, panel discussions, and tutorials on aspects of designing, building, testing, debugging, and maintaining applications within and for the UNIX environment. There will also be ample opportunity at this symposium to meet your peers and make contact with others with similar interests.

This symposium will provide valuable information to designers, programmers, and managers who are planning to port existing applications into the UNIX environment or move development and maintenance teams from proprietary environments to UNIX.

### **Important Dates for Refereed Paper Submissions**

Extended Abstracts Due: December 4, 1992

Notifications to Authors: December 16, 1992

Final Papers Due: February 12, 1993

### **Other Important Dates**

Pre-registration materials will be available in mid-January, 1993

### **Tutorial Program**

Mon. & Tues., March 29 -30, 1993

### **Technical Sessions**

Wed., March 31 - Thu., April 1, 1993

### **Birds-Of-a-Feather Sessions**

Tues., March 30 - Thu., April 1, 1993

### **USENIX Reception**

Weds. evening, March 31, 1993

USENIX is co-sponsoring this event with UniForum Canada, a non-profit membership organization.

### **Suggested Topics:**

Topics may include, but are not limited to:

Graphical User Interfaces – The X Window System - User Interface Design & Standards. Open Look, Motif, NeWS, and so on. What is a style guide? Importance of consistency and ease of use.

Porting Issues – Issues surrounding the tasks of porting an existing application to UNIX, as well as issues of making UNIX applications portable to other architectures and other platforms.

Networking – Client/Server design issues, etc.

Project Management – Using UNIX tools to support project management. CASE - What, When, Why, Who, How.

O/S Issues – Overcoming limitations set by hardware and operating systems.

Security – The impact of security features. Schemes for maintaining security within an application.

Transaction Processing – Implementing distributed transaction processing for UNIX applications.

Fourth Generation Languages – What advantages and disadvantages do 4GL's have in a UNIX environment?

Distributed Applications – How do you make the best use of existing UNIX functionality (such as e-mail) to build UNIX applica-

tions? What are the issues of building and/or using distributed databases?

Object Oriented Programming – Productivity, languages, techniques, case studies, etc.

Object Oriented Databases – Advantages, etc.

The Corporate Internet – High Speed for the Elite, or Connectivity for the Masses? ISDN, TCP/IP, OSI, UUCP. Governments, privateers, service providers, co-operatives, telecoms. Network philosophy - open road, tollbooths, freeloaders or lifeblood.

Delivering/Installing Applications – What's the best way? How to prevent piracy, worms, viruses, etc. How to do updates effectively and securely.

Testing & Certifying Binary Applications – Who does this? What does this achieve? How long does it take? Applications and POSIX.1 Conformance Testing.

Standards - ABI/API/ANDF – How, What, Where, When, Why? What are they? How are these standards used? How do they affect applications? What features does each have? What benefits are derived from using each? Where should they be used/followed? When will they be real? How do you keep up with new standards? Why are they necessary?

### **Submission Details**

Papers may feature real-life experiences, as well as research topics. Both case-study and technical papers will be accepted. Case studies should describe existing systems and include implementation details and may also include performance data where practical.

Submissions must be in the form of extended abstracts (1500-2500 words; 3-5 pages in length). Shorter abstracts might not give the program committee enough information to judge your work fairly and, in most cases, your submission will be rejected. Longer abstracts and full papers simply cannot be read by the committee in the time available. Feel free to append a full paper to an extended abstract; this is sometimes useful during evaluation. The extended abstract should represent your paper in *short form*. The committee wants to see that you have a real project, that you are familiar with the work in your area, and that you can clearly explain yourself.

Please note that presentations are usually scheduled to last 25 minutes. Your presentation should provide an overview of your paper and entice your audience to read it in the proceedings and

hopefully follow up on your solution, or take your advice into consideration.

Papers will be judged on technical merit, relevance to the theme, and suitability for presentation. Papers are welcome from software (and hardware) vendors who wish to share their innovative solutions and techniques, but be forewarned that product marketing will not be tolerated.

Persons interested in participating in panel discussions should contact <[woods@usenix.org](mailto:woods@usenix.org)>.

### **Tutorial Program**

Tutorial Coordinator: Dan Klein  
<[dkv@usenix.org](mailto:dkv@usenix.org)> Tel: 412-421-2332

Explore topics essential to successful use and development of UNIX and UNIX-like operating systems, X windows, networking and interoperability, advanced programming languages, and related areas of interest. The USENIX Association's well-respected tutorial program offers you introductory and advanced, intensive yet practical tutorials. Courses are presented by skilled teachers who are hands-on experts in their topic areas.

In an effort to continue to provide the best possible tutorial slate, USENIX is soliciting proposals for new tutorials. If you are interested in presenting a tutorial, contact the Tutorial Coordinator (see above).

