Getting Things Done

In those dark days at the end of projects (and occasionally while waiting for a delivery of a new purchase), my mind contemplates the difficulty of ever finishing something. Sometimes, I conclude that it’s a miracle that the world works as well as it does.

I have an expectation that when I purchase an audio CD that almost every one of its 660 million bytes will be correct. I believe that when I purchase a computer or a program, they will work with little error. Likewise appliances, automobiles, and furniture. I expect extraordinary service at my bank, from the hardware store clerk, and from package delivery services.

I have enough trouble just trying to put the correct postage on a package, yet Federal Express manages to deliver hundreds of thousands of them in 24 hours and track them all in real-time. When I contemplate the amount of engineering and general creativity that goes into software products, peripherals, computer systems, or any other product that one ultimately purchases, it’s mind-blowing.

Consider this newsletter. For any given issue, we have one or two dozen contributors (feature writers), the features editor (me), a “supervising editor” (Ellie Young), the “do-all-the-real-work” editor (Carolyn Carr), a proofreader, a printing house, and a mailing house. My interface to the world of getting ;login: “out the door” is via electronic mail and telephone. I hardly ever see any paper. I send off articles, talk on the phone, and watch my USMail box. Like clockwork, a wonderfully formatted issue shows up every two months. I have no idea how it happens, other than via my interfaces. It’s astounding.

If putting together a small newsletter is as challenging as it appears to me, the difficulty of a large project (e.g., landing on the moon for the first time) staggers my imagination.

Civilization is truly amazing.

RK

The closing date for submissions to the next issue of ;login: is April 15.
General Information

\textit{\texttt{login}}: is the official newsletter of the USENIX Association.

\textit{\texttt{login}}: (ISSN 1044-6397) Volume 18, Number 2 (March/April 1993) 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 \textit{\texttt{login}}. Subscriptions for nonmembers are $50 per year. Second-class postage paid at Berkeley, CA and additional offices. POSTMASTER: Send address changes to \textit{\texttt{login}}, USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.

Contributions Solicited

You are encouraged to contribute articles, book reviews, and announcements to \textit{\texttt{login}}. Send them via email to \texttt{<login@usenix.org>} or through the postal system to the Association office. Send SAGE material to \texttt{<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 \texttt{<kolstad@usenix.org>}
Ellie Young, Staff Editor \texttt{<ellie@usenix.org>}
Carolyn S. Carr, Managing Editor and Typesetter \texttt{<carolyn@usenix.org>}
Stephen Walli, Standards Report Editor \texttt{<stephe@mks.com>}
Bryan MacDonald, SAGE Editor \texttt{<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: \texttt{<office@usenix.org>}

Copyright \textcopyright{} 1993 USENIX Association. The Association acknowledges all trade references made herein.

\textit{\texttt{login}}: was produced on a SUN 3/50 with Framemaker 3.0 software provided by Frame Technology and printed on a QMS 860 donated by Quality Micro Systems. The newsletter is printed on recycled paper.

Conferences & 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
\texttt{<conference@usenix.org>}

Tutorials

Daniel V. Klein, Tutorial Coordinator
Telephone: 412/421-2332
\texttt{<dtk@usenix.org>}

Supporting Members

Frame Technology, Inc.
Matsushita Graphic Communication Systems
Mt Xinu
Network Computing Devices, Inc.
Quality Micro Systems
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 \texttt{board@usenix.org}

President:
Stephen C. Johnson \texttt{<scj@usenix.org>}
Vice President:
Michael D. O'Dell \texttt{<mo@usenix.org>}
Secretary:
Evi Nemeth \texttt{<evi@usenix.org>}
Treasurer:
Rick Adams \texttt{<rick@usenix.org>}

Directors:
Eric Allman \texttt{<eric@usenix.org>}
Tom Christiansen \texttt{<tchrist@usenix.org>}
Lori Grob \texttt{<gro@usenix.org>}
Barry Shein \texttt{<bzs@usenix.org>}

USENIX Association

Executive Director
Ellie Young \texttt{<ellie@usenix.org>}

2 \textit{\texttt{login}}: March/April 1993
Continuing a tradition started by Kirk McKusick, this letter will make a few observations about USENIX in 1992, and some thoughts about future directions.

1992 was a good year for USENIX. Financially, we finished the year solidly in the black after a couple of difficult years. The staff did an excellent job of holding the line on expenses, and several of our workshops were very well attended. Our two major conferences, however, drew fewer attendees than we expected; I'll have some more to say about this later.

We continue to struggle with the Journal production schedule; in general, the situation improved in 1992, and some excellent articles were published. We are trying to line up enough issues to provide some backlog so no individual author can tie up the production schedule. The circulation continues to increase, with distribution agreements with several overseas UNIX organizations and an increasing number of libraries subscribing.

1992 also saw a greatly increased number of \login: pages; more of the articles were technical, rather than involved with transient scheduling or administrative matters. We continue to offer the popular snitch reports as well, although it is beginning to look like only Gary Larson or Charles Addams could do justice to the Standards situation today. Rob Kolstad has demonstrated the successful editor's ability to arm-twist authors until they agree to write, and then wallop them with Email until they deliver; thanks, Rob, for doing a thankless job so well.

We had a board election in 1992, at which yours truly was elected president. As has been the tendency in past years, we had over half the previous board continuing to serve. This continuity makes life much easier on the staff and both the new and old board members.

Probably the biggest event of 1992 was the formation of SAGE. The Systems Administration community has become one of the most vital subgroups in our organization; our LISA conferences have grown each year, and are a vital and effective contribution to the profession. SAGE is a natural outgrowth of this focus, since there are still few places in the industry where systems administration issues can get a nonpartisan technical airing. SAGE now has 600 members, a newly elected board of directors, a section in \login:, and a host of working groups getting organized to tackle specific issues in systems administration.

Outside of USENIX, this has been a turbulent year. The European UNIX organization (EurOpen) held a conference last fall that was less than successful, and led to a sweeping reorganization, with the individual countries' organizations becoming more independent. We have a number of agreements, some formal, some informal, with European and Australian organizations; these cover everything from newsletter contents and journal distribution to guest representatives attending each others' meetings. We will watch the situation in Europe carefully and will probably engage in more cooperation with individual groups in 1993.

On the business front, UNIX has been under attack from a variety of sources, primarily by the nonexistent Windows NT. Luckily, the UNIX vendors have their own nonexistent products with which to answer the threat. Meanwhile, we had very successful workshops on microkernels and Mach, and a keynote speech in San Diego on PenPoint. Most likely, some of these airballs will start to hit the ground in 1993, but many of our members are already working on these "new" systems' replacements.

Closer to home, UniForum has a new Executive Director, and a board that seems committed to encouraging cooperation with USENIX. We will almost certainly be doing some joint projects in the standards and educational areas, and are discussing additional areas of cooperation.

The one thing on the horizon that is less than rosy is the attendance at our two general conferences. The attendance at these conferences is falling even as the workshop attendance is growing rapidly. There are a lot of theories—the recession, people only getting one conference a year, a diminishing number of systems programmers as UNIX gets more mature, locations that are not in technology centers, etc. The fact is, people continue to report that they like to attend the general conferences and find them useful, although most of the workshops rate higher in both categories. There are a lot of workshops we want to add, from mobile computing to application development. Perhaps we should just roll with the punches here, expect that general conferences will be a bit smaller than they have been, and...
budget accordingly. Perhaps we should go to one general conference a year, and add more workshops. We will be discussing this issue intensively in 1993 and would appreciate your input.

I would like to close with a special word of thanks to the board members, program committee members, and other volunteers who give so many hours to USENIX, and especially to our staff,

whose commitment to the membership and what we stand for goes way beyond considering it “just a job.” And to encourage readers to get involved — send in ideas, papers, articles, suggestions, and offers to help. You will meet and work with some very interesting people!

Board Meeting Summary

by Ellie Young, Executive Director

Below is a summary of the actions taken at the regular quarterly meeting of the USENIX Board of Directors which convened in San Diego, CA on January 24, 1993.

Attendance:

USENIX Board of Directors: Rick Adams, Eric Allman, Tom Christiansen, Lori Grob, Steve Johnson, Kirk McKusick, Evi Nemeth, Mike O’Dell, Barry Shein

USENIX Staff: Ellie Young, Judy DesHarnais, Dan Klein, Diane DeMartini

Others: Greg Rose, Mick Farmer, Jeff Haemer, Peter Collinson, Elizabeth Zwicky, Rob Kolstad, Dan Geer

Conferences

Winter ‘93 Conference. DesHarnais reported that attendance was less than projected and final attendance would be about 1,200. Kolstad reported that the conference convener model worked well, and a new “The Guru Is In” session was added.

Everyone agreed that this conference had a great program with something for everyone. A discussion about conference attendance ensued. It was suggested that a report on the total number of person days at all USENIX events in the past few years be generated to see what the effect the workshops may be having on the general conferences. It was also suggested that we might do a survey to find out why people aren’t coming back to the general conferences. The board was asked to give suggestions for new tutorial topics to Klein. Johnson presented a gift to Kolstad from the Board and staff in appreciation of his services as conference convener for San Diego.

Applications Development Symposium. After some discussion concerning the reasons for the low number of submissions and our subsequent postponement of this event, it was decided that we do want to have this symposium, and Kolstad and Young were asked to come up with a plan.

SANS II. It was suggested that Kolstad, Christiansen, and Zwicky produce a report for our membership that explains the differences between LISA and SANS conferences. [See page 20 for this report - Ed.]

Mobile Computing Symposium. Geer reported that Bob Metcalfe might give the keynote address, and he encouraged the board to get people to submit papers, as well as provide any leads about tabletop demos.

SAGE

Zwicky reported that SAGE had elected a new board. Johnson asked that she convey the USENIX board’s congratulations to them. She reported that SAGE had almost 600 members, and that although the working groups are not very active, they have been contributing material. It was suggested that USENIX/SAGE have a booth at UniForum, and that we have proceedings and USENIX/SAGE information at the SANS conference.

Local Technical Groups (LTG) Document & International Affiliate Groups

Johnson reported on the committee’s efforts at drafting a document, and that the rationale for LTGs was that they focus on social and technical issues from a local standpoint. Zwicky said there were several different types of established locals, with some more structured than others, e.g., BAYLISA, BACKBAY LISA, Motorola, and two separate international groups. After more discus-
sion, it was agreed that we should invite SAGE to make a couple of LTGs, and also collect mid and low-level details regarding the document, and run the final draft by our attorney.

Rose explained that he, along with Hal Miller and Zwicky, had drawn up a draft agreement which would allow SAGE-Australia to affiliate with SAGE in order to exchange information, i.e., accreditation, job descriptions, etc.

Policies

Adams’ proposal to replace section 6.2.2. of the Policies document concerning the Reserve Funds, with the establishment of an Endowment Fund (in which surplus monies that are not needed to finance the Association’s ongoing activities are invested with long-term growth and stability in mind) was passed.

Update on UniForum

Johnson reported on his recent meeting with their new Executive Director and Young, and that they would like to cooperate in several areas such as co-funding the standards report editor activities, doing joint membership promotions, cosponsoring educational seminars and colocating an event with their Spring Conference/Exhibition. It was agreed that there are a number of areas for cooperation, and we would consider some proposals, with standards being the first.

EurOpen

Farmer, as a member of their newly elected interim executive board, reported that they were considering a new model for a less expensive EurOpen, without an executive director, as follows:

National groups will pay a flat fee to EurOpen for membership. EurOpen is now a federation of national groups, and not the members of those national groups.

EurOpen will offer a minimal set of services to the national groups based on this fee, e.g., a news sheet containing information about the national groups’ activities.

EurOpen ceases to organize conferences on its own, but works in conjunction with a national group to organize a joint event.

Additional services may be purchased from EurOpen, e.g., a newsletter.

Their governing board will be considering proposals at their meeting in May, and will vote in a new executive board.

Conference Office IP Link

Adams reported that Alternet could provide a dedicated 56K line at below cost, which would mean a substantial savings for service. It was agreed that since the service is donated and we would be paying Pacific Bell directly for line charges, there is no conflict of interest, and we should accept this donation of service.

International Outreach

Grob and Shein formed a committee to look into making proposals for doing more outreach internationally.

Other Business

Adams pointed out that 32% of our membership revenue is foreign. Johnson suggested that we might want to offer a lifetime membership option.

Next Meeting

It was decided to hold the next meeting March 20, in San Francisco, alongside UniForum, and Young would try to arrange a social event with the UniForum Board as well.
# USENIX Association Financial Statements

## STATEMENT OF REVENUE AND EXPENSES

### AND CHANGES IN FUND BALANCE

For the Years Ending November 30, 1992 & 1991

### STATEMENT OF CASH FLOWS

For the Years Ending November 30, 1992 & 1991

### REVENUE

<table>
<thead>
<tr>
<th></th>
<th>1992</th>
<th>1991</th>
</tr>
</thead>
<tbody>
<tr>
<td>Membership Dues</td>
<td>$328,354</td>
<td>$303,165</td>
</tr>
<tr>
<td>Product</td>
<td>163,319</td>
<td>144,917</td>
</tr>
<tr>
<td>Conferences</td>
<td>2,046,205</td>
<td>1,661,835</td>
</tr>
<tr>
<td>SAGE</td>
<td>13,625</td>
<td>0</td>
</tr>
<tr>
<td>Interest</td>
<td>48,993</td>
<td>69,405</td>
</tr>
<tr>
<td>Other</td>
<td>3,320</td>
<td>4,078</td>
</tr>
<tr>
<td><strong>Total Revenue</strong></td>
<td><strong>$2,603,816</strong></td>
<td><strong>$2,183,400</strong></td>
</tr>
</tbody>
</table>

### EXPENSES

<table>
<thead>
<tr>
<th></th>
<th>1992</th>
<th>1991</th>
</tr>
</thead>
<tbody>
<tr>
<td>Membership Services/General Admin.</td>
<td>$500,444</td>
<td>$533,827</td>
</tr>
<tr>
<td>Conference</td>
<td>1,395,046</td>
<td>1,344,462</td>
</tr>
<tr>
<td>SAGE Expenses</td>
<td>5,314</td>
<td>0</td>
</tr>
<tr>
<td>Newsletter &amp; Journal</td>
<td>131,870</td>
<td>113,744</td>
</tr>
<tr>
<td>Products</td>
<td>67,949</td>
<td>69,094</td>
</tr>
<tr>
<td>Projects</td>
<td>92,414</td>
<td>100,956</td>
</tr>
<tr>
<td>Depreciation</td>
<td>20,184</td>
<td>34,814</td>
</tr>
<tr>
<td><strong>Total Expenses</strong></td>
<td><strong>$2,213,221</strong></td>
<td><strong>$2,196,897</strong></td>
</tr>
</tbody>
</table>

### Excess Revenue Over Expenses

<table>
<thead>
<tr>
<th>1992</th>
<th>1991</th>
</tr>
</thead>
<tbody>
<tr>
<td>$390,595</td>
<td>$(13,497)</td>
</tr>
</tbody>
</table>

### Fund Balance Beginning of Year

<table>
<thead>
<tr>
<th>1992</th>
<th>1991</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1,193,699</td>
<td>$1,207,196</td>
</tr>
</tbody>
</table>

### Fund Balance End of Year

<table>
<thead>
<tr>
<th>1992</th>
<th>1991</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1,584,294</td>
<td>$1,193,699</td>
</tr>
</tbody>
</table>