### **Invited Talks**

Interim Invited Talks and Panel Coordinator:  
Greg Woods <[woods@usenix.org](mailto:woods@usenix.org)>

As part of the technical sessions, a series of invited talks provides introductory and advanced information about a variety of interesting topics, such as using standard UNIX tools and employing specialized applications. We welcome suggestions for topics as well as request proposals for particular talks. In your proposal, state the main focus, include a brief outline, and be sure to emphasize why your topic is of general interest to our community.

### **Birds-of-a-Feather Sessions**

BOF Scheduling: USENIX Conference Office  
<[conference@usenix.org](mailto:conference@usenix.org)>

Birds-of-a-Feather sessions (BoFs) bring together devotees of many varied disciplines for discussions, announcements, mingling, and strategy sharing during evenings at the symposium. Schedule a BoF in advance or on-site.

### **Work-in-Progress Reports**

WIPS Coordinator: Greg Woods  
[<woods@usenix.org>](mailto:<woods@usenix.org>)

These reports provide researchers with 10 minutes to speak on current work and receive valuable feedback. Present your interim results, novel approaches, or newly-completed work. Schedule your report in advance or on-site.

### **For More Information**

Materials containing all details of the technical and tutorial program, conference registration, hotel and airline discount and reservation information will be mailed in January of 1993.

For further information about the Symposium, contact the Program Chair:

### **Program Chair**

Greg A. Woods  
#3 - 46 Three Valleys Drive  
Don Mills, Ontario  
M3A 3B5, CANADA  
[<woods@usenix.org>](mailto:<woods@usenix.org>)  
+1 416 443-1734  
+1 416 595-5425 [FAX]

### **Current Program Committee:**

Rob Kolstad, Berkeley Software Design  
[<kolstad@bsdi.com>](mailto:<kolstad@bsdi.com>)  
Evan Leibovitch, Sound Software  
[<evan@telly.on.ca>](mailto:<evan@telly.on.ca>)  
Peter Renzland, Ontario Government  
[<peter@renzland.org>](mailto:<peter@renzland.org>)  
Dan Tomlinson, Compusoft  
[<compus!dan@uunet.uu.net>](mailto:<compus!dan@uunet.uu.net>)  
Greg Woods, Elegant Communications  
[<woods@usenix.org>](mailto:<woods@usenix.org>)  
Elizabeth Zwicky, SRI International  
[<zwicky@erg.sri.com>](mailto:<zwicky@erg.sri.com>)

---

## **Mach Symposium**

---

### **Preliminary Announcement & Call for Papers Santa Fe, NM April 19-21, 1993**

Extended Abstracts Due: December 4, 1992

#### **Background**

The use and influence of Mach on the operating systems community continues to grow. From its beginnings as a small research project, Mach has spread to become the basis for commercial products from a variety of vendors, and a key component of innovative research efforts in both academic and industrial environments. At the same time, research and development continue to evolve Mach itself. The community of researchers and developers working with Mach is proving to be a very productive source of innovative systems.

Activity in this field has been sufficiently widespread that the USENIX Association is pleased to once again sponsor a Mach symposium to bring

together researchers, engineers, vendors and users of Mach systems. We will encourage discussion of all past and present Mach-related research, development, production, and applications activities.

#### **Symposium Overview**

The symposium will be spread over three days. The first day will be devoted to tutorials on Mach 3.0, and will include both introductory/overview and advanced programming tracks. These tutorials should be of interest to both those desiring an introduction to Mach, and to programmers interested in learning how to take better advantage of Mach features. The following two days will concentrate on presentation of refereed papers on past and present Mach-related work. Long breaks between presentations will provide opportunities for informal discussion. Some time will be available for descriptions of work in progress.

Extended abstracts of 1500-2500 words (9000-15000 bytes or 3-5 pages) should be sent to David Black at the address below (those submitting hardcopy abstracts must send five copies). Shorter abstracts run a significant risk of rejection as there will be little on which the program committee can base an opinion. In addition to the extended abstract, authors must also supply an outline of the full paper and an estimate of its length.

A good extended abstract will contain the following information in one form or another:

Abstract	100-300 words (half a page) included verbatim in the final paper
Introduction	The problem; its importance; previous work
Solution	Issues, decisions, tradeoffs, rationale
Implementation details	
Evaluation	Performance results; effort required; lessons learned.

### Conclusion

The extended abstract allows the program committee to analyze the content of the proposed paper. An outline lists the headings, major points, and many minor points for each section of the actual paper.

The outline should provide an idea of the form and style of your paper. This layout is not cast in concrete; just submit enough material to convince the committee that they want to accept the paper!

Longer abstracts (up to and including full papers) will be reviewed; the program committee appreciates any additional material that makes it easier to predict the content, organization, and style of the final paper. Authors should exercise appropriate restraint in determining the amount of material to submit.