### BALANCE SHEET

As of November 30, 1992 & 1991

### ASSETS

<table>
<thead>
<tr>
<th></th>
<th>1992</th>
<th>1991</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cash</td>
<td>$597,002</td>
<td>$938,741</td>
</tr>
<tr>
<td>Accounts Receivable</td>
<td>10,033</td>
<td>12,337</td>
</tr>
<tr>
<td>Prepaid Expenses</td>
<td>75,730</td>
<td>94,443</td>
</tr>
<tr>
<td>Inventory</td>
<td>44,749</td>
<td>45,362</td>
</tr>
<tr>
<td><strong>Total Current Assets</strong></td>
<td><strong>$727,514</strong></td>
<td><strong>$1,090,883</strong></td>
</tr>
</tbody>
</table>

### Fixed Assets

<table>
<thead>
<tr>
<th></th>
<th>1992</th>
<th>1991</th>
</tr>
</thead>
<tbody>
<tr>
<td>Securities</td>
<td>$885,706</td>
<td>$191,601</td>
</tr>
<tr>
<td>Net Property &amp; Equipment</td>
<td>31,367</td>
<td>51,551</td>
</tr>
<tr>
<td><strong>Total Fixed Assets</strong></td>
<td><strong>$917,073</strong></td>
<td><strong>$243,152</strong></td>
</tr>
</tbody>
</table>

### Total Assets

<table>
<thead>
<tr>
<th>1992</th>
<th>1991</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1,644,587</td>
<td>$1,334,035</td>
</tr>
</tbody>
</table>

### LIABILITIES & FUND BALANCE

### Current Liabilities

<table>
<thead>
<tr>
<th></th>
<th>1992</th>
<th>1991</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accrued Expenses</td>
<td>$11,408</td>
<td>$84,026</td>
</tr>
<tr>
<td>Deferred Revenue</td>
<td>48,885</td>
<td>56,310</td>
</tr>
<tr>
<td><strong>Total Liabilities</strong></td>
<td><strong>$60,293</strong></td>
<td><strong>$140,336</strong></td>
</tr>
</tbody>
</table>

### Fund Balance

<table>
<thead>
<tr>
<th>1992</th>
<th>1991</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1,584,294</td>
<td>$1,193,699</td>
</tr>
</tbody>
</table>

### Total Liabilities & Fund Balance

<table>
<thead>
<tr>
<th>1992</th>
<th>1991</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1,644,587</td>
<td>$1,334,035</td>
</tr>
</tbody>
</table>

---

6 login: March/April 1993
Are you ever curious to know how your USENIX membership dues are spent? Here are a few charts that might help. The first shows sources of membership dues income, which totaled $328,000 in 1992. The second chart shows how that money was parcelled out. Note that the Conference office does not receive any monies from membership; it is totally funded by income generated from Conferences and Symposia.

You should also know that only about one half of Executive office payroll and general expenses are covered by membership dues, the balance is funded by income from Conferences & Symposia and from publications sales. The third chart shows how the executive office spends its money. The 'Other' category includes items such as taxes and licenses, systems support, and expenses associated with the production and processing of membership renewal notices, information and materials.

### Membership Income Sources

- **Individuals**: 75%
- **Students**: 13%
- **Educational**: 11%
- **Corporate**: 1%

---

March/April 1993
Where Do Your Membership Dues Go?

Executive Office Expenses

Rent
Board Meetings
Depreciation
Accounting
Board & Staff Travel
Office Equipment & Supplies
Postage
Telephone
Insurance
Credit Card Expense
Database Consultants & Temp. Help
Utilities
Legal
Elections
Other
Summary of Technical Sessions on File Systems and IO

by John J. Wallace
<jwallace@epoch.com>

Introduction

Of the 45 technical papers featured at the Winter '93 Conference, this report summarizes seven from the File Systems and IO technical sessions. I also discuss related sessions in "More Reading." My goal is to highlight each paper, focusing on motivations and conclusions. Refer to the proceedings for more complete information.

Of the seven papers, three dealt with hierarchical file systems and two with management of removable media. Hence, I have roughly categorized the papers as follows: innovative file systems, hierarchical file management, and removable media management.

Innovative File Systems

Michael Olsen from University of California at Berkeley <mao@cs.berkeley.edu> described "The Design and Implementation of the Inversion File System." Unlike some database systems that are built on top of a file system, Inversion is a file system built on top of the POSTGRES database (hence the name). Building Inversion on POSTGRES offered a number of advantages:

• Inversion recovers fast via POSTGRES's rollback mechanisms.

• Since POSTGRES supports a no-overwrite stage model, a user can examine the state of a file at any point in history, not just at predefined backup intervals.

• POSTGRES (and, hence, Inversion) uses a variety of tertiary storage devices, including a WORM jukebox.

• Users can define file types and functions that can be performed on typed files.

• Users can perform ad hoc queries on the file system, file data, and file metadata, avoiding special tools like find.

• Inversion supports an explicit transaction model.

A good deal of the paper discusses implementation: directories, file attributes, and data are all stored as relations. The paper also discusses performance, which is "reasonable:" 30 to 80 percent as fast as a native file system over NFS.

Paul Eggert <eggert@twinsun.com> from Twin Sun discussed "File Systems in User Space." These aren't file systems, per se, but a mechanism to intercept certain system calls (via shared libraries) and interpret path names in special ways. Any symbolic link can reference an executable that is interpreted at run time when determining the ultimate path or content of a file.

Paul defines an "Intentional" file system as one where the path name must be interpreted to provide the intention. A simple example, from the paper, produces a tar image on demand:

```bash
ln -s '(tar cf - *.ch)' dist.tar
```

Now, any reference to the "intentional" file `dist.tar` would actually run `tar` to produce a tar image, but never (in concept) store the tar image persistently.

This simple concept gets quite confusing as Paul describes ways to intentionalize entire directory trees. Some in the audience felt this was little more than a collection of cute hacks, and Paul could not convince them otherwise.

Hierarchical File Management

Neil Webber <nvw@epoch.com> from Epoch Systems presented an architecture for "Operating System Support for Portable Filesystem Extensions." Neil is trying to solve the problem of adding file system services "on top of" the native file system in a portable, well-understood manner. Such services might include compression, encryption, or hierarchical storage management.

Neil first describes the state of the practice: adding file system services within the kernel is fundamentally unportable and expensive. Everyone implemented the Virtual File System differently; and underlying architectural differences from one kernel to another force file system extensions to be reimplemented for every UNIX variant.

Neil then proposes an architecture that allows code outside the kernel to affect filesystem operations. He embeds "detection mechanism" within the kernel file system that communicates with a user-level policy daemon that determines what to do with individual requests (read, write, etc.). By embedding the detection within the kernel, but leaving the policy at user-level, OS vendors allow third parties to add file system...
extensions in a portable, well-behaved manner. Neil is currently rallying OS vendors around this implementation.

Ethan Miller of the University of California at Berkeley (<elm@cs.berkeley.edu>) described “An Analysis of File Migration in a UNIX Supercomputing Environment.” He analyzed file migration and reference patterns at the supercomputer center at the National Center for Atmospheric Research.

Ethan collected migration statistics on the 23 terabyte system for 24 months. Although the paper contains extensive analysis of the trace, some tidbits follow. These mostly affirm that hierarchical file systems work.

- Hierarchies work: magnetic disk provides data more quickly than a tape robot, which provides data more quickly than hand-mounted tape.
- People cause migration faults: read traffic is correlated to human work hours.
- There's a lot of useless data: 50% of all files were never read; and another 25% were read only once.
- Read caching works: when a file was read more than once, the additional accesses came soon after the first. (For 70% of the cases, additional accesses came in less than one day.)
- Migration systems should treat small files specially: since 40% of reads went to “small” files of less than a megabyte, the migration system must read these files quickly. Since they consume less than 1% of the total storage, small files could be economically migrated to fast, more expensive tertiary storage (say magnetic or optical disk).

John Kohl (<jkohl@cs.berkeley.edu>) of the University of California at Berkeley described “Highlight: Using a Log-structured File System for Tertiary Storage Management.” John (and company) saw synergy between LFS’s sequential write pattern and the performance characteristics of tertiary storage, such as tape, where sequential access performs much better than random.

Their implementation uses the standard LFS segment layout on the magnetic disk cache, but a modified cleaner moves unaccessed data to segments on tertiary storage. The paper discusses a variety of migration policies but gives few details on the one chosen. Other changes to LFS include:

- A read-only magnetic disk cache for segments read from tertiary storage
- A pseudo-disk driver that stripes different devices (magnetic disk, tertiary storage) and creates a uniform block address space
- Processes to access tertiary storage

The end of the paper discusses performance.

Removable Media Management

In the first of two papers on this subject, Howard Alt (<halt@central.sun.com>) of SunSoft presented work he’s done on “Removable Media in Solaris.” This work has two main applications: ease of use for the desktop, and media management within complex system software.

Howard wants to make removable media like floppies and CDs easier to use. This technology allows desk-top users to insert, eject, and access removable media by logical names, avoiding the need to become privileged or use awkward (to some) mount options and device names. However, complex system applications like backup also need to label and manage removable media.

Howard's architecture includes a user-level NFS server that handles the media namespace, and a kernel “mux driver” that manages access to physical drives. As media are inserted and ejected, their labeled names appear in the name space managed by the NFS server. However, access to the media traps directly to the special file system, short-circuiting the NFS server. The mux driver maps access to the logical media to a physical device. If the media is not present in a device when accessed, the access pends while an operator (human or robot) locates and mounts the media. Shared libraries allow third parties to add additional devices, robots, labels, and media types. See Howard for details.

Christopher Calabrese (<cjc@ulysses.att.com>) of Bell Labs then presented “An Advanced Tape Cataloging System for UNIX Systems.” Chris discusses the requirements for managing a large tape archive (up to 100,000 tapes) and his implementation. Chris did not embed media support into the kernel, but instead, provides UNIX-like commands (tcd, trm, lls, etc.) and APIs (topen, tread, etc.) that applications use to access labeled tape. Further, there is no concept of drive multiplexing. Instead, the application binds (mounts) a labeled tape to a physical drive. The current system manages 12,000 tapes, with an additional 1000 added per month.

More Reading

Margo Seltzer (<margo@das.harvard.edu>) presented work on “An Implementation of a Log-Structured File System for UNIX” for BSD 4.4.
This was curiously included in the “Kernel Improvement” session, which I missed because of an overdue road trip to Point Loma. Nevertheless, the paper completely describes the motivation for the work, its implementation, and resulting performance.

Although based on the Sprite-LFS, this work aimed to provide a production-quality replacement for BSD’s traditional Fast File System (FFS). Performance measurements showed that LFS performed faster than FFS when measuring maximum read and write bandwidth (because of FFS’s block interleave policy). But LFS’s maximum write bandwidth was not as fast as EFS, an extended version of FFS that does contiguous block allocation and large disk I/Os. This is largely due to LFS’s tendency to batch writes to disk, whereas EFS can more favorably overlap compute and disk IO. A multi-user software development benchmark shows the three file systems (LFS, FFS, EFS) performing similarly, with no more than (roughly) a 10% spread between the three.

Other papers covered similar topics: faster AFS, NFS file caching, kernel support for OLTP, etc. Peter Honeyman also led a panel highlighting last year’s USENIX File Systems Workshop. I refer you to the respective proceedings.

Invited Talks Report from San Diego

by Peg Schafer
<peg@bbn.com>

I love going to USENIX; I learn so much. One of the most enjoyable features this year was the invited talk track. Presented is a sampling of the talks:

Ever want to play with an amino acid? You could at Tom Ferrin’s presentation on molecular visualization, utilizing MidasPlus, the molecular modeling system developed at the Computer Graphics Lab, UCSF. Wonderful! From a short tutorial on X-ray crystallography and how molecules bind, we were introduced to the world of interactive 3D graphics. Molecular modeling requires human knowledge, numerical calculations, computer graphics, and AI. Using a SGI Indigo, capable of transforming one million 3D vectors per second, attendees could glide around a DNA molecule and gain a new understanding of modern chemistry!

Blazon, the language of heraldry, proved to be most intriguing. Dan Klein, in his presentation “From Blazon to Postscript,” led us to examine several “language” systems which describe something generally graphical. Notable examples are dance, musical, and chemical notations. Language is everywhere for Dan – in weaving, the descriptions of tartans and brands. (Yep, the type of brands cowboys still use on cows in Montana.) Eventually we got around to computer languages developed for graphical functions. Dan bemoans the state of drawing languages today and believes it will be a long time before we have a language as powerful as Blazon. There is some merit in this argument.

A view of UNIX history from “Down Under” was presented by our favorite Australian, Greg Rose. We learned the truth about UNIX now – it was built to play a space war game! We always suspected... Actually a very sound historical overview filled in some gaps of understanding for me. I love the empty black boxes in the “What version was that?” slide. Now that I know where UNIX has been, do I know where UNIX is going? Ha!

Internet MultiMedia Mail – “Everyone gots to have it” but why don’t we see it in use everywhere? Nathaniel Borenstein is actually doing some real things to make it happen. In his presentation “MIME & Metamail: Moving Multimedia Mail into the Mainstream?” he describes a proposed format which would allow multimedia mail to go beyond “the islands of X.400.” Nathaniel points out that extensions to RFC 822 are necessary to transition from the huge installed base of old standard mailers to multimedia mail. MIME (Multipurpose Internet Mail Extensions) offers a simple, standardized way to represent and encode a wide variety of media types. All of this is to further more interesting functionality such as multimedia journals delivered to your desktop. I am looking forward to it!

I generally think of myself as a culturally aware and sensitive person. Not so, when it comes to computing considerations! Jeff Haemer’s presentation on the really tricky issue of internationalization revealed a can of worms I never dreamed existed. The ideal is to have code be portable across cultural and linguistic boundaries simply by setting a few defines. Everything changes. For example printf – the word order of nouns and adjectives is very language dependent. Our default assumptions about such standards as regular expressions and date formats have flown out the window. If you are very bored with standard computing problems – go into internationalization!

This is just a sampling of the invited talks. I found them useful because they opened interesting and informative areas in computer science that I have never considered.
Winter ‘93 Conference Report
by Christopher (C. J.) Rath
<crrath@bnr.ca>

Introduction


Conference Opening Remarks

The first USENIX lifetime achievement award was given to CSRG. The award, called “The Flame,” is a red glass sculpture depicting a stylized flame. Recipients of the award are called “Keepers of the Flame.” [See page 17 for award—Ed.]

Keynote Speech; Robert Carr, Go Corp.

Two-thirds of the U.S. workforce do not directly interact with computers as part of their work. Along these lines, it is interesting to note that even computer professionals become disenfranchised when they leave their desks. For example, only a small minority of the USENIX attendees are taking notes on a computer during this speech.

Go Corp.’s conviction is that wherever pencil and paper are in use, a potential exists for use of a computer. However, the current keyboard paradigm will not work in many of these traditional situations for both technical and social reasons.

FAX, Email, and cellular phone technologies have all experienced exponential growth in recent years. This makes them prime candidates for integration with emerging mobile computing technologies.

Go Corp.’s innovations in mobile computing have resulted in what the press has called “Pen Computing.” However, from Go Corp.’s perspective, it is not the pen-on-glass technology which is the key element of Pen Computing; rather, it is the enfranchisement of people that is key. This is due primarily to the portability and usability of the pen computing environment.

One of the keys to the success of mobile computing will be social acceptance. If it’s not socially acceptable to pull out your computer and use it, then the majority of users will shun the technology. Pen and paper is already a socially acceptable paradigm, with “any time/anywhere” being a key enabler of mobile computing technology.

The concept of file systems is one of the most difficult to teach. This paradigm and many others are prominent barriers to the increased use of computing and new paradigms, and methods of teaching must be found in order to increase use and acceptance of mobile computing.

Mobile devices must be very network robust. That is, connecting to and disconnecting from one or more networks must be handled as a normal daily occurrence, not an exception. The mobile device must also manage I/O intelligently – deferring outgoing data until a network connection is available, and pulling down incoming data at regular intervals.

Some interesting statistics which indicate the potential market for a properly engineered and implemented mobile computer:

• Desktop PC workers comprise 20% of the U.S. workforce.

• Mobile workers comprise 40% of the U.S. workforce; 20% mobile in the office; 20% mobile in the field.

Robert Carr then gave a long demo of PenPoint, Go Corp.’s OS. PenPoint appears to be a very powerful and easy-to-use user interface, implemented in a very orthogonal fashion. The file system paradigm has been replaced with that of a book: pages, sections, chapters, table of contents, etc. The OCR technology embedded in PenPoint worked much better than I anticipated (on printed text, anyway).

Talk: HELLO WORLD, by Rob Pike

Rob Pike spoke about the internationalization of Plan 9, a research OS developed by Bell Labs.

The most difficult part of supporting non-7 bit character sets (i.e., 16 or 32 bit characters) is simply making the initial transition. Once your software can handle char not being a byte, then it no longer matters exactly which character set or encoding scheme is used.

A good example is the malloc-ing of space for a buffer. Traditional C programmers would have written malloc(BUF SIZ); now you must write malloc(BUF SIZ * sizeof(char)).

Most of the changes required to support the new character sets and encoding schemes can be implemented unobtrusively. For example, the malloc() in the previous paragraph. If we begin to make these sorts of changes now then there will be less work to do later.

Talk: ES – The Extensible Shell

This shell has been implemented in a very simple and elegant way, which differs greatly from existing shells. The biggest difference is that es allows the user to easily subclass the built-in commands
and operators, using es's own commands.

This shell looks like it might be worth closer examination. The source for es is available for ftp. Email <haahr@adobe.com or byron@netapp.com> for further information.

Talk: Internationalization

This talk documented just how involved a process it is to fully internationalize an application. To do the job properly you must do more than just handle non-ASCII character sets. Internationalization of an application must take into account cultural as well as language issues.

Talk: TCP/IP Network Administration

Hostname selection is covered by RFC 1178. People and organizations choosing system names should get a copy of this RFC and follow it. DNS is much simpler to configure and maintain than straight UUCP. It appears to me that there may be immediate and long term benefits to converting to DNS from UUCP, even if one doesn’t have a direct Internet connection.

Talk: WAFE

WAFE is a replacement for TCL’s TK library. TK implements Motif-like widgets, while WAFE uses real Athena or Motif widgets. While TK only works with TCL, WAFE has been designed to work with any application. WAFE has been used by developers along with C and FORTRAN programs, Perl scripts, SQL-Plus, and other applications. WAFE provides an easy way to create an X UI for any program.

A UI Builder application is provided with WAFE; it generates either WAFE or Perl scripts. WAFE can be ftp’ed from ftp.wu-wien.ac.at: pub/srclXllwafe/.

Talk: MIME - Multimedia Email

The MIME specification was created with the help of the X.400 people. In fact, MIME can be used as an X.400 transport layer. The MIME specification appears to be very robust, yet not overly complex.

Talk: Object Databases

Two types of object database technology are currently being developed and marketed: extended relational and object oriented (OO).

Since an object consists of Data plus Procedures, there arises a problem of procedure specification:

- What language should procedures be written in?
- Where can/should procedures execute, the client or the server?

These same issues arise with respect to types (i.e., classes).

The extended relational databases side-step (to a great extent) the language and type-evaluation issues. This is because they are not tightly bound to either the application or the language. On the other hand, object oriented databases are generally tightly bound to the application and language: Procedure and Type specification and processing are shared. This means:

- Procedures may execute in either the client or the server.
- Objects obtain their persistence via inheritance from a DB base-class. The application may then derive new classes which are immediately usable by the DB.
- Because the application does not manage object persistence, the application is simpler to write: Write as though you are managing generic objects, and they will simply be persistent. That is, the database is effectively transparent to the application.

Object oriented databases come in two implementations:

- Memory-mapped, pointer-based implementations which overload the pointer de-reference operator so that they can page in objects from disk, directly into virtual memory. Objects are brought in from disk a memory-page at a time.
- Object-handle implementations which manage objects in a much more traditional manner. Objects are brought in from disk one at a time.

Memory-mapped implementations very quickly exhaust virtual memory, and so they do not scale up very well. Object-handle implementations run a little slower; however, virtual memory exhaustion is not an issue.

BOF Session: Administration Of NeXT Systems

Twenty people attended the BOF. The NeXTSTEP 3.0 release was discussed in some detail. While some minor problems were reported, only one of the 20 people advocated not upgrading to 3.0. The largest installation represented was SwissBank. They have have a worldwide WAN with 800 NeXTs connected. [NeXT has since ceased production of its hardware – Ed.]

BOF Session: BSDI

BSDI produces a BSD 4.3 derived version of UNIX which comes with source code that is AT&T free. It only runs on 386/486 PCs. BSDI is currently shipping release 0.9.4 on CD-ROM. Over 650 cop-
ies have been sold, and half a dozen distributors exist outside the U.S.

Their DOS emulator currently only supports 8086 code. SCO UNIX binary compatibility is under development and will be ready by the summer. While BSDI is not fully POSIX compliant, they are working on it and expect to begin conformance testing later this year.

A binary only release of BSDI may be available shortly after release 1.0 goes out. A Sun Sparc port of BSDI will begin once release 1.0 is out. An AT Ultra VGA card, using BSDI’s X Windows, server clocks 85,000 Xstones.

Vendor Display Highlights

PageSat Inc.:

They offer a satellite delivered USENET news feed. This is a full news feed, over 50 MB/day, including most major regional groups from around the world.

Users with access to Email can request that files be sent to them via the satellite link. This means that if you are now paying connect time to ftp large files from other Internet sites, you may be able to reduce your connect charges by using this email service.

I believe that many companies could benefit from PageSat’s service.

Get info from djcl@pagesat.com.

ClariNet:

They are an electronic news & information supplier, offering:

• UPI wire service
• An electronic newspaper published in USENET format
• “Library Of Tomorrow,” an electronic bookstore where fiction and nonfiction books can be purchased for about $5 per novel-length work.

Contact info@clarinet.com for more information.

Walnut Creek CD-ROM:

Offering a selection of CD-ROM titles starting at $25. Call (510) 947-5996 for info.

O’Reilly Books:

Their latest offering is UNIX Power Tools. This looks like an excellent reference book for both new and experienced users. It’s 1100 pages of UNIX tips. The programs and utilities referred to are supplied on CD-ROM (comes with the book), along with Sun3, Sun4, HP700, RS6000, DEC, and SCO UNIX binaries.

UUPSI:

Offering dial-up Internet service (just like ALTER-NET). Get information from info@psi.com.

Mortise Kern Systems (MKS):

The latest release of the MKS Toolkit includes full UUCP, via a TSR. Get information from inquiry@mks.com.

Free Software Foundation (FSF):

Their latest printed manuals are bound using the new lay-flat technology. The printed manuals make an excellent investment for regular users of FSF tools, especially with the new binding. Get more information from gnu@prep.ai.mit.edu.

Best Paper and Presentation Awards

The following awards were presented at the San Diego Conference:

The Best Student Paper Award went to Steve McCanne and Van Jacobson of Lawrence Berkeley Laboratory for “The BSD Packet Filter: A New Architecture for User-level Packet Capture.” The Best Paper Award went to Wayne Christopher for “The Nachos Instructional Operating System.” Co-authors are Steven J. Procter and Thomas E. Anderson.

A Best Presentation Award was given to Margot Seltzer and Stephen A. Uhler. Seltzer received a cash award for presenting the paper “An Implementation of a Log-Structured File System.” Her co-authors are Keith Bostic, M. Kirk McKusick, and Carl Staelin. Stephen A. Uhler, Bellcore, presented “Phonestation, Moving the Telephone Onto the Virtual Desktop.”
This talk describes the threads work in progress at Data General for its next major release of DG/UX, a completely re-engineered UNIX SMP kernel with nine years of maturity on Eclipse and AViiON systems. DG's implementation of threads is novel in that it is kernel-based, yet extremely efficient.

The presentation concentrates on the performance advantages of kernel-based threads for commercial applications and on the use of fast kernel traps on RISC processors to achieve high performance in a kernel-based implementation. Finally, it backs up these claims with real performance measurements taken from a production-quality kernel; for example, on a 25MHz 88100, DG/UX can deliver a cached thread create in 4.5 usec and a thread-to-thread context switch in 13 usec.

A Cryptographic File System
Matt Blaze
AT&T Bell Laboratories
Holmdel, NJ 07733
<mab@research.att.com>

As computing systems (especially distributed ones) grow in size, issues of data security and privacy become increasingly complex. Cryptographic techniques can help ensure that data are not read by unauthorized persons, but most encryption software requires either that special purpose application software be used or that the user manually encipher and decipher files as needed.

The Cryptographic File System (CFS) makes it easier to take advantage, in a secure manner, of file system services (storage, backup, etc.) on potentially insecure servers and networks.

CFS provides a transparent UNIX file system interface to directory hierarchies which are automatically DES encrypted with user-specified keys. Users “attach” an encrypted directory by providing a key, the name of a directory where the encrypted files are to be stored, and the name of a cryptographic “mount point” to be created under /crypt. Directories under /crypt are accessible with all standard system calls and tools to the users who created them. The underlying encrypted files (with encrypted names) can reside on any accessible file system (including remote file systems such as NFS); routine system administration tasks, such as file backup and restore, can be performed on the encrypted directories in the ordinary manner without knowledge of the key. When run on a client workstation, CFS ensures that cleartext is never stored on a disk or transmitted over a network. CFS uses a standard portable NFS client interface and has been implemented for a variety of UNIX platforms.

Simon: A Database System for UNIX Systems Administration
Jon Finke, Senior Network Systems Engineer
Information Technology Services
Rensselaer Polytechnic Institute
110 8th Street Troy NY, 12180
<finkej@rpi.edu>

RPI has developed (and continues to expand) a database driven system administration system known as Simon.

Simon takes snapshots from the Registrar and Payroll, and manages the creation and expiration of essential UNIX accounts for our system here. It also provides support for accounting and billing for AFS disk usage, and will very shortly be used to manage the host tables (DNS RR files) and printer configuration (/etc/printcap) on our systems. Eventually, it will manage the system configuration on several hundred UNIX workstations.

Internet Resource Discovery: Integrating Prospero and Gopher
Steven Augart
University of Southern California
Information Sciences Institute

This talk presented a brief outline of Prospero’s role in Internet resource discovery and our planned integration of Prospero with Gopher. Prospero is a per-user customizable directory service based upon the Virtual System Model.

Gopher is a service that allows the user to browse through menus configured by Gopher service providers. We plan to integrate these approaches to resource discovery by making Gopher menus available through Prospero and by writing a Prospero-based Gopher-like browser that allows per-user customization of all Prospero directories, including ones formerly only accessible through Gopher.

For further information about Prospero, send E-mail to <info-prospero@isi.edu>.
First Annual Lifetime Achievement Award

At the Winter '93 Conference in San Diego, CA, the Association inaugurated its Lifetime Achievement Award by honoring the Computer Systems Research Group (CSRG) of the University of California at Berkeley. This award will be presented annually to honor intellectual achievement and unparalleled service to the UNIX community, especially for work which has not been honored elsewhere. The 1993 award honors 180 participants and supporters who have contributed to the CSRG Berkeley UNIX project over the years. The award especially recognizes seven individuals as "Keepers of the Flame" for their leadership in the CSRG effort: Ozalp Babaoglu, Keith Bostic, William N. Joy, Michael J. Karels, Samuel J. Leffler, Marshall Kirk McKusick, and Keith Sklower. Each year, those individuals designated as Keepers of the Flame will receive an original glass sculpture titled The Flame, which was designed by Lewis Olson of Noslo Glass Studios, Corning, New York.

The CSRG has been responsible for producing the Berkeley UNIX (BSD) releases. This effort was initially funded by DARPA to serve as a foundation for DARPA-sponsored research. Along the way, the BSD UNIX releases have formed the basis of many commercial UNIX products. The wide availability of the BSD networking code is often cited as a major contributor to the success of the TCP/IP protocol and the resulting explosive growth of the Internet. The large number of individuals and organizations recognized by this award is testament to the pervasiveness of this effort throughout the research community and the degree to which the work originally centered at Berkeley has become a community project.

The Association plans to bestow the Lifetime Achievement Award annually at its Winter Technical Conference.
The USENIX Association presents the 1993 Lifetime Achievement Award to the Computer Systems Research Group, University of California at Berkeley 1979-1993.

Presented to honor profound intellectual achievement and unparalleled service to our Community.

At the behest of CSRG principals we hereby recognize the following individuals and organizations as CSRG participants, contributors and supporters.

---

## Keepers of the Flame

Ozalp Babaoglu  
Keith Bostic  
William N. Joy  
Samuel J. Leffler  
Marshall Kirk McKusick  
Michael J. Karels  
Keith Sklower

---

## Supporting Organizations

- Carnegie-Mellon University Mach Project
- Computer Systems Engineering Group, Lawrence Berkeley Laboratory
- Department of Defense Advance Research Projects Agency (DARPA)
- The Hewlett-Packard Company
- The Institute of Electrical and Electronic Engineers, Inc.
- Project Athena and The Massachusetts Institute of Technology
- University of British Columbia

- Compaq Computer Corp.
- Cornell University
- Gray Research Inc.
- Digital Equipment Corp.
- The Free Software Foundation
- IBM Corporation
- MCI
- The Open Software Foundation
- Sun Microsystems Inc.
- UUNET Technologies Inc.
- University of Wisconsin

---

## Major Contributors

- Eric Allman
- Ken Arnold
- Jim Bloom
- Ralph Campbell
- James Clark
- Kevin Dunlap
- Robert Elz
- Robert Fabry
- Domenico Ferrari
- Tom Ferrin
- Jeff Foryst
- Akito Fujita
- Rick Gingell
- George Goble
- Susan L. Graham
- Rob Grunwitz
- Trent Hein
- Mike Hibler
- Van Jacobson
- Bill Jolitz
- Bob Kriddle
- Jim Kulp
- Jay Lepreau
- Zhishun Alex Liu
- Rob Gingell
- George Goble
- Susan L. Graham
- Rob Grunwitz
- Trent Hein
- Mike Hibler
- Van Jacobson
- Bill Jolitz
- Bob Kriddle
- Jim Kulp
- Jay Lepreau
- Zhishun Alex Liu
- Supporting Organizations
- Compaq Computer Corp.
- Cornell University
- Gray Research Inc.
- Digital Equipment Corp.
- The Free Software Foundation
- IBM Corporation
- MCI
- The Open Software Foundation
- Sun Microsystems Inc.
- UUNET Technologies Inc.
- University of Wisconsin

---

## Contributors

- Kim Leskeman
- Kevin Lew
- John Lions
- Cynthia Livingston
- Tim Long
- Chris Malabdy
- Steven McCanne
- Eamonn McManus
- Gregory Minshall
- Noah Morgan
- Scooter Morris
- Adam S. Moskovitz
- Keith Muller
- Mike Musser
- Tony Nardo
- Chris Newcomb
- Mirtam Arosa Nihart
- Landon Curt Noll
- Jim R. Oldroyd
- Mike Olson
- Joseph Orton
- Richard Osterbridge
- Vern Paxson
- Jonathon Payne
- Rob Gingell
- George Goble
- Susan L. Graham
- Rob Grunwitz
- Trent Hein
- Mike Hibler
- Van Jacobson
- Bill Jolitz
- Bob Kriddle
- Jim Kulp
- Jay Lepreau
- Zhishun Alex Liu
- Rick Macklem
- Arthur David Olson
- Jan-Simon Pendry
- Marshall Rose
- Pauline Schwartz
- Donn Seeley
- Bill Shannon
- Mark Teitelbaum
- Avadis Tevanian, Jr.
- Chris Torek
- Kazuma Utsushi
- Michael Wayne Young

---

March/April 1993
Software Installation on Large Systems
by Paul Anderson
Department of Computer Science
University of Edinburgh, U.K.
<paul@dcs.ed.ac.uk>

Introduction

The Software Installation Workshop at the 1992 LISA conference was organized to discuss some of the problems of installing and configuring third-party software in large heterogeneous installations. Most current third-party software packages provide their own installation procedures which often seem to be designed primarily for use on small, single-vendor systems. In large installations, these procedures can be totally inappropriate, causing considerable difficulties for system administrators; in extreme cases, it may even be impossible for a particular site to configure a package in any useful way at all, limiting the choice of available software and affecting purchasing decisions.