The submission package must include:

- The extended abstract
- Outline of rest of paper
- Cover letter, detailing:
  - Title of paper
  - Authors
  - Estimate of paper length
  - Contact author (liaison to program committee)
  - E-mail address and daytime phone number for contact author
  - Hours during which the daytime phone number can be used
  - Surface mail address
  - Optional FAX and home phone numbers

If hardcopy is being submitted, send five copies of the submission.

The submission should be sent electronically to [dlb@osf.org](mailto:dlb@osf.org), or by surface mail (five copies of abstract) to David Black at the address listed

below. Submissions made by FAX will not be accepted.

Electronic submissions must be in plain text that can be reviewed in the form submitted; the program committee does not have the time to run formatting tools (e.g., TeX, LaTe<sub>X</sub>) or to figure out why a printer refuses to print some PostScript document.

All submissions will be acknowledged. Authors of approved abstracts will be required to submit full-length papers (8-15 pages) approximately five weeks after notification of acceptance. Formatting guidelines will be provided.

Areas of interest include, but certainly are not limited to:

- Applications and support for programming languages
- Mach 2.5 and related systems (e.g., OSF/1)
- Mach 3.0 and servers
- Mach-based operating system implementation and emulation
- Use of Mach subsystems in other operating systems
- Multiprocessor and parallelization experience
- Distributed systems, including multicomputers, clusters, etc.
- Real Time
- Security
- Performance
- Productization experiences
- Comparisons of Mach with other operating systems ( e.g., Chorus, Sprite, Amoeba, V, and of course, UNIX)
- Future work

The program committee is especially interested in papers describing applications and/or system servers that take advantage of Mach features in addition to papers describing the evolution of Mach kernel technology. Submissions are strongly encouraged from efforts across the entire spectrum from research projects to product development efforts (including work that falls between these endpoints).

### Important dates:

Extended abstracts: December 4, 1992  
Notification to Authors: January 18, 1993  
Camera-ready, full papers: February 26, 1993

For further information about the symposium, contact the program chair at the address on the following page.

## **Program Chair**

David Black  
Research Institute  
Open Software Foundation  
1 Cambridge Center, 11th Floor  
Cambridge, MA 02142  
Voice: +1 (617) 621-7347  
FAX: +1(617) 621-8696  
E-Mail: *dlb@osf.org*

## **Program Committee**

David Black, *Open Software Foundation*  
David Golub, *Carnegie Mellon University*  
Alan Langerman, *Orca Systems, Inc.*  
Jay Lepreau, *University of Utah*  
Avadis Tevanian, Jr., *NeXT, Inc.*

# **SEDMS IV Preliminary Call**

### **Preliminary Call for Participation: Symposium on Experiences with Distributed and Multiprocessor Systems IV (SEDMS IV) San Diego, California September 1993**

*Sponsored by: The USENIX Association  
In cooperation with: ACM SIGARCH, SIGCOMM,  
SIGOPS and SIGSOFT (Pending)*

*IEEE-CS Technical Committees on Distributed Processing, Operating Systems, Software Engineering, and Design Automation (Pending)*

#### **Goals**

The goal of this symposium is to bring together individuals who have built, are building, or will soon build distributed and multiprocessor systems. SEDMS IV will provide a forum for individuals to exchange information on their experiences, both good and bad, including experiences with coding aids, languages, debugging and testing technology, reuse of existing software, and performance analysis. The presentations should emphasize the lessons learned from use of such systems and tools.

Extra-long breaks between sessions and work-in-progress presentations will be provided to facilitate a workshop-like atmosphere during parts of the symposium. We will also have discussion panels on submitted themes.

#### **Submissions**

Six copies of each submission or panel proposal should be sent to the program committee chair (address below) to arrive no later than April 27, 1993. Submissions of full papers are invited on any topics related to the theme of the symposium. The committee will give preferential con-

sideration to submissions describing experiences with actual systems. Papers describing purely theoretical work will not be accepted. Panel proposals should include a description of the relevance to the goals of the SEDMS, and the qualifications of the participants suggested.