The workshop attempted to identify the main source of these difficulties and give some hints and recommendations for improving software installations so that they are more compatible with large systems.

Software Installation and Configuration

The POSIX draft P1003.7.2 proposes a standard software distribution layout and describes an installation process consisting of several stages: These stages cover the identification of files to be installed (selection), verification of certain prerequisites (analysis), and copying of the files onto the system (load). The proposal provides many welcome features, such as the ability to de-install a package and perform remote installations.

During the load phase, provision is made for vendor-supplied scripts to be executed with the intention that these scripts should configure the software package itself for the environment. Changes to the actual environment (such as modifications to system files) are expected to occur in a further stage (configuration), although details of this are explicitly excluded from the standard. These configuration issues which are not covered by the POSIX standard are precisely those areas that frequently cause difficulties when installing software packages onto large heterogeneous systems. The following sections describe some of the issues which were identified as significant problems at the workshop.

Location of Files

The single largest cause of difficulties with software installation is probably a lack of flexibility in configuring the pathnames used by the package. Different vendors use significantly different pathnames in their basic operating systems, and most large multivendor installations have their own standard locations for various types of files.

Current installation procedures typically specify some fixed absolute pathnames for some, or all, of the files used by the package. At best, this can involve extra work and confusion because the supplied pathnames do not conform to the normal site conventions. At worst, one or more of the pathnames will conflict with some other package or will be on an inappropriate filesystem. For example, /usr/local/lib is often mounted read-only, in which case it is not a suitable place for log files or other writable data. The POSIX standard provides an option for a package to be locatable—i.e., to specify a alternative root directory for the installation. This obviously avoids the worst of these problems, but it is only optional, and it is barely sufficient for large installations where it is most useful if different pathnames can be specified for different categories of files. For example:

• In a heterogeneous network, it is very useful to separate out those files which are architecture-dependent and those files which can be shared between architectures.

• Files which need to be writable must be separated from read-only files which are frequently replicated and stored on read-only filesystems.

• Files which need to be private to a particular workstation should be separated from files which would be common across the whole network.

• The directory from which the package will finally run is sometimes different from the directory into which it is initially installed. This is the case, for example, if a package is installed...
into /home/package but subsequently distrib-
uted to a different location on multiple servers
(typically using rdist). Installation programs
which use pwd to determine the current direc-
tory and subsequently configure this into the
software are particularly troublesome because
they usually do not work in conjunction with
the automounters which are common in large
NFS installations.

File Ownership and Security

Large installations are generally more sensitive to
file ownership and permission issues, due to the
increased scale, and the importance of security.
This means that it is important that files are not
installed with inappropriate usernames or pro-
tections (for example, as a result of assumptions
about the default umask at installation time). At
runtime, packages should also use setuid or setgid
to some normal user in preference to setuid root
wherever possible.

As with the choice of pathnames, a large installa-
tion is also more likely to suffer from conflicts
between user and group names if a package
insists on being installed as a particular user or
group (the same is true of user and group IDs). It
is usually quite reasonable for a package to be
installed with its own username, but it must be
possible to override the default name (and ID) if
required.

During the installation itself, system administra-
tors will be very reluctant to run processes as root
unless this is absolutely essential and their conse-
quences are well understood. A dummy installa-
tion option, which simply shows the actions that
a real installation would perform, was suggested
as a useful way of checking the consequences of
such a procedure. In most cases, however, the
bulk of the installation process should be capable
of running under normal user permissions. This
is sufficient even for installation of files in public
areas (such as /usr/local/bin) with careful use of
group access permissions.

Changes to the system configuration

In some cases, configuration of a software pack-
age involves modification of critical system files, such as:

- Changes to kernels and device drivers.
- Changes to inetd.conf.
- Changes to password and host databases.
- Changes to rc files.

On a small system, it is often possible for the
configuration process to make the necessary
modifications automatically, but this is rarely
successful in a large system, due to conflicts
with existing customizations, or the use of com-
pletely different procedures (e.g., NIS instead of
a local password file). The POSIX standard does
not attempt to address this issue at all and there
seems to be no good multivendor solution to
the problem of making safe changes to the sys-
tem configuration. At present, most system
administrators would probably prefer to super-
vice such critical operations manually, even
though this is time consuming and frequently
requires special solutions, when many different
machines are involved.

Some Common Difficulties

- Distributed authentication schemes such as NIS
  and Kerberos which mean that the local pass-
  word file, if any, may not contain the expected
  information. Attempts to read (or, even worse,
edit) the password file are unlikely to be suc-
cessful on a large system.

- Similarly, the use of DNS or NIS means that
  there may be no valid local host file.

- Automounters are virtually standard in large
NFS installations and may affect the apparent
contents of directories and the pathnames
returned by pwd. Even simple symbolic links in
place of directories have been known to cause
problems with some installation software.

Some Other Problems

A number of other problems that have been
encountered during software installation are due
to a lack of “network awareness” by the installa-
tion procedure. For example:

- The installation may well be performed
remotely and might not have the terminal type,
or window system that would be expected on a
stand-alone machine of the appropriate type.

- Node-locked software licensing is usually inap-
propriate in big networks and the mechanisms
used to implement some schemes simply do not
work in a network context. Even network-based
license managers do not address the needs of
large networks, such as robustness against the
failure of a single-license server.

Conclusions

The POSIX 1003.7.2 standard goes some way
towards alleviating the problems of installing
software onto large distributed systems. How-
ever, there are many configuration issues that are
not addressed by the standard and an awareness
of the special needs of large systems is necessary.
when designing software installation and configuration procedures.

Further Information

The mailing list <soft-managers@nas.nasa.gov> was formed at the LISA workshop for continuing discussion of the above issues (mailing list subscription requests to <soft-managers-request@nas.nasa.gov>).

Draft copies of the POSIX 1003.7.2 standard are available for anonymous ftp from dcdmjw.fnal.gov and the mailing list <bsm@ui.org> is used for detailed discussions of the drafts (mailing list subscription requests to <bsm-request@ui.org>).

Acknowledgements

Thanks to all the participants at the LISA workshop and those who joined in subsequent discussions on the mailing list. In particular, thanks to John Stewart for managing the list, and Matthew Wicks for clarifying the POSIX standard.

SANS II & LISA - What's the Difference?

by Tom Christiansen
<tcchrist@pixel.convex.com>

Several four-letter acronyms spring to mind when thinking of organizations and activities related to systems administration - LISA, SANS, and SAGE. What they have to do with each other and how they're different is a frequently asked question, which I will attempt to answer here.

The USENIX LISA (Large Installation Systems Administration) conferences have been successful vehicles for addressing the needs of systems administrators since 1987. At the early LISAs, attendees were usually experienced systems administrators of large installations, who were relatively UNIX savvy, connected to the Internet, and familiar with USENIX.

This focus has shifted somewhat in recent years as attendance has grown. The "LI" part of LISA is now largely superfluous, as systems administrators from all sorts of sites participate. There, nevertheless, remains a technical core of seasoned systems programmers engaged in the research and development of solutions for problem areas in systems administration. LISA is designed to serve their needs and to advance the state of the art. The vendor displays at LISA complement these endeavors by providing information on products and services that are commercially available.

The SANS II, Conference on Tools & Techniques for Systems Administration, Networking, and Security, on the other hand, differs somewhat in its focus. It grew out of the FedUnix conferences and its audience has largely been composed of individuals associated with federal contractors and Beltway locals. Most of those who attend a SANS do not have access to the Internet, and many aren't on USENET either. SANS is designed to identify the current state of the art for cost-effective system administration and security, so that the techniques and tools used by the most effective managers can be adopted by those who are still seeking solutions.

SANS complements LISA by filling a need that has grown with the recent success of UNIX. Each year thousands of new people take on the responsibility for managing UNIX systems. Only a few have a strong background in UNIX. Many come from environments with PCs and mainframes. People for whom UNIX systems and network management is a brand new experience are more interested in understanding existing practices and strategies than they are in devising new ones. SANS satisfies this demand by providing a venue for many of the the most knowledgeable experts in systems management to talk about the latest, practical solutions and techniques to problems. In addition to topics in systems administration, SANS also covers network management and distributed security.

This year, SAGE (the Systems Administrators Guild) and USENIX are cosponsoring SANS. We are assisting SANS in program formulation by providing program committee members, speakers, and assistance in promoting this event. By supporting both LISA and SANS, we will help ensure that system administrators can find appropriate learning opportunities that fit their unique backgrounds.

This year SANS II will be held in Arlington, VA in April, and and LISA VII in Monterey, CA in November.
SAGE Board Meeting Highlights

by Pat Parseghian, SAGE Secretary

SAGE’s first elected board of directors met for the first time on January 26, 1993 at the USENIX conference in San Diego.

Our first official act was to choose officers for the coming year: Steve Simmons will serve as president, Peg Schafer as treasurer, and Pat Parseghian as secretary. Elizabeth Zwicky holds the position of past president. All board members were able to attend the meeting. In addition to those mentioned above, the board includes Carol Kubicki, Paul Moriarty, and Pat Wilson. Guests at the meeting included Paul Evans and Bjorn Satdeva (from SAGE’s interim board), Greg Rose (SAGE-AU), Tom Christiansen, Ellie Young, and Cynthia Deno (USENIX).

Budget

The board reviewed the budget with Ellie Young. The 1993 budget projects a deficit, even though membership is growing at a faster rate than expected. A deficit is not unusual for a young organization, and USENIX will cover the shortfall as necessary. Some expenses were eliminated from the budget, thanks to our status as a USENIX STG, i.e., we do not need to incorporate or take out a separate liability insurance policy to cover SAGE’s directors. The cost of the services provided by the USENIX office is a rough estimate that will be re-evaluated after six months.

Publicity

We discussed several ways to increase SAGE’s visibility, such as articles in trade publications, mentions in system administration courses and books, and having booths at related organizations’ conferences (e.g., UniForum, DECUS, Interop). Pat Wilson agreed to head a committee, joined by Steve Simmons, Peg Schafer, and Bryan McDonald (chairman of the Publications Working Group) to focus on this issue. We agreed that our message should be that SAGE is “strongly rooted in UNIX”; although we’d like to reach out to all system administrators, we don’t want to spread ourselves too thin from the start.

SANS II

The World Conference on Tools and Techniques for System Administration, Networking, and Security will be held in Washington, D.C., April 18-23. The conference is being run by FedUNIX, and cosponsored with USENIX and SAGE.

Affiliated Groups

The board voted to announce that we have affiliated with SAGE-AU (Australia), and agreed that we should support other SAGE international groups (as well as Local Technical Groups). Pat Parseghian volunteered to be the contact for international groups. Greg Rose represented SAGE-AU and noted that they will soon be actively recruiting members. SAGE-AU would like to see SAGE begin addressing international issues.

The board thanked Bjorn Satdeva for his work on developing a document describing SAGE Local Technical Groups (LTGs).

We agreed that the best approach might be to hand-craft the first few LTGs that approach us, rather than to try to prepare an all-encompassing document at the start.

Working Groups

It was agreed to develop a charter for the working groups, identify a specific task for each group to accomplish, track its progress, and determine whether a group should be dissolved. Carol Kubicki, Paul Moriarty, Pat Parseghian, and Pat Wilson formed a subcommittee to tackle these issues and review the working groups.

We accepted Dave England’s offer to serve as chairperson of the SAGE-vendors Working Group (after the original chairperson stepped down). It was decided to restrict participation in the working groups to SAGE members. This requirement will take effect at the Summer USENIX conference, in order to make it easy for people to join.

Further, no new working groups would be designated until we have drafted a document that describes how the working groups should operate.

SAGE Computer

Many SAGE functions are handled by USENIX on their computer system, and some organizations have made cycles and disc space available to SAGE for online archives.
We agreed that a separate computer for SAGE is not a pressing issue, but one that we should carry forward so we can prepare a specification to match our needs.

Job Placement

The SAGE board agreed that it should be a matter of policy for SAGE not to be involved in job placement or individual recommendations. SAGE board members are free to participate in such activities as long as it is clear they are not acting in any official SAGE capacity.

Terms of Officers

There was a consensus that we expected to re-elect officers each year, following the annual election for new board members. Paul Evans reported that the article he wrote after the SAGE meeting in San Antonio specifically mentioned that officers would serve one-year terms.

Board Meeting Schedule

It was decided to meet at least three times per year at the Winter and Summer USENIX conferences, and at the LISA conference. The next meeting will be held as a conference call on March 5.

---

**History of SAGE**

by Paul Evans

<ple@erg.sri.com>

SAGE has been in existence less than a year, but it is already a rapidly expanding organization with almost 600 members. While the creation of SAGE is a very recent event, very few of its members know how it came about. It therefore seemed like a good idea to give our readers some background about how SAGE got started, and some sense of who the people behind SAGE are, and what they accomplished.

The impetus for SAGE came from the board of directors of BayLISA, an organization for system administrators in the San Francisco Bay Area. BayLISA was founded in December 1990 by seven system administrators who had attended the USENIX LISA IV conference in Colorado Springs (October 1990), and who wanted a local organization that could promote contacts and exchanges of technical information between systems administrators. Bjorn Satdeva has been the President of BayLISA since its inception, and the other founding board members were Shoshana Abrass, John Detke, Paul Evans, Paul Moriarty, Arch Mott, and Elizabeth Zwicky. They were joined by Tina Darmohray, Arnold de Leon, Laura Kirk de Leon and Bryan McDonald. BayLISA held its first program meeting in January 1991, and holds regular meetings every month, usually with an invited speaker discussing a particular topic. Some of the past speakers and topics have been Larry Wall on Perl, Cricket Liu on DNS and BIND, and Paul Vixie on gateway security.