#### **Important Dates**

Submissions due	April 27, 1993
Notifications mailed	June 14, 1993
Camera ready copy due	July 20, 1993

#### **For Further Information, contact**

Peter Reiher (General Chair)  
Computer Science Dept.  
Boelter Hall, UCLA  
Los Angeles, CA 90024  
(310) 206-8696  
*<reiher@wells.cs.ucla.edu>*

David Cohn (Program Chair)  
Computer Science and Engineering Dept.  
University of Notre Dame  
Notre Dame, IN 46556  
(219) 239-6694  
*<dlc@cse.nd.edu>*

#### **Program Committee**

John R. Nicol, *GTE Laboratories, Inc.*  
Volker Tscharmer, *GMD FOKUS Berlin*  
Dag Johansen, *University of Tromso*  
Karsten Schwan, *Georgia Tech*  
Partha Dasgupta, *Arizona State University*  
Brett Fleisch, *UC, Riverside*  
David Pitts, *UMass Lowell*  
Debra Hensgen, *University of Cincinnati*  
John Barr, *Motorola*  
Marc Pucci, *Bellcore*  
Michael Scott, *University of Rochester*  
Mike O'Dell, *Bell Communications Research*  
Roy Campbell, *University of Illinois*  
Ed Lazowska, *University of Washington*

# LISA Groups

## BACK BAY LISA

A new New England forum, Back Bay LISA (BB LISA), met for the first time on July 29th. The group will mirror the activities of the West Coast's Bay LISA group - by holding regular meetings covering system and network administration topics.

The first meeting was attended by over 60 people from Boston and the surrounding areas. Tom Coppeto from the MIT Network Operations Group gave a presentation on Network and User Administration at MIT Athena. The talk was a great success.

The group will be meeting monthly; the next meeting is August 26th. Initially, no dues are payable, and everyone is welcome to attend meetings.

There is a mailing list, [bblisa@inset.com](mailto:bblisa@inset.com), which will carry announcements and discussions. To join the mailing list, send email to [bblisa-request@inset.com](mailto:bblisa-request@inset.com).

Future meeting topics will include:

- Analyzing System Administration Costs
- INN and Netnews Administration
- All about Obfuscated C
- Bibliographic Tools
- Project Athena
- What is World.STD.COM?
- Managing an Archive Server
- Running a Thinking Machine
- WAIS: Wide Area Archive Servers
- EFF: The Electronic Frontier Foundation
- Sendmail: the Future – Security from the FBI's Perspective
- Why and How to Join the Internet

For more information, join the mailing list, or contact:

JR Oldroyd  
[jr@inset.com](mailto:jr@inset.com)  
(617) 890 4930

Peg Schafer  
[peg@bbn.com](mailto:peg@bbn.com)  
(617) 873 2626

Rich \$alz  
[rsalz@osf.org](mailto:rsalz@osf.org)  
(617) 621 7253

Lisa Bloch  
[lab@sug.org](mailto:lab@sug.org)  
(617) 232 0514

Tom Heft  
[heft@husc.harvard.edu](mailto:heft@husc.harvard.edu)  
(617) 495 1273

## BAY LISA

The Bay-LISA group meets monthly to discuss topics of interest for administration of sites with more than 100 users and/or computers. The meetings are free and open to the public.

### General Meeting Information:

**Date:**  
Third Thursday of every month at 7:30 PM.  
Please do not arrive before 7 PM.

**Place:**  
DEC Digital Santa Clara III  
2465 Mission College Boulevard  
Santa Clara

From 101 take the Great America Parkway/Bowers Avenue exit. Take Great America Parkway, keep right. Come to the first light, turn right and keep left. At the first light, turn left into the parking lot of the office complex where Santa Clara III (WRO3) is. The main entrance is right off a fountain in the center. The meeting room is on the first floor.

**Info:**  
Send e-mail to [baylisa-info@sysadmin.com](mailto:baylisa-info@sysadmin.com), or you may contact:

Bjorn Satdeva  
(408) 241-3111  
[bjorn@sysadmin.com](mailto:bjorn@sysadmin.com)

Video tapes of previous meetings are available for Bay-LISA members. Send e-mail to [baylisa-info@sysadmin.com](mailto:baylisa-info@sysadmin.com) for further information.

# Publications Order Form

## Conference & Workshop Proceedings

Qty	Proceedings	Member Price	Non-Member Price	Subtotal	Overseas Postage	Total
<b>WINTER/SUMMER CONFERENCES</b>						
<input type="checkbox"/>	San Antonio	Summer '92	\$23	\$30	\$14	\$
<input type="checkbox"/>	San Francisco	Winter '92	30	39	22	\$
<input type="checkbox"/>	Nashville	Summer '91	32	38	22	\$
<input type="checkbox"/>	Dallas	Winter '91	28	34	18	\$
<input type="checkbox"/>	Anaheim	Summer '90	22	22	15	\$
<input type="checkbox"/>	Washington, DC	Winter '90	25	25	15	\$
<input type="checkbox"/>	Baltimore	Summer '89	20	20	15	\$
<input type="checkbox"/>	San Diego	Winter '89	30	30	20	\$
<input type="checkbox"/>	San Francisco	Summer '88	29	29	20	\$
<input type="checkbox"/>	Dallas	Winter '88	26	26	15	\$
<input type="checkbox"/>	Phoenix	Summer '87	35	35	20	\$
<input type="checkbox"/>	Washington, DC	Winter '87	10	10	15	\$
<input type="checkbox"/>	Atlanta	Summer '86	37	37	20	\$
<input type="checkbox"/>	Denver	Winter '86	25	25	15	\$
<input type="checkbox"/>	Portland	Summer '85	45	45	25	\$
<input type="checkbox"/>	Dallas	Winter '85	15	15	10	\$
<input type="checkbox"/>	Salt Lake City	Summer '84	29	29	20	\$
<input type="checkbox"/>	Washington, DC	Winter '84	25	25	15	\$
<input type="checkbox"/>	Toronto	Summer '83	32	32	20	\$
<input type="checkbox"/>	San Diego	Winter '83	28	28	15	\$
<b>LARGE INSTALLATION SYSTEMS ADMINISTRATION</b>						
<input type="checkbox"/>	LISA V	Sept. '91	20	23	11	\$
<input type="checkbox"/>	LISA IV	Oct. '90	15	18	8	\$
<input type="checkbox"/>	LISA III	Sept. '89	13	13	9	\$
<input type="checkbox"/>	LISA II	Nov. '88	8	8	5	\$
<input type="checkbox"/>	LISA I	April '87	4	4	5	\$
<b>C++</b>						
<input type="checkbox"/>	C++ Conference	Aug. '92	30	39	20	\$
<input type="checkbox"/>	C++ Conference	Apr. '91	22	26	11	\$
<input type="checkbox"/>	C++ Conference	Apr. '90	28	28	18	\$
<input type="checkbox"/>	C++ Conference	Oct. '88	30	30	20	\$
<input type="checkbox"/>	C++ Workshop	Nov. '87	30	30	20	\$
<b>SECURITY</b>						
<input type="checkbox"/>	UNIX Security III	Sept. '92	30	39	11	\$
<input type="checkbox"/>	UNIX Security II	Aug. '90	13	16	8	\$
<input type="checkbox"/>	UNIX Security	Aug. '88	7	7	5	\$
<b>MACH</b>						
<input type="checkbox"/>	Mach Symposium	Nov. '91	24	28	14	\$
<input type="checkbox"/>	Mach Workshop	Oct. '90	17	20	9	\$

Continued - see reverse

Qty	Proceedings	Member Price	Non-Member* Price	Subtotal	Overseas Postage	Total
<b>DISTRIBUTED &amp; MULTIPROCESSOR SYSTEMS (SEDMS)</b>						
___	SEDMS III	Mar. '92	30	36	\$_____	20
___	SEDMS II	Mar. '91	30	36	\$_____	20
___	SEDMS	Oct. '89	30	30	\$_____	20
<b>GRAPHICS</b>						
___	Graphics Workshop V	Nov. '89	18	18	\$_____	10
___	Graphics IV	Oct. '87	10	10	\$_____	10
___	Graphics III	Nov. '86	10	10	\$_____	5
___	Graphics II	Dec. '85	7	7	\$_____	5
<b>OTHER WORKSHOPS</b>						
___	File Systems	May '92	15	20	\$_____	9
___	Micro-Kernel & Other Kernel Arch.	April '92	30	39	\$_____	20
___	UNIX Transaction Processing	May '89	12	12	\$_____	8
___	Software Management	Apr. '89	20	20	\$_____	15
___	UNIX & Supercomputers	Sept. '88	20	20	\$_____	10

Discounts are available for bulk orders. Please inquire.

Total price of Proceedings \_\_\_\_\_ \*\*  
 Calif. residents add sales tax \_\_\_\_\_  
 Total overseas postage \_\_\_\_\_  
 Total enclosed \_\_\_\_\_

\*\*If you are paying member price, please include member's name and/or  
 membership number \_\_\_\_\_

#### PAYMENT OPTIONS\*

\_\_\_ Check enclosed - payable to USENIX Association

\_\_\_ Charge my: \_\_\_ VISA  \_\_\_ MC  Account # \_\_\_\_\_ Exp.Date \_\_\_\_\_

\_\_\_ Purchase order enclosed

Signature \_\_\_\_\_

\* Outside the USA? Please make your payment in US currency by one of the following:

- Check - issued by a local branch of a US Bank
- Charge (VISA, MasterCard, or foreign equivalent)
- International postal money order

#### Shipping Information

#### Ship to:

Please allow 2 - 3 weeks for delivery. Overseas orders  
 are shipped via air printed matter.

Please mail or fax this order form with payment to:

USENIX Association  
 2560 Ninth Street, Suite 215  
 Berkeley, CA 94710  
 FAX 510/548-5738

• If you are not a member and wish to receive our membership information packet, please check this box.

# Membership Application

## M E M B E R S H I P   I N F O R M A T I O N

Any individual or institution may become a member by filling out an application form and paying the appropriate annual fee.