At the USENIX LISA V conference in San Diego (October 1991), several people approached Bjorn Satdeva and Elizabeth Zwicky for assistance in setting up local groups for systems administrators similar to BayLISA. For the Boston area, there were a sufficient number of systems administra-
tors to make a local group, Back Bay LISA, possible. But others who expressed interest live and work in areas where this is not the case. This led to an ongoing discussion among the members of the BayLISA board during the fall of 1991 about whether or not BayLISA should be in the business of promoting the formation of other local groups; if not, what kind of organization should be doing so; and what should be done to serve the needs of systems administrators who, for geographical reasons, aren't able to participate in a local group.

At the end of December 1991, Paul Moriarty suggested to the BayLISA board that they consider whether or not BayLISA should get more involved in advocating system administration as a career. There was a brief discussion at the end of the January 1992 board meeting in which Moriarty elaborated on his belief that the conditions in the systems administration community were right for putting together a national organization of systems administrators. It was agreed that the next BayLISA board meeting would be devoted exclusively to the discussion of what was described as a "Sysadmin Guild/SIG."

At the February meeting, Paul Evans discussed the circumstances surrounding the rise of engineering professional societies, which took place in an environment characterized by concern about the status of engineering professions, demands that standards for work performance come from professional peers and not from employers, and the establishment of engineering fields as recognized academic disciplines. The consensus of the group was that systems administration appears to be in the early stages of emerging as a profession, and faces many similar concerns.

The group discussed the types of activities such a professional organization for systems administrators might undertake, among which were providing standardized job descriptions and a certification program, and fostering local groups like BayLISA and Back Bay LISA. At this meeting, the group decided that such a project was not in the purview of BayLISA, a purely local and to some extent social organization, and that it should be carried out as an entirely separate activity. It was also at this meeting that the name SAGE, derived from "Systems Administrators' Guild," was first suggested. Many members of the working group felt that the word "guild" accurately captured the apprenticeship process by which most systems administrators are trained, and liked the fact that the word "sage" has a nice connotation of wisdom.

The SAGE working group met weekly throughout the spring. Bjorn Satdeva served as the moderator for the discussions, and Tina Darmohray and Arnold de Leon acted as secretaries, taking notes for the group. Paul Evans wrote the first draft of the SAGE charter and led the discussions aimed at refining it. Shoshana Abrass drafted a preliminary business plan for the organization. The goal of all this activity was to have some concrete proposal to put before the systems administration community in time for the USENIX Summer Conference in San Antonio (June 1992), and to be able to have an operating organization by the time of the USENIX LISA VI Conference in Long Beach (October 1992).

Initially, the SAGE working group conceived of SAGE as being an independent organization. In March, Bjorn Satdeva contacted Ellie Young of USENIX to ask general questions about what was involved in setting up and running such an organization. The result was an unexpected proposal that SAGE, instead, become a USENIX Special Technical Group. The SAGE working group felt that associating with USENIX would bring substantial benefits to both SAGE and USENIX, and therefore decided to negotiate an agreement under which USENIX would "launch" SAGE as its first STG.

In the late spring, the attention of the working group focused increasingly on the events planned for the USENIX conference in San Antonio in early June. Two new members, Steve Simmons and Pat Wilson, joined the working group at this time. Because they were "geographically challenged" (Steve lives in Michigan and Pat in New Hampshire), they participated in board meetings by conference call, an arrangement that took a little while to get used to, but proved surprisingly effective. Because SAGE was to be the first STG within USENIX, and there was therefore no model for what an STG should look like, there were extensive negotiations between Steve Johnson and Ellie Young for USENIX and Bjorn Satdeva, Paul Evans, Arnold de Leon, and John Detke for SAGE and about how the relationship between USENIX and the STGs in general, and SAGE in particular, should be structured. Also at this time, in anticipation of a USENIX board resolution launching SAGE, the working group elected officers, Elizabeth Zwicky becoming president, John Detke treasurer and Tina Darmohray secretary. Bryan McDonald was designated publications coordinator and editor of the SAGE newsletter.

SAGE was officially launched by the USENIX board at its June 8, 1992 meeting, and the members of the working group were appointed to the SAGE interim board. The SAGE by-laws and the STG
document that were approved at this meeting were not entirely consistent, and several more months of work were required to bring them into agreement. Steve Simmons moderated the first public meeting of SAGE, held on June 13 and attended by over fifty systems administrators, at which most of the organizational details, such as the number and terms of board members and officers, were decided. At the conclusion of the public meeting, a number of working groups were set up to make recommendations to the SAGE board about how it could best serve its members in a number of different areas, such as education, job descriptions, and online services.

With the San Antonio meeting successfully completed, the interim SAGE board turned its attention to handing over a viable organization to a new elected board. The interim board made arrangements for the election of the new board, and assisted in the creation of SAGE's first international affiliate, SAGE-Australia. At the USENIX LISA Conference in Long Beach (October 1992) Steve Johnson, Ellie Young, Elizabeth Zwicky, and Paul Evans completed work on the final versions of the SAGE by-laws and the USENIX STG document. Also at the LISA Conference, Rob Kolstad moderated a well-attended SAGE board candidates' forum. At the Sun User Group meeting in December, Bjorn Satdeva, Elizabeth Zwicky, Steve Johnson, and Ellie Young, completed the final version of the USENIX LTG (Local Technical Group) document. At the same time, the Job Descriptions Working Group and Online Working Group made substantial progress toward their goals, while the Publications Working Group produced SAGE sections for three issues of *login:* The year ended with the election of six new members of the board of directors from a field of twelve candidates.

When the newly elected SAGE board met for the first time at the Winter USENIX Conference in San Diego, they took control of an established organization with 575 members. At this point, SAGE has a firm organizational structure, which the system administration community can now use to achieve its goals.
There are four major species of UNIX sysad:

1) *The Technical Thug.* Usually a systems programmer who has been forced into system administration; writes scripts in a polyglot of the Bourne shell, *sed, C, awk, perl,* and APL.

2) *The Administrative Fascist.* Usually a retentive drone (or rarely, a harridan ex-secretary) who has been forced into system administration.

3) *The Maniac.* Usually an aging cracker who discovered that neither the Mossad nor Cuba are willing to pay a living wage for computer espionage. Fell into system administration; occasionally approaches major competitors with bizarre schemes.

4) *The Idiot.* Usually a cretin, morphodite, or old COBOL programmer selected to be the system administrator by a committee of cretins, morphodites, and old COBOL programmers.

How To Identify Your System Administrator

**Situation:** Low disk space

*Technical Thug:* Writes a suite of scripts to monitor disk usage, maintain a database of historic disk usage, predict future disk usage via least squares regression analysis, identify users who are more than a standard deviation over the mean, and send mail to the offending parties. Places script in *cron.* Disk usage does not change, since disk-hogs, by nature, file all mail away in triplicate without reading it.

*Administrative Fascist:* Puts disk usage policy in *motd.* Uses disk quotas. Locks accounts that go over quota.

*Maniac:* 
```bash
# cd /home
# rm -rf 'du -s * | sort -rn | head -1 | awk '{print $2}''
```

*Idiot:* 
```bash
# cd /home
# cat 'du -s * | sort -rn | head -1 | awk '{ printf "%s/\n", $2}'' | compress
```

**Situation:** Excessive CPU usage

*Technical Thug:* Writes a suite of scripts to monitor processes, maintain a database of CPU usage, identify processes more than a standard deviation over the norm, and *renice* offending processes. Places script in *cron.* Ends up renicing the production database into oblivion, bringing operations to a grinding halt, much to the delight of the *xtrek* freaks.

*Administrative Fascist:* Puts CPU usage policy in *motd.* Uses CPU quotas. Locks accounts that go over quota. Allows no exceptions, thus crippling development work, much to the delight of the *xtrek* freaks.

*Maniac:* 
```bash
# kill -9 'ps -auxww | sort -rn +8 -9 | head -1 | awk '{print $2}''
```

*Idiot:* 
```bash
# compress -f 'ps -auxww | sort -rn +8 -9 | head -1 | awk '{print $2}''
```

**Situation:** New account creation

*Technical Thug:* Writes perl script that creates home directory, copies in incomprehensible default environment, and places entries in *etc/passwd, etc/shadow,* and *etc/group* (by hand, NOT with *passmgmt*). Slaps on *setuid* bit; tells a nearby secretary to handle new accounts. Usually, said secretary is still dithering over the difference between 'enter' and 'return'. No new accounts are ever created.

*Administrative Fascist:* Puts new account policy in *motd.* Since people without accounts cannot read the *motd,* nobody ever fulfills the bureaucratic requirements; so, no new accounts are ever created.

*Maniac:* "If you’re too stupid to break in and create your own account, I don’t want you on the system. We’ve got too many idiots on this box anyway."

*Idiot:* 
```bash
# cd /home; mkdir "Bob's home directory"
# echo "Bob Simon:gandalf:0:0:/dev/tty:compress -f" > /etc/passwd
```

**Situation:** Root disk fails

*Technical Thug:* Repairs drive. Usually is able to repair filesystem from boot monitor. Failing that,
front-panel toggles microkernel in and starts script on neighboring machine to load binary boot code into broken machine, reformat and reinstall OS. Lets it run over the weekend while he goes mountain climbing.

**Administrative Fascist:** Begins investigation to determine who broke the drive. Refuses to fix system until culprit is identified and charged for the equipment.

**Maniac, Large System:** Rips drive from system, uses sledgehammer to smash same to flinders. Calls manufacturer, threatens lawsuit. Abuses field engineers while they put in a new drive and reinstall the OS.

**Maniac, Small System:** Rips drive from system, uses ball-peen hammer to smash same to flinders. Calls Requisitions, threatens pets. Abuses bystanders while putting in new drive and reinstalling OS.

**Idiot:** Doesn’t notice anything wrong.

**Situation:** Poor network response

**Technical Thug:** Writes scripts to monitor network, then rewires entire machine room, improving response time by 2%. Shrugs shoulders, says, “I’ve done all I can do,” and goes mountain climbing.

**Administrative Fascist:** Puts network usage policy in `motd`. Calls up Berkeley and AT&T, badgers whoever answers for network quotas. Tries to get `xtrek` freaks fired.

**Maniac:** Every two hours, pulls Ethernet cable from wall and waits for connections to time out.

**Idiot:**
```
# compress -f /dev/en0
```

**Situation:** User questions

**Technical Thug:** Hacks the code of `emacs`’ doctor-mode to answer new users questions. Doesn’t bother to tell people how to start the new “guru-mode”, or for that matter, `emacs`.

**Administrative Fascist:** Puts user support policy in `motd`. Maintains queue of questions. Answers them when he gets a chance, often within two weeks of receipt of the proper form.

**Maniac:** Screams at users until they go away. Sometimes barters knowledge for powerful drink and/or sycophantic adulation.

**Idiot:** Answers all questions to best of his knowledge until the user realizes few UNIX systems support punched cards or JCL.

**Situation:** *Stupid* user questions

**Technical Thug:** Answers question in hex, EBCDIC, and/or French until user gives up and goes away.

**Administrative Fascist:** Locks user’s account until user can present documentation demonstrating their qualification to use the machine.

**Maniac:**
```
# cat >> -luser/.cshrc alias vi 'rm !*;unalias vi;grep -v BoZo ~/.cshrc > ~/.z; mv -f ~/.z ~/.cshrc'
```

**Idiot:** Answers all questions to best of his knowledge. Recruits user to become part of system administration team.

**Situation:** Religious war, BSD vs. System V

**Technical Thug:** BSD. Crippled on System V boxes.

**Administrative Fascist:** System V. Horrified by the people who use BSD. Places frequent calls to DEA whenever requests for BSD features are made.

**Maniac:** Prefers BSD, but doesn’t care as long as his processes run quickly.

**Idiot:**
```
# cd c:
```

**Situation:** OS upgrade

**Technical Thug:** Reads source code of new release, takes only the modules he likes.

**Administrative Fascist:** Instigates lawsuit against the vendor for having shipped a product with bugs in it in the first place.

**Maniac:**
```
# uptime
1:33pm up 19 days, 22:49, 167 users, load average: 6.49, 6.45, 6.31
```
```
# wall
Well, it's upgrade time. Should take a few hours. And good luck on that 5:00 deadline, guys! We're all pulling for you!
```
```
^D
```

**Idiot:**
```
# dd if=/dev/rmt8 of=/vmunix
```

**Situation:** Balky mail

**Technical Thug:** Rewrites `sendmail.cf` from scratch. Rewrites `sendmail` in SNOBOL. Hacks kernel to implement file locking. Hacks kernel to implement “better” semaphores. Rewrites `sendmail` in assembly. Hacks kernel to...

**Administrative Fascist:** Puts mail use policy in


*mo*/. Locks accounts that go over mail use quota. Keeps quota low enough that people go back to interoffice mail, thus solving problem.

**Maniac:**

```
# kill -9 'ps -auxww | grep sendmail | awk '{print $2}''
# rm -f /usr/spool/mail/*
# wall
Mail is down. Please use interoffice mail until we have it back up.
```

```
# write max
I've got my boots and backpack. Ready to leave for Mount Tam?
```

**Idiot:**

```
# echo "HELP!" | mail tech_support.AT.vendor.com%kremvax%bitnet I BIFF!!
```

**Situation:** Users want phone list application

**Technical Thug:** Writes RDBMS in perl and Smalltalk. Users give up and go back to post-it notes.

**Administrative Fascist:** Oracle. Users give up and go back to post-it notes.

**Maniac:** Tells the users to use flat files and `grep`, the way God meant man to keep track of phone numbers. Users give up and go back to post-it notes.

```
# dd ibs=80 if=/dev/rdisk001s7 | grep "Fred"
```

**Hobbies, Technical**

**Technical Thug:** Writes entries for Obsfuscated C contest. Optimizes INTERCAL scripts. Maintains ENIAC emulator. Virtual reality.

**Administrative Fascist:** Bugs offices. Audits card-key logs. Modifies old TVs to listen in on cellular phone conversations. Listens to police band.

**Maniac:** Volunteers at Survival Research Labs. Bugs offices. Edits card-key logs. Modifies old TVs to listen in on cellular phone conversations. Jams police band.

**Idiot:** Ties shoes. Maintains COBOL decimal to roman numeral converter. Rereads flowcharts from his salad days at Rand.

---

**Analogies to Live Without: You don’t need a manual to drive a car.**

by Elizabeth Zwicky

<zwicky@erg.sri.com>

When people start complaining about how complex computers are, they often assert that manuals should be unnecessary - after all, you don't need a manual to drive a car. You can rent a car of a make you've never driven, get in it, and drive it away.

There are two basic problems with this assertion. First, there's an initial learning curve difference. When I learned to drive a car, I had been riding in cars for 15 years; I had a solid base of relevant knowledge gained from riding bicycles; and I was trained by a 3 month process involving books, classroom instruction, films, simulators, and many, many hours of practice, all of it with someone present to coach me. During this process, I used various different cars, in order to help me learn to transfer these skills from car to car. When I learned to use a computer, I had only seen it done once or twice before, and I was plopped down in front of one with a book and another novice. Furthermore, this casual approach actually worked; using only a book and my wits, I learned to make the computer do what I wanted. I don't believe that this would have been possible with the car.

I am now more or less able to make any computer work, from an IBM PC to a IBM 370, using only a book. On the other hand, the only cars I can drive are automatic transmission passenger cars; without another training process, I can't drive a motorcycle or a truck.