There are five classes of membership:

**Student:** \$20  
Open to any full-time student at an accredited educational institution. A copy of the current student I.D. card must be provided.

**Individual:** \$65  
Open to any individual or institution. Individual Members may vote.

**Corporate:** \$300  
Corporate Membership is open to any individual or institution.

**Educational:** \$160  
Educational Membership is open to accredited educational institutions.

**Supporting:** \$1000  
Open to any individual or institution that wants to support the Association to a greater degree than through the Corporate Membership fee.

Corporate, Educational and Supporting members receive all services available to Individual Members, plus copies of the proceedings from all conferences and workshops that are held during the term of membership.

## M E M B E R S H I P   A P P L I C A T I O N

New  Renewal

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_ Country \_\_\_\_\_

Phone \_\_\_\_\_ email address: \_\_\_\_\_

- \$20 Student (full-time)  
*(with copy of I.D. card)*  \$160 Educational Institution  
 \$65 Individual  \$300 Corporate  
 \$1000 Supporting

### PAYMENT OPTIONS

- Check enclosed payable to USENIX Association.  Purchase order enclosed (Educational and Corporate members only).  
 Please charge my:  Visa  MasterCard



Account # \_\_\_\_\_ Exp. Date \_\_\_\_\_

Signature \_\_\_\_\_

Outside the U.S.A.? 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

- Please send me information on purchasing USENIX Software Distribution Tapes. 9/92  
 Please send me information on purchasing the Second Berkeley Software Distribution Tape (version 2.11).

### USENIX Mailing List

- I do not want my address made available to other members.  
 I do not want my address made available for commercial mailings.

CORPORATE, EDUCATIONAL AND SUPPORTING MEMBERS MUST COMPLETE THE FORM ON THE REVERSE SIDE.

# Local User Groups

The Association will support local user groups by doing a mailing to assist in 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](mailto: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) 278-2573  
[brent@CSUFresno.edu](mailto:brent@CSUFresno.edu) or [csufres!brent](mailto:csufres!brent)

Commerical institutions or individuals:  
Gordon Crumal (209) 251-2648  
[csufres!gordon](mailto:csufres!gordon)

## CA - Orange County:

Meets the 2nd Monday of each month

UNIX Users Association of Southern California  
Paul Muldoon (714) 556-1220 ext. 137  
New Horizons Computer Learning Center  
1231 E. Dyer Rd., Suite 140  
Santa Ana, CA 92705

## CO - Boulder:

Meets monthly at different sites. For meeting schedule, send email to [friug-info@friug.org](mailto:friug-info@friug.org).

Front Range UNIX Users Group  
Software Design & Analysis, Inc.  
1113 Spruce St., Ste. 500  
Boulder, CO 80302  
Steve Gaede (303) 444-9100  
[gaede@friug.org](mailto:gaede@friug.org)

## FL - Coral Springs

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

## FL - Western:

Meets the 1st Thursday of each month.

Florida West Coast UNIX Users Group  
Richard Martino (813) 536-1776  
Tony Becker (813) 799-1836  
[mcrsys!tony](mailto:mcrsys!tony)  
Ed Gallizzi, Ph.D. (813) 864-8272  
[e.gallizzi@compmail.com](mailto:e.gallizzi@compmail.com)  
Jay Ts (813) 979-9169  
[unet!pdn!tscs!metran!jan](mailto:unet!pdn!tscs!metran!jan)  
Dave Lewis (407) 242-4372  
[dhl@ccd.harris.com](mailto:dhl@ccd.harris.com)

## FL - Orlando:

Meets the 3rd Thursday of each month.

Central Florida UNIX Users Group  
Mikel Manitius (407) 444-8448  
[mike@aaa.com](mailto:mike@aaa.com)

## FL - Melbourne

Meets the 3rd Monday of every month.

Space Coast UNIX User's Group  
Steve Lindsey (407) 242-4766  
[lindsey@vnet.ibm.com](mailto:lindsey@vnet.ibm.com)

## KS or MO - Kansas:

Meets on 2nd Monday of each month.

Kansas City UNIX Users Group (KUUG)  
813B St.  
Blue Springs, MO 64015  
(816) 235 5212  
[mlg@cstp.umkc.edu](mailto:mlg@cstp.umkc.edu)

## 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  
Mark Landry (404) 365-8108

## MI - Detroit/Ann Arbor:

Meets on the 2nd Thursday of each month in Ann Arbor.

Southeastern Michigan Sun Local Users Group  
and Nameless UNIX Users Group  
Steve Simmons office: (313) 769-4086  
home: (313) 426-8981  
[scs@lokkur.dexter.mi.us](mailto:scs@lokkur.dexter.mi.us)

**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) 220-2427  
*pnessutt@dmshq.mn.org*

**MO - St. Louis:**

St. Louis UNIX Users Group  
P.O. Box 2182  
St. Louis, MO 63158  
Terry Linhardt (314) 772-4762  
*unet!jgalstl!terry*

**NE - Omaha:**

Meets monthly.

/usr/group/nebraska  
P.O. Box 31012  
Omaha, NE 68132  
Phillip Allendorfer (402) 423-1400

**New England - Northern:**

Meets monthly at different sites.

Peter Schmitt 603) 646-2085  
Kiewit Computation Center  
Dartmouth College  
Hanover, NH 03755  
*Peter.Schmitt@dartvax!dartmouth.edu*

**NJ - Princeton:**

Meets monthly.

Princeton UNIX Users Group  
Mercer County Community College  
1200 Old Trenton Road  
Trenton, NJ 08690  
Peter J. Holsberg (609) 586-4800  
*mccc!pjh*

**NY - New York City:**

Meets every other month in Manhattan.

Unigroup of New York City  
G.P.O. Box 1931  
New York, NY 10116  
Peter Gutmann (212) 618-0973  
*peterg@murphy.com*

**OK - Tulsa:**

Meets 2nd Wednesday of each month.

Tulsa UNIX Users Group, \$USR  
Stan Mason (918) 560-5329  
*tulsix!smason@drd.com*

Mark Lawrence (918) 743-3013  
*mark@drd.com*

**TX - Austin:**

Meets 3rd Thursday of each month.

Capital Area Central Texas UNIX Society  
P.O. Box 9786  
Austin, TX 78766-9786  
*officers@caactus.org*  
Tom Painter (512) 835-5457  
*president@caactus.org*

**TX - Dallas/Fort Worth:**

Dallas/Fort Worth UNIX Users Group  
660 Preston Forest, Suite 177  
Dallas, TX 75230  
Kevin Coyle (214) 991-5512  
*kevincd@shared.com*

**TX - Houston:**

Meets 3rd Tuesday of each month.

Houston UNIX Users Group  
(Hounix) answering machine (713) 684-6590  
Bob Marcum, President (713) 270-8124  
Chuck Bentley, Vice-president  
(713) 789-8928 *chuckb@hounix.uucp*

**WA - Seattle:**

Meets monthly.

Seattle UNIX Group Membership Info.  
Bill Campbell (206) 947-5591  
6641 East Mercer  
Mercer Island, WA 98040-0820  
*bill@celestial.com*

**Washington, D.C.:**

Meets 1st Tuesday of each month.

Washington Area UNIX Users Group  
9811 Mallard Drive  
Laurel, MD 20708  
Alan Fedder (301) 953-3626

**CANADA - Toronto:**

143 Baronwood Court  
Brampton, Ont. Canada L6V 3H8  
Evan Leibovitch (416) 452-0504  
*evan@telly.on.ca*

## LISA Groups

### Back Bay LISA

Forum covering all aspects of System and Network Administration, for large and small installations. Meets Monthly, various locations in Boston

JR. Oldroyd  
The Instruction Set  
601 Trapelo Road  
Waltham MA 01254  
(617) 890 4930  
[r@inset.com](mailto:r@inset.com)

Mailing list:[bblisa@inset.com](mailto:bblisa@inset.com)  
List Requests:[bblisa-request@inset.com](mailto:bblisa-request@inset.com)

### BAY LISA

The Bay-LISA group meets monthly in Santa Clara, CA, to discuss topics of interest for administration of sites with more than 100 users and/or computers.

Send e-mail to [baylisa-info@sysadmin.com](mailto:baylisa-info@sysadmin.com), or you may contact:  
Bjorn Satdeva  
(408) 241-3111  
[bjorn@sysadmin.com](mailto:bjorn@sysadmin.com)

## Computing Systems Coming Attractions

The Summer '93 issue (5:3) will be mailed in October. The editors expect that Fall issue (5:4) to be published by mid-December. Prompt renewal of your membership will ensure that you receive these issues promptly. Here's a preview of the articles that will soon appear:

### *Computing Systems* Vol. 5:3

Fine-Grained Access Control in a Transactional Object-Oriented System -- L.-F. Cabrera, A.W. Luniewski & J.W. Stamos (IBM)

Choices, Frameworks and Refinement -- R.H. Campbell, N. Islam & P. Madany (UIll.)

Implementing Atomic Objects with the RelAX Transaction Facility -- M. Mock & R. Kroeger (GMD) & V. Cahill (TC/Dublin)

Architectural Support for Persistent Object Systems -- J. Rosenberg (U.Sydney)

Casper: a Cached Architecture Supporting Persistence -- F. Vaughan, T. LoBasso, A. Dearle, C. Martin, C. Barter (U. of Adelaide)

Corrigendum to Sakkinen (5.1)

### *Computing Systems* Vol. 5:4

#### SPECIAL ISSUE: Internet Services

Guest Editorial -- Peter Deutsch (McGill U.)