Second, it's simply not true that you don't need a manual to drive a new car. I have been in a car with three other reasonably intelligent computer professionals, at a gas station, unable to open the gas tank for the several minutes that it took us to figure out how to open the glove compartment to get at the manual. (As it turned out, the manual wasn't in the glove compartment, but the release for the gas cap was, much to our surprise.) I once had to refer to the manual to figure out how to get the key out of the ignition, in a car with the same manufacturer and model year as the car I normally drove. Sure, you can generally start the car, and with a little diligence you can usually work out how to turn on the lights and run the windshield wipers (particularly if it's light out and not raining), and you can almost always get the seatbelts to work, but what good does that do you if you can't put gas in it?
At that, cars are much better in this respect than a lot of the other appliances we work with. Nobody ever asserts that computers should be more like VCRs, because almost everybody has a VCR at home blinking “12:00.” Half the coffee-makers I’ve seen are effectively impossible to use without a manual, and three of the seven buttons on the little CD player I use at work do incomprehensible modal things. Compared to this, UNIX starts to look real good.

I’m not saying that current computer interfaces are paragons of human engineering; I swear at my computer plenty. But let’s make fair comparisons. Cars have been around for about 100 years now, and computers for less than half that. The tasks that cars perform are extremely well understood and cover only a very small range – they move around in two dimensions, and they keep people inside reasonably safe while doing so. Computers, on the other hand, perform a wide range of dissimilar tasks. Finally, cars are incredibly familiar objects, which most Americans have experience with starting only days after they are born. With all these advantages, we still put vastly more effort into teaching people to drive than into teaching them to use computers. Why is it the computer interface that people say is hard to use?

For one thing, you only have to learn to drive once. Cars are so standardized within types that you can buy a new car without a large extra increment of learning. You have to look at a manual, but not for long. This is partly because they’ve been around longer, and partly because they are specialized to a small range of tasks. You do have to go through a longer learning period if you need a vehicle that has new capabilities; that may be a short time, if you’re adapting to a van or an off-road vehicle, or a long time, if you’re adapting to an 18-wheeler. Second, so much of people’s time is spent around cars that they are effectively experts. People experience different cars all the time, and even when they aren’t actually driving, they’re picking up information about the cars they’re in. Third, driver’s education is a cultural norm. Everybody knows what it takes to learn to drive, whether they’ve done it or only seen it done on TV. Fourth, it seems only fair that it takes a while to learn to do something that is so dangerous. Several tons of steel, garnished with protective devices, inspires respect.

Computer training doesn’t transfer as easily from one machine to the next, most people aren’t computer experts, computer training isn’t something that everybody does, and computer accidents rarely kill people. Thus, people actually hold computers to a higher usability standard than cars. Computers actually do pretty well, considering how flexible and how new they are. It would be nice if they did better, but cars are not a standard for improvement.

System Administration Tools Your Vendor Never Told You About: The Laminator

by Elizabeth Zwicky
<zwicky@erg.sri.com>

A laminator is about $300 worth of completely noncomputerized electrical equipment. It takes flat things that involve pieces of plastic, and heats them up while squishing them. Different laminators allow different configurations, but the general principle is that you put a piece of paper between two pieces of plastic, run it through the machine, and come out with a solid block of plastic with the paper embedded in it. (This is the gizmo vendors use at trade shows to turn your business card into a luggage tag.)

The point of this is that it produces signs, tags, and stickers that say whatever you want but withstand age and mishandling well. In combination with a laser printer, it produces slick-looking signs and notices. (In combination with a color printer, it produces absolutely amazing signs.) We laminate all of our signs (including the ones that we move around, like “This equipment under repair. Do not attempt to use or reassemble.”) and any instructions that we want to put next to equipment. As well as full sheets, you can laminate tags for cables, and you can make labels. The printing wears off laser-printed labels quite fast, but laminated labels last forever, and since you’re printing plain paper they don’t jam in the printer.
The Ten Commandments for C Programmers

(Annotated Edition)
by Henry Spencer
<henry@zoo.toronto.edu>

Thou shalt run lint frequently and study its pro-
nouncements with care, for verily its perception
and judgement oft exceed thine.

This is still wise counsel, although many modern
compilers search out many of the same sins, and
there are often problems with lint being aged and
infirm, or unavailable in strange lands. There are
other tools, such as Saber C, useful to similar
ends.

"Frequently" means thou shouldst draw thy
daily guidance from it, rather than hoping thy
code will achieve lint's blessing by a sudden act of
repentance at the last minute. De-linting a pro-
gram which has never been linted before is often
a cleaning of the stables such as thou wouldst not
wish on thy worst enemies. Some observe, also,
that careful heed to the words of lint can be quite
helpful in debugging.

"Study" doth not mean mindless zeal to eradicate
every byte of lint output - if for no other reason,
because thou just canst not shut it up about some
things - but that thou should know the cause of
its unhappiness and understand what worrisome
sign it tries to speak of.

Thou shalt not follow the NULL pointer, for
chaos and madness await thee at its end.

Clearly the holy scriptures were mistranscribed
here, as the words should have been "null
pointer," to minimize confusion between the con-
cept of null pointers and the macro NULL (of
which more anon). Otherwise, the meaning is
plain. A null pointer points to regions filled with
dragons, demons, core dumps, and numberless
other foul creatures, all of which delight in frolic-
ing in thy program if thou disturb their sleep. A
null pointer doth not point to a 0 of any type,
despite some blasphemous old code which impi-
ously assumes this.

Thou shalt cast all function arguments to the
expected type if they are not of that type already,
even when thou art convinced that this is unne-
cessary, lest they take cruel vengeance upon thee
when thou least expect it.

A programmer should understand the type struc-
ture of his language, lest great misfortune befall
him.

Contrary to the heresies espoused by some of the
dwellers on the Western Shore, 'int' and 'long' are
not the same type. The moment of their equiva-
Ience in size and representation is short, and the
agony that awaits believers in their interchange-
ability shall last forever and ever once 64-bit
machines become common.

Also, contrary to the beliefs common among the
more backward inhabitants of the Polluted East-
ern Marshes, NULL does not have a pointer type,
and must be cast to the correct type whenever it
is used as a function argument.

(The words of the prophet Ansi, which permit
NULL to be defined as having the type 'void *
are oft taken out of context and misunderstood.
The prophet was granting a special dispensation
for use in cases of great hardship in wild lands.
Verily, a righteous program must make its own
way through the Thicket Of Types without lazi-
ly relying on this rarely-available dispensation to
solve all its problems. In any event, the great
deity Dmr who created C hath wisely endowed it
with many types of pointers, not just one, and
thus it would still be necessary to convert the
prophet's NULL to the desired type.)

It may be thought that the radical new blessing of
"prototypes" might eliminate the need for cau-
tion about argument types. Not so, brethren.
Firstly, when confronted with the twisted
strangeness of variable numbers of arguments,
the problem returns... and he who has not kept
his faith strong by repeated practice shall surely
fall to this subtle trap. Secondly, the wise men
have observed that reliance on prototypes doth
open many doors to strange errors, and some
indeed had hoped that prototypes would be
decreed for purposes of error checking but would
not cause implicit conversions. Lastly, reliance on
prototypes causeth great difficulty in the Real
World today, when many cling to the old ways
and the old compilers out of desire or necessity,
and no man knoweth what machine his code may
be asked to run on tomorrow.
If thy header files fail to declare the return types of thy library functions, thou shalt declare them thyself with the most meticulous care, lest grievous harm befall thy program.

The prophet Ansí, in her wisdom, hath added that thou shouldst also scourge thy Suppliers, and demand on pain of excommunication that they produce header files that declare their library functions. For truly, only they know the precise form of the incantation appropriate to invoking their magic in the optimal way.

The prophet hath also commented that it is unwise, and leads one into the pits of damnation and subtle bugs, to attempt to declare such functions thyself when thy header files do the job right.

Thou shalt check the array bounds of all strings (indeed, all arrays), for surely where thou typest “foo” someone someday shall type “supercalifragilisticexpialidocious”.

As demonstrated by the deeds of the Great Worm, a consequence of this commandment is that robust production software should never make use of gets(), for it is truly a tool of the Devil. Thy interfaces should always inform thy servants of the bounds of thy arrays, and servants who spurn such advice or quietly fail to follow it should be dispatched forthwith to the Land Of Rm, where they can do no further harm to thee.

If a function be advertised to return an error code in the event of difficulties, thou shalt check for that code, yea, even though the checks triple the size of thy code and produce aches in thy typing fingers, for if thou thinkest “it cannot happen to me,” the gods shall surely punish thee for thy arrogance.

All true believers doth wish for a better error-handling mechanism, for explicit checks of return codes are tiresome in the extreme and the temptation to omit them is great. But until the far-off day of deliverance cometh, one must walk the long and winding road with patience and care, for thy Vendor, thy Machine, and thy Software delight in surprises and think nothing of producing subtly meaningless results on the day before thy Thesis Oral or thy Big Pitch To The Client.

Occasionally, as with the `ferror` feature of `stdio`, it is possible to defer error checking until the end when a cumulative result can be tested, and this often produceth code which is shorter and clearer. Also, even the most zealous believer should exercise some judgement when dealing with functions whose failure is totally uninteresting... but beware, for the cast to void is a two-edged sword that sheddeth thine own blood without remorse.

Thou shalt study thy libraries and strive not to re-invent them without cause, that thy code may be short and readable and thy days pleasant and productive.

Numberless are the unwashed heathen who scorn their libraries on various silly and spurious grounds, such as blind worship of the Little Tin God (also known as “Efficiency”). While it is true that some features of the C libraries were ill-advised, by and large it is better and cheaper to use the works of others than to persist in re-inventing the square wheel. But thou should take the greatest of care to understand what thy libraries promise, and what they do not, lest thou rely on facilities that may vanish from under thy feet in future.

Thou shalt make thy program’s purpose and structure clear to thy fellow man by using the One True Brace Style, even if thou likest it not, for thy creativity is better used in solving problems than in creating beautiful new impediments to understanding.

These words, alas, have caused some uncertainty among the novices and the converts, who knoweth not the ancient wisdoms. The One True Brace Style referred to is that demonstrated in the writings of the First Prophets, Kernighan and Ritchie. Often and again it is criticized by the ignorant as hard to use, when in truth it is merely somewhat difficult to learn, and thereafter is wonderfully clear and obvious, if perhaps a bit sensitive to mistakes.

While thou might think that thine own ideas of brace style lead to clearer programs, thy successors will not thank thee for it, but rather shall revile thy works and curse thy name, and word of this might get to thy next employer. Many customs in this life persist because they ease friction and promote productivity as a result of universal agreement, and whether they are precisely the optimal choices is much less important. So it is with brace style.

As a lamentable side issue, there has been some unrest from the fanatics of the Pronoun Gestapo over the use of the word “man” in this Commandment, for they believe that great efforts and loud shouting devoted to the ritual purification of the language will somehow redound to the benefit of the downtrodden (whose real and grievous woes tendeth to get lost amidst all that thunder and fury). When preaching the gospel to the narrow of mind and short of temper, the word
“creature” may be substituted as a suitable pseudo-Biblical term free of the taint of Political Incorrectness.

Thy external identifiers shall be unique in the first six characters, though this harsh discipline be irksome and the years of its necessity stretch before thee seemingly without end, lest thou tear thy hair out and go mad on that fateful day when thou desirest to make thy program run on an old system.

Though some hasty zealots cry “not so; the Millennium is come, and this saying is obsolete and no longer need be supported,” verily there be many, many ancient systems in the world, and it is the decree of the dreaded god Murphy that thy next employment just might be on one. While thou sleepest, he plotteth against thee. Awake and take care.

It is, note carefully, not necessary that thy identifiers be limited to a length of six characters. The only requirement that the holy words place upon thee is uniqueness within the first six. This often is not so hard as the belittlers claimeth.

Thou shalt foreswear, renounce, and abjure the vile heresy which claimeth that “All the world’s a VAX,” and have no commerce with the benighted heathens who cling to this barbarous belief, that the days of thy program may be long even though the days of thy current machine be short.

This particular heresy bids fair to be replaced by “All the world’s a Sun,” or “All the world’s a 386” (this latter being a particularly revolting invention of Satan), but the words apply to all such without limitation. Beware, in particular, of the subtle and terrible “All the world’s a 32-bit machine,” which is almost true today but shall cease to be so before thy resume grows too much longer.

---

**One Day at School**

by Jeff West  
< jwest@els.cray.com>

I’m taking the CS 480 course, Operating System Principles, at the U. this semester. Last night the professor got into explaining the UNIX `fork()` call and one of the students just couldn’t get it through his head that fork actually makes a copy of the process you are running (thus yielding two processes).

The code that he was using to demonstrate was as follows:

```c
if (fork() != 0)
{
    printf("I'm the parent.\n");
}
else
{
    printf("I'm the child.\n");
}
```

The student/professor banter went something like:

S: So the kernel will return the process ID and we’ll print “I’m the parent.”

P: Correct.

S: But if it returns a 0 we print, “I’m the child.”

P: Correct.

S: So which one will it return?

P: Both.

[loooooong pause]

S: But the kernel can only return one value.

P: Correct.

S: Well which one does it return?

P: Both.

[loooooooonger pause]

Abbott and Costello would have been proud.
About PEM

by Christopher (C.J.) Rath
JetForm Corp.
<crath@bnr.ca>

To: Matt Bishop

I was very interested in your article on PEM. [See the November/December 1992 issue of ;login: - Ed.] Thank you for taking the time to write it. I look forward to your following article.

After the invited talk "Resource Discovery and Network Measurement in the Global Internet" by Michael Schwartz of U. of Colorado was presented, a colleague and I had a long discussion regarding PEM in the light of Michael's work.

Some of what Michael did was to track Email headers in order to discover patterns of communication. This has privacy implications. Do I necessarily want someone tracking who I communicate with?

The immediate solution we discussed was that of encrypting everything in the header except the To: address line. While this does cover part of the trail there is still exposure due to message routers building a Path: header component.

The Path: component can be spoofed through the use of what I will call "PEM firewalls." These would be Internet mail gateways to which users would register. Their purpose is to spoof Path: trackers. The easiest way to express how I think they would work is by example:

1) I send mail to John Smith, who has communicated the address to me in a secure fashion. Say, I Email to <mailer@pem1.com>. I encrypt the message twice: The message body with John's public key, and the message header (from:, reply-to:, subject:, etc.) with mailer@pem1.com's public key. I also provide an extra header line (encrypted) which has John's email id in it, which pem1.com will make use of.

2) pem1.com decodes the header and determines who the message is for by looking at the extra header line. It then reroutes the message. John may be on a subnet, of which pem1.com is a gateway, or John may be somewhere else (even through yet another PEM fire wall). Exactly how this message is handled depends upon where John is.

3) The from: and reply-to: lines of my message to John do not necessarily contain a real address. They may be pointers to a PEM fire wall where I have registered myself.

I may be overly paranoid in suggesting these PEM fire walls. In any case I think you've got the idea.

When I returned home from USENIX I reread your article. It doesn't address this idea of header encryption at all (that I can see). Would you discuss this aspect of PEM (or why it isn't an aspect if that is the case) in an upcoming article?