Distributed Indexing of Autonomous Internet Services -- P.B. Danzig, S.-H. Li & K. Obraczka (USC)

A weak-consistency architecture for distributed information services -- R.A. Golding (UCSC)

The Prospero File System -- B.C. Neuman (ISI, USC)

A Comparison of Internet Resource Discovery Approaches -- A. Emtage (Bunyip Info. Sys.), B. Kahle (Thinking Machines), B.C. Neuman (USC), M.F. Schwartz (U.Colorado)

[possibly one more article, to be determined]

# Calendar of Events

## 1992

- Oct 19-23 \* LISA VI, Long Beach, CA  
19-23 IEEE 1003, Utrecht, Netherlands  
26-30 Interop, San Francisco, CA  
27-30 ISO/IEC JTC1 SC22 WG15  
Reading, UK  
Nov 25-27 EurOpen/UniForum  
Utrecht, Netherlands  
Dec 5-11 DECUS, Las Vegas NV  
7 Sun User Group, San Jose, CA  
UKUUG/UKnet, Manchester, UK

## 1993

- Jan 11-15 IEEE 1003, New Orleans, LA  
25-29 \* USENIX, San Diego, CA  
Feb 22-24 Sun Open Sys. Expo, Chicago, IL  
Mar 15-18 UniForum, San Francisco, CA  
29- \* UNIX Applications Development  
Apr 1 Toronto, Canada  
19-21 \* Mach III, Santa Fe, NM  
19-23 IEEE 1003  
May 3 - 7 EurOpen, Seville, Spain  
Jun 5-11 DECUS, Atlanta, GA  
21-25 \* USENIX, Cincinnati, OH  
Jul 12-16 IEEE 1003  
  
Sept 20-21 \*Micro-Kernels II  
23-24 \* SEDMS IV  
Oct 18-22 IEEE 1003  
25-29 Interop, San Francisco, CA  
Nov 1-5 \* LISA VII  
Autumn EurOpen/UniForum  
Utrecht, Netherlands  
Dec 4-10 DECUS, San Francisco, CA

## 1994

- Jan 17-21 \* USENIX, San Francisco, CA  
Feb 14-17 UniForum, Dallas, TX  
Mar 23-25 UniForum, San Francisco, CA  
Apr 18-22 EurOpen  
May 7-13 DECUS, New Orleans, LA  
6-10 \* USENIX, Boston, MA  
Sep 12-16 Interop, San Francisco, CA  
Autumn EurOpen/UniForum  
Utrecht, Netherlands  
Nov 12-18 DECUS, Anaheim, CA

## 1995

- Jan 16-20 \* USENIX, New Orleans, LA  
Feb 21-23 UniForum, Dallas, TX  
May 1-5 EurOpen  
13-19 DECUS, New Orleans, LA  
Jun 19-22 \* USENIX, San Francisco, CA  
Nov 2-8 DECUS, San Francisco, CA

## 1996

- Jan 22-26 \* USENIX, San Diego, CA  
Mar 11-14 UniForum, San Francisco, CA  
May 18-24 DECUS, Orlando, FL  
Nov 16-22 DECUS, Anaheim, CA

This is a combined calendar of planned conferences, workshops, and standards meetings related to the UNIX operating system. If you have a UNIX-related event that you wish to publicize, please contact [login@usenix.org](mailto:login@usenix.org). Please provide your information in the same format as above. This calendar has been compiled with the assistance of Alain Williams of EurOpen.

\* = events sponsored by the USENIX Association.

ACE: Advanced Computing Environments  
ACM: Association for Computing Machinery  
AFUU: Association Francaise des Utilisateurs d'UNIX

AUUG: Australian UNIX Users Group  
DECUS: Digital Equipment ComputerUsers Society  
ECUG: European C++ User Group

EurOpen: European Forum for Open Systems  
GUUG: German UNIX Systems User Group  
IEEE: Institute of Electrical and Electronics Engineers

IETF: Internet Engineering Task Force  
Interex: Internat'l Assoc.- Hewlett-Packard Comp.Users  
JUS: Japan UNIX Society

LISA: USENIX Systems Administration Conference  
NIST: National Institute of Standards & Technology  
SEDMS: Symposium on Experiences with Distributed  
and Multiprocessor Systems

Sinix: Singapore UNIX Association  
UKUUG: United Kingdom UNIX Systems Users Group

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

**Second Class Postage  
PAID  
At Berkeley, California  
and Additional Offices**

POSTMASTER: Send address changes to ;login:, USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710

---

## **What's Inside?**

---

***File Systems Workshop Report***  
***Winter 1993 Conference Program***  
***Standards Activity Update***