Matt Bishop's Reply:

Christopher,

Yes, I'll discuss it later. But PEM is designed to provide specific security services: confidentiality of message, data authentication (integrity), user authentication, and (if you use an asymmetric cryptosystem like RSA for the interchange cryptosystem) nonrepudiation. It very specifically is not designed to prevent traffic analysis which, as you note, is a much more complex problem. (As you note, the PEM headers are not encrypted.)

By the way, one problem with your proposed solution is political: you need to get users to trust that the PEM fire walls are not recording traffic flowing through them. That's one reason we didn't consider providing a solution to the traffic analysis attacks feasible, because the goal of PEM is to minimize trust in the system. Currently you need only trust the system the software is running on, and the software itself; if you have special equipment (a crypto box, for example, or a trusted PC for doing PEM), your notion of "trust" can be very limited. Note you need not trust your certifying authority or the policy certification authority or the Internet Society, because they never see your private key, and you can check your certificate, and theirs, using the IS's public key and certificates. (All this will be described in the next column.)

If this is confusing, I apologize; one of my future columns will be on what types of trust in the PEM system you need. ("System" here being the protocols and generic systems implementing them, not any specific software or hardware implementation.)

Hope this helps!
Matt Bishop
Middle Aged Addiction

by Rob Kolstad
<kolstad@bsd1.com>

Like so many of us, some of my first exposures to computing were via computer games. I remember playing blackjack against an IBM 1620 back in seventh grade and carrying the console printout home with reverence. It was amazing to me. Such things are valued for their rarity, of course. Few students are quite as enthralled about IBM 1620 blackjack these days.

The first peak in my game-playing sophistication came in the middle of high school with the UCLA Executive Game #2—a simulation of a competitive industry in which opponents determined the price of a product, its advertising and R&D budgets, factory improvement and maintenance costs, and stock dividends. The computer referred the marketplace and produced astounding (to me at the time) financial reports. A similar simulation is now used nationally by Junior Achievement.

In the 1980s, I attended the University of Illinois at which the PLATO computer system was designed and implemented. PLATO's goal was the revolutionizing of teaching. PLATO had lessons, multimedia displays (touch panels, rear projection slides, voice, random-access audio), and a sophisticated programming language for determining correct answers. PLATO's surprising strength, though, was its games. The way the designers implemented shared memory coupled with the extraordinarily powerful processors (for their day), made PLATO's games just fantastic. I particularly enjoyed EMPIRE, a space game in which players pilot ships belonging to one of four space-faring races and attempt to conquer the known universe.

1987 Computer Oracle

by UC Berkeley CS Faculty

[Ed. Note: Thanks to Dr. Carlo Sequin for collecting this oracle information. No particular evaluations of the predictions are given here; the predictions themselves are fascinating.]

At the faculty retreat in Spring 1987, the [UCB] faculty submitted random predictions about the future of computer science, and then each faculty member cast 5 votes for the most relevant of all the submitted quotes. Here are the predictions that received any votes at all, ordered by topical area.

Computer Science

The fundamental problems of our science will continue to be what they have always been: the

All three of the above games were simulations; it only took me 15 years to realize this common thread. I recently came to own two of the newer genre of computer simulations: SimCity and Sid Meiers' CIVILIZATION. In both of these games, the “player” is the leader who makes decisions about the creation of cities, in the case of SimCity, and entire civilizations, in CIVILIZATION.

Interestingly enough, the two new games are accurate enough that one can draw interesting (and, I presume, somewhat valid) conclusions about the results of good and bad decisions. Over Christmas, I returned briefly to Norman, Okla. (my boyhood home) after five years of absence. The urban growth (and decay) patterns matched quite well with the results of SimCity, given the various geographical patterns and zoning, which were far easier to observe, given the game's insight. My parents showed me a history book with the original plats from the late 1800s. Again, the SimCity simulations made it easy to understand the way the city grew.

The CIVILIZATION game is harder, for me, than SimCity. It has a goal (well, it has five different goals) and one can certainly win or lose. I find it humbling when the Egyptians come across the sea in their triremes and take away one of my better cities (like Paris). I haven't played the game much above level 3 (of 5), yet. I'm told that higher level opponents are very sophisticated. The game is complex enough to hold my interest for hours at a stretch (hours which feel like minutes).

If you enjoy simulations that challenge your thinking, teach you principles of local and global management, and are very, very addictive, you might want to try one of these games. Both are available for PCs; SimCity is soon to be available for Suns (according to the rumor mill).
understanding of knowledge representation, data structures and algorithms. Computer Science will be less concerned with computation than with managing information and communication.

The most important problems in Computer Science are:

1. Knowledge Representation - efficient, effective representation of knowledge for vision, speech...
2. Software Crisis - reliable, predictable construction of large complex software systems.
3. Distributed Algorithms - design, implementation and analysis of distributed algorithms.
4. Manufacturing Automation - information management and vision/robotics.

In five-to-ten years, any “Survey Talk” by one of our colleagues at a departmental colloquium will be understood by at most one other colleague and two graduate students.

Parallel Computing

Parallel computing is our only hope for improved cost/performance in 5 years – if we don’t make advances, our field will stagnate.

The most important task ahead of us is to understand how to use parallel computing. Parallel computing will only make marginal advances; it will be concentrated on special problems like matrix inversion or image processing. In general, sequential computers will still be faster for the same price.

Workstations – Network

By 1992 there will be only two classes of computers: personal computers and supercomputers. 99% of us will use PCs. By 1992 the workstation market has crashed: no new applications can be found that would demand higher performance.

We must move beyond UNIX to integrated programming/writing environments that support multimedia data representations.

Remote resource sharing will become increasingly important (running pieces of your AI program in Burma and pieces in Australia).

Hardware – Software

[Hardware] manufacturers have pulled far ahead of software technology and are increasing the gap. In particular: in 1990 we can have desktop workstations each equivalent in power to 50 – 80 VAX 11/780s, with superb displays, large memories, large disks. But most of us will have no particular use for such speed, and we will find the same barriers: software in disarray, high learning costs for users, expensive maintenance costs.

World dollar purchases of hardware per year will peak and decline by 1992. IBM will get more than 50% of its revenue from software. Hardware and architecture will become uninteresting.

Computer architecture will soon become “just a matter of software.”

Languages

In ten years, most programming will be done in nonprocedural languages.

Logic Programming will be used by a larger fraction of all computer scientists than use it today.

More than two thirds of the programs written by our graduate students will be written in LISP (or some dialect thereof). LISP will be the Fortran of the 90’s.

Artificial Intelligence and Expert Systems

By 1992 one-half of the current faculty will be doing what is now classified as AI.

The field of AI will still not have licked the common-sense understanding problem.

Expert systems will be reduced to front ends for data-base systems, and the fields will merge.

Expert system shells will have gone the way of PL/1.

Our [UCB] Computer Science Division

The demand for Computer Science Ph.D.s will have grown by 400%.

The number of undergraduate Computer Science majors will have dropped below 200.

The major portion of our industrial funding will come from Japan.

Corrigendum: USENIX Proceedings

by Rob Kolstad
<kolstad@bsd1.com>

Rob Pike has graciously forgiven me and supplied the page I inadvertently omitted from his paper in 1993 Winter USENIX conference proceedings. I much appreciate Rob’s patience in this matter. The correct page follows on page 35.
NAME
  UTF, Unicode, ASCII, rune - character set and format

DESCRIPTION
  The Plan 9 character set and representation are based on Unicode and on a proposed X-Open multibyte
  FSS-UCS-TF (File System Safe Universal Character Set Transformation Format) encoding. Unicode rep¬
  resents its characters in 16 bits; FSS-UCS-TF, or just UTF, represent such values in an 8-bit byte stream.

  In Plan 9, a rune is a 16-bit quantity representing a Unicode character. Internally, programs may store char¬
  acters as runes. However, any external manifestation of textual information, in files or at the interface
  between programs, uses a machine-independent, byte-stream encoding called UTF.

  UTF is designed so the 7-bit ASCII set (values hexadecimal 00 to 7F), appear only as themselves in the
  encoding. Runes with values above 7F appear as sequences of two or more bytes with values only from 80
  to FF.

  The UTF encoding of Unicode is backward compatible with ASCII: programs presented only with ASCII
  work on Plan 9 even if not written to deal with UTF, as do programs that deal with uninterpreted byte
  streams. However, programs that perform semantic processing on ASCII graphic characters must convert
  from UTF to runes in order to work properly with non-ASCII input. See rune(2).

  Letting numbers be binary, a rune x is converted to a multibyte UTF sequence as follows:

  01. x in [00000000.0bbbbbbb] ^ 0bbbbbbb
  10. x in [00000bbb.bbbbbbbb] ^ 110bbbbbb, 10bbbbbb
  11. x in [bbbbbbbb.bbbbbbbb] ^ 1110bbbbbb, 10bbbbbb, 10bbbbbb

  Conversion 01 provides a one-byte sequence that spans the ASCII character set in a compatible way. Con¬
  versions 10 and 11 represent higher-valued characters as sequences of two or three bytes with the high bit
  set. Plan 9 does not support the 4, 5, and 6 byte sequences proposed by X-Open. When there are multiple
  ways to encode a value, for example rune 0, the shortest encoding is used.

  In the inverse mapping, any sequence except those described above is incorrect and is converted to rune
  0080.

FILES
  /lib/unicode
    table of characters and descriptions, suitable for look(1).

SEE ALSO
  ascii(1), tcs(1), rune(2), keyboard(6), The Unicode Standard.
An Update on UNIX-Related Standards Activities

by Stephen R. Walll
Report Editor
USENIX Standards Watchdog Committee
<stephe@usenix.org>

Corwin’s Razor

In December, I wrote at length about a couple of fundamental problems with the structure and process of defining the POSIX family of standards. POSIX will collapse under its own structure if not rescued soon was the premise.\(^1\) It was motivated by the concern that we will lose the existing valuable model for portable applications programming, if POSIX continues along its current path. These concerns settled around test methods requirements placed on the working groups, and language independent specifications (LIS).

It provoked some people to do the net equivalent of “writing to their congressman,” and the Email addresses at the end of the article received some mail. It also provoked some discussion on the net, which is good, because this stuff is important to you if you care about writing C, Ada, and Fortran programs that are as portable as possible across the widest possible set of architectures. Your viewpoint is important! Ultimately, it was a source of a lot of debate and discussion at the IEEE POSIX meetings in January, which is responsible for developing these source-code portability standards.

One of the driving arguments in these discussions was who is the customer for this work. This sentiment is best embodied by something said by Bill Corwin, the chair of POSIX.4 (Real-time extensions), which became:

Corwin’s Razor: If they’re not willing to put their money where their mouth is, they’re not a customer.

Let’s see where it got us.

Test Method Madness Part II

I expressed a lot of concern with the creation of test methods standards. These are standards containing test assertions, based on the test methodologies outlined in POSIX.3, which could act as the basis for a test suite. I was concerned about the setting up of a directly competing body of text, used to create test suites to measure conformance of implementations of base standards, before the base standard had even received widespread implementation.

After the last editorial hit the net, I discovered something that only fueled my concerns. The test methods that were balloted as part of the XOM (Object Management) API, and X.400 API (P1224 and P1224.1) received no comment in ballot. Changes made to the test methods were done by the technical editor during ballot to keep them synchronized as best possible with changes made to the base API text. The POSIX.17 (X.500 Directory Services) test methods received very few comments in ballot, in relation to the volume of comments on the base API.

All three of these test methods standards are about to go to the IEEE Standards Board in March for final approval. One might think that their test methods weren’t read very carefully by the balloting group, which was concentrating on balloting the base API that it cared about. Would you want NIST to choose these standards as the specification of a conformance test suite?

All three of these documents had their test methods written for them by a couple of contractors in the employ of X/Open. X/Open wants these base specifications to get through the standardization process. The process demands there be test methods for the base documents to exit ballot. There are now test methods. At least X/Open was willing to put its money where its mouth is. POSIX.10 (Supercomputing Profile) has just completed its first ballot cycle, and no, they received no ballot comments on their test methods section either.

There is an example, however, of a test methods standard that is, IMHO, a well-constructed standard. POSIX.3.1 (IEEE Std 2003.1-1992) is the test methods standard for the POSIX.1 base functionality. What’s different about it:

- It lags the original standard by four years, since POSIX.1 was originally approved in 1988 (IEEE Std 1003.1-1988).
- At least four implementations of real test suites fed its creation. (The NIST PCTS, Perennial’s POSIX.1 test suite, X/Open’s VSX, built by UniSoft, and Mindcraft’s CTS.)

---

• People that wanted to have a standard for test methods to measure conformance to POSIX.1 got together (i.e., found the time and money) and did the hard work of defining, drafting, and balloting a document.

• The document was balloted separately and seriously by a group of people that seriously cared about the outcome of the standard (i.e., the implementors of the base POSIX.1 standard), as well as the people that cared about having a standard test suite.

Many working groups that have written test methods for their base functional definitions, even just partially, feel that their base specifications are stronger for the effort. They aren’t sure whether or not they’ve written good test methods. They don’t care. Their base specification is better. That’s what they came together to do.

You can’t legislate a standard. Especially from a group of volunteers. Just because one group of people want a standard “test suite,” does not mean that the group that originally got together to define and draft and ballot a standard for some base functionality wants to do the extra work of defining, drafting, and balloting a test methods standard. Corwin’s Razor.

So what happened? Most of the Steering Committee on Conformance Testing’s (SCCT’s) discussions agreed that the market will speak. If and when it cares about standards for test methods, people will find the money and energy.

A motion was made in the Sponsor Executive Committee (SEC) to remove the testing requirement from balloting base documents. It was modified to “suspend” the requirement. It passed. The SCCT will consider if there are alternatives to the current process, but until such time as they report back to the SEC, the requirements placed on the wrong group of people have been lifted.

It does not mean POSIX considers test methods standards to be bad things. POSIX.3.1 stands as an example of a well-developed test methods standard. It also does not mean POSIX doesn’t care about building good documents. Quite the contrary. A tool exists, called “writing test methods,” that working groups can use, when and where appropriate, to improve the clarity and preciseness of their base specification. It does mean that the POSIX standards working groups feel that people that want test methods standards should go to the effort of building them.

LIS – Again!

The scope of POSIX.1 says that it is a standard operating system interface to support applications portability at the source code level. It is to be used by systems implementors and application developers. This would tend to indicate it should be a readable document. It is the official “contract” with which an applications developer can call up their systems vendor and say “this is supposed to behave this way,” or vice versa. Just like the ANSI C standard. Together, these two documents define an environment in which to design more portable applications, written in C. (The Ada based POSIX.5 has similar statements in its scope.)

I raised concerns over the structure and format of the programming-language-independent specifications method of defining POSIX standards. It takes what I believe is a useful single-book, single-context format, as used in the current C-based POSIX.1 and Ada-based POSIX.5, and makes it hard to read and harder to use by creating a two-book, two-context standard.

The LIS debate is far more political and emotional. ISO is involved. The ISO Working Group responsible for bringing IEEE POSIX documents forward as international standards, WG15, requires that they be brought forward as thick, semantic LI specifications with attendant thin, syntactic language bindings.

This requirement was agreed to by the U.S. member body to WG15, which passed it on to the IEEE working groups back in 1988, which also agreed to do it. Some argue that the United States is not fulfilling its obligations if they don’t follow the LIS approach.

This is just plain wrong. The IEEE is the POSIX standards development organization. The IEEE is a “transnational” organization, open to all. While the IEEE POSIX working groups are predominantly attended by people living in the United States, a fair number of people hail from other locations, and the IEEE POSIX working groups have never not entertained a person’s point of view. They really don’t care where you live. The “U.S. member body” to WG15 is merely the administrative point between the IEEE and ISO WG15.

The IEEE then spent a lot of time and effort, first defining a methodology, then applying that methodology where they could. As with the test-methods tool, working groups discovered holes and errors in their base text as they reviewed it critically, asking the question, “How would I express this concept such that I could write the Ada equivalent of it?” Or Fortran. They discovered they can more clearly express some of the meaning in their base documents.
The people most able to do this work in the IEEE POSIX Working Group’s experience were the people in POSIX.5 (Ada) and POSIX.9 (Fortran). They did the painful exercise of critically reading the text of C-based POSIX.1, and recasting it into the words and programming semantics/syntax of their own language.

The funny thing is, they did this without the benefit of a language-independent specification of POSIX.1. The only way that the POSIX.1/LIS could be created was for the IEEE Technical Committee on Operating Systems to open the purse strings and pay a contractor to write the first drafts of an LIS of POSIX.1 and its attendant C binding.

And these other language groups, which pay money to come to IEEE POSIX meetings to do the work of building POSIX standards in their own language (which they understand well), are concerned with the current POSIX LIS methodology being used and the format of the documents it produces. I think I hear Corwin’s Razor being sharpened.

The POSIX.5 Ada Working Group built their version of POSIX.1 without the POSIX.1/LIS. They feel it would have been easier to build their document if one had existed, but they don’t know what that LIS would have looked like. They don’t particularly like the one that has been produced.

They further chose to ignore the ISO requirement of “thin” syntactic bindings, which don’t reproduce the semantic description of the “thick” base LIS document. They wanted their standard to be self-contained and readable, such that programmers would only require the one book on their desk. Their gamble failed! ISO WG15 refused to accept their standard for international standardization.

But wait! ISO WG9, the ISO Ada language group wants to fast-track the IEEE POSIX.5 standard. They feel it is a good standard. So maybe their gamble didn’t fail.

So who actually benefits by presenting the standard as an LIS? Language-bindings writers certainly. But the people who already care enough to participate seem to be doing so. It doesn’t take a huge number of people to set up a working group. There were only about twelve in the Fortran group (POSIX.9).

So what happened at the SEC? A motion came forward to remove the requirement for the current LIS method of defining POSIX standards. It was massaged into something considerably more diplomatic, which passed, creating an ad hoc committee to investigate the problem in detail, and without particular restrictions, and report back at the April 1993 meetings.

This is a good thing. To change the direction of POSIX at this point is not a trivial task to be taken lightly, nor decided too quickly. There will be ramifications no matter what the outcome of the report from the ad hoc.

I chair the ad hoc committee. If I am going to stir up all this fuss, then I am going to be made accountable for it. I am interested in your thoughts or concerns, so by all means Email me.

There was another wonderful quote that came out at the January meetings, that essentially said: Standards organizations that choose to make themselves irrelevant deserve what they get.

This was actually made in reference to a completely different problem, but I believe it is very appropriate here. If we make these standards unusable, they won’t be used. We will lose the “contract” for a portable programming model between applications developers and systems implementors.

I am repeating the list of Email addresses from the end of “POSIX – Caving in Under Its Own Weight.” I believe it is still important that you make your concerns known to the people that can actually make the decision about this.
<table>
<thead>
<tr>
<th>Position</th>
<th>Name</th>
<th>E-mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE Concerns</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chair SEC</td>
<td>Jim Isaak</td>
<td><a href="mailto:isaak@decvax.dec.com">isaak@decvax.dec.com</a></td>
</tr>
<tr>
<td>Vice Chair Interpretations</td>
<td>Andrew Twigger</td>
<td><a href="mailto:att@root.co.uk">att@root.co.uk</a></td>
</tr>
<tr>
<td>Vice Chair Balloting</td>
<td>Lorraine Kevra</td>
<td><a href="mailto:l.kevra@att.com">l.kevra@att.com</a></td>
</tr>
<tr>
<td>Chair Steering Comm on</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conf Testing</td>
<td>Roger Martin</td>
<td><a href="mailto:rmartin@swe.ncsl.nist.gov">rmartin@swe.ncsl.nist.gov</a></td>
</tr>
<tr>
<td>Chair Project Management Comm</td>
<td>Shane McCarron</td>
<td><a href="mailto:s.mccarron@ui.org">s.mccarron@ui.org</a></td>
</tr>
<tr>
<td>Chair POSIX.1</td>
<td>Paul Rabin</td>
<td><a href="mailto:rabin@osf.org">rabin@osf.org</a></td>
</tr>
<tr>
<td>Chair POSIX.2</td>
<td>Hal Jespersen</td>
<td><a href="mailto:hj@posix.com">hj@posix.com</a></td>
</tr>
<tr>
<td>Chair POSIX.3</td>
<td>Lowell Johnson</td>
<td><a href="mailto:31gj@rsvl.unisys.com">31gj@rsvl.unisys.com</a></td>
</tr>
<tr>
<td>Chair POSIX.4</td>
<td>Bill Corwin</td>
<td><a href="mailto:wmc@littlei.intel.com">wmc@littlei.intel.com</a></td>
</tr>
<tr>
<td>Chair POSIX.5</td>
<td>Jim Lonjers</td>
<td><a href="mailto:lonjers@prc.unisys.com">lonjers@prc.unisys.com</a></td>
</tr>
<tr>
<td>Chair POSIX.6</td>
<td>Dennis Steinauer</td>
<td><a href="mailto:dsteinauer@nist.gov">dsteinauer@nist.gov</a></td>
</tr>
<tr>
<td>Chair POSIX.7</td>
<td>Martin Kirk</td>
<td><a href="mailto:m.kirk@xopen.co.uk">m.kirk@xopen.co.uk</a></td>
</tr>
<tr>
<td>Chair POSIX.8</td>
<td>Jason Zions</td>
<td><a href="mailto:jason@cnl.hp.com">jason@cnl.hp.com</a></td>
</tr>
<tr>
<td>Chair POSIX.9</td>
<td>Michael Hannah</td>
<td><a href="mailto:mjhanna@sandia.gov">mjhanna@sandia.gov</a></td>
</tr>
<tr>
<td>Chair POSIX.12</td>
<td>Bob Durst</td>
<td><a href="mailto:durst@mitre.org">durst@mitre.org</a></td>
</tr>
<tr>
<td>USENIX Institutional Rep</td>
<td>Jeff Haemer</td>
<td><a href="mailto:jh@usenix.org">jh@usenix.org</a></td>
</tr>
<tr>
<td>EuroOpen IR</td>
<td>Stephen Walli</td>
<td><a href="mailto:stephe@mks.com">stephe@mks.com</a></td>
</tr>
<tr>
<td>DECUS IR</td>
<td>Loren Buhle</td>
<td><a href="mailto:buhle@xrt.upenn.edu">buhle@xrt.upenn.edu</a></td>
</tr>
<tr>
<td>OSF IR</td>
<td>John Morris</td>
<td><a href="mailto:jsm@osf.org">jsm@osf.org</a></td>
</tr>
<tr>
<td>UNIX International IR</td>
<td>Shane McCarron</td>
<td><a href="mailto:s.mccarron@ui.org">s.mccarron@ui.org</a></td>
</tr>
<tr>
<td>X/Open IR</td>
<td>Derek Kaufman</td>
<td><a href="mailto:d.kaufman@xopen.co.uk">d.kaufman@xopen.co.uk</a></td>
</tr>
<tr>
<td>WG15 Concerns</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Convener WG15</td>
<td>Jim Isaak</td>
<td><a href="mailto:isaak@decvax.dec.com">isaak@decvax.dec.com</a></td>
</tr>
<tr>
<td>US Head of Delegation</td>
<td>John Hill</td>
<td><a href="mailto:hill@prc.unisys.com">hill@prc.unisys.com</a></td>
</tr>
<tr>
<td>Canadian HoD</td>
<td>Arnie Powell</td>
<td><a href="mailto:arniep@canvm2.vnet.ibm.com">arniep@canvm2.vnet.ibm.com</a></td>
</tr>
<tr>
<td>UK HoD</td>
<td>David Cannon</td>
<td><a href="mailto:cannon@exeter.ac.uk">cannon@exeter.ac.uk</a></td>
</tr>
<tr>
<td>German HoD</td>
<td>Ron Elliot</td>
<td>elliott%<a href="mailto:aixsm@uunet.uu.net">aixsm@uunet.uu.net</a></td>
</tr>
<tr>
<td>Dutch HoD</td>
<td>Herman Wegenaar</td>
<td>(phone: +31 50 637052)</td>
</tr>
<tr>
<td>Japanese HoD</td>
<td>Nobuo Saito</td>
<td><a href="mailto:ns@slab.sfc.keio.ac.jp">ns@slab.sfc.keio.ac.jp</a></td>
</tr>
<tr>
<td>French HoD</td>
<td>Jean-Michel Cornu</td>
<td><a href="mailto:jean-michel.cornu@afuu.fr">jean-michel.cornu@afuu.fr</a></td>
</tr>
<tr>
<td>Danish HoD</td>
<td>Keld Simenson</td>
<td><a href="mailto:keld@dkuug.dk">keld@dkuug.dk</a></td>
</tr>
<tr>
<td>New Zealand HoD</td>
<td>Keith Hopper</td>
<td><a href="mailto:kb@waikato.ac.nz">kb@waikato.ac.nz</a></td>
</tr>
</tbody>
</table>
Report on POSIX.0: The POSIX Guide
Kevin Lewis <klewis@gucci.ent.dec.com> reports on the January 11-15, 1993 meeting in New Orleans, LA:

First off, let me say that this will be a relatively short report given that the group’s exclusive activity currently is ballot resolution. To fulfill my promise made from last meeting, I have provided a more detailed balloting profile. It is as follows:

<table>
<thead>
<tr>
<th>Vote</th>
<th>Ballots</th>
</tr>
</thead>
<tbody>
<tr>
<td>affirmative</td>
<td>28</td>
</tr>
<tr>
<td>negative</td>
<td>30</td>
</tr>
<tr>
<td>abstentions</td>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>objections</td>
<td>494</td>
</tr>
<tr>
<td>comments</td>
<td>640</td>
</tr>
<tr>
<td>total</td>
<td>1134</td>
</tr>
</tbody>
</table>

There were 69 ballots returned (85%) of which 48% were affirmative.

The January meeting was a rolled-up sleeves session where the group focused totally on ballot resolution. It was a bit of a struggle because we transitioned some section leaders. New people took over for some, and others were absent.

It became apparent by week’s end that we wouldn’t resolve all the ballots. It also became apparent that, as this process continues, a more authoritarian role on the part of the section leaders and the chair will be necessary to expedite ballot resolution.

The ballot-resolution group agreed that section leaders should use electronic means to survey the group on issues where they feel such help is needed.

The next deadline will be March 8, at which time all ballot resolutions will be submitted to the ballot coordinator. The ballot coordinator will work with the technical editor to prepare an interim draft for the April meeting as a last look by the whole group before it goes out for recirculation.

The group attempted (as it is prone to do at times) to step back onto some old ground, re-opening discussions that had been resolved or falling down potential rat holes. On these occasions, the chair had to bring the the group back in to the process of resolving the ballot at hand. There is a balancing act that the group must maintain. On one hand, there is the desire to ensure that the document does not go out with any major defects. On the other hand, we need to keep the resolution process moving forward. This sometimes compels the group to open up broader discussions than are really necessary.

The group consensus is to err on the side expediting the process in order to get this work into the hands of the balloting group, which has been asking for it.

Report on POSIX.2: Shell and Utilities
David Rowley <david@mks.com> reports on the January 11-15 meeting in New Orleans, LA:

A Brief Update

The POSIX.2 standard (IEEE Std 1003.2-1992) should be available from the IEEE in April as a 2-volume set for $95. The standard consists of both the “Dot 2 Classic” and “Dot 2a” components, previously balloted as separate standards. The IEEE Standard (based on Draft 12 from the ballot group) is identical (at least from a technical standpoint) to ISO/IEC Draft International Standard 9945-2:1992.

NIST expects to issue the new draft FIPS (Federal Information Processing Standard) for POSIX.2 early in April, with the final version expected by late 1993.

POSIX.2b work continues, now on draft 5. The group is still wrestling with the ISO 1001 tape format for PAX.

Test method development for the base POSIX.2 standard nears completion, and a full recirculation of the P2003.2 document is expected by early summer.

X/Open has awarded the role of integrator for the combined POSIX.2 / XPG4 Commands and Utilities test suite project to a joint venture between BSI (British Standards Institute) and Mindcraft, Inc. (Palo Alto, CA). The suite is expected to be available early in 1994.

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 com-
mands, 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 project that 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.

Report on POSIX.5: Ada Bindings to POSIX.1

Del Swanson <dswanson@mhs.sp.paramax.com> reports on the January 11-15, 1993 meeting in New Orleans, LA:

The POSIX.5 working group has been working to produce Ada language bindings to POSIX standards. The Ada binding for POSIX.1, IEEE Std 1003.5-1992 (aka POSIX.5), has now been published as an IEEE standard. Suitable kudos were spread around to the contributors at the January meeting in New Orleans. The next target is the development of bindings to the Real-Time Extensions standards being developed by the POSIX.4 group.

The binding to POSIX.4 is relatively straightforward. A draft thin binding to POSIX.4 was prepared by one of our members on contract to the U.S. government. This draft has now been updated by the group, and massaged into IEEE format. This document, POSIX.20 (draft 1), was circulated for mock ballot in December, with comments due in by February 4. Our goal is to go to real ballot soon after the April meetings, and have POSIX.20 approved as a standard hard on the heels of POSIX.4 LIS (Programming-Language-Independent Specification).

Meanwhile, work has begun concurrently on the binding to POSIX.4a (threads extensions). An initial draft has been prepared, and was debated at the January meeting. Significant changes to it are now expected to be put on hold until the next version of POSIX.4a appears. The POSIX.5 group met with the POSIX.4 group in January to get an update on the status of the threads work.

Orthogonal to this update, some members of the POSIX.5 group are becoming concerned about the relationship of the threads interface and the updates to the Ada language standard that is commonly called Ada 9x. Some significant changes and enhancements are expected in the tasking model for Ada 9x, and in some respects, they have an adverse impact on the ability to implement an Ada runtime library using POSIX threads. These concerns are being provided to the POSIX.4 group, for consideration in the ballot resolution process.

In January, we also met with a delegation of the group that is formulating the POSIX.1 LIS. Several members of the POSIX.5 group had objected to a few points in the ballot. In the discussion, general agreement was reached on issues of naming, I/O formulations, and implications of concurrency within POSIX processes. Signal concerns were also discussed, but it remains to be seen whether mutually agreeable formulations result.

The core of the issue is that Ada runtime libraries require the exclusive use of a few of the signals to implement Ada scheduling and exception delivery. The Ada binding to POSIX.1 specifies this, and it is our perspective that the LIS should allow such exclusivity.

Some members of the group have been facing problems with the IEEE standards office related to the copyright of the Ada binding. We have also been receiving reports from several others of similar problems. The copyright, of course, states that none of the material in the standard may be reproduced in any fashion without the permission of the IEEE.

Well, how does one compile an Ada package specification (or a C header file, for that matter), which happens to be part of the standard, without copying it, and reproducing it in the file system of a computer? How does one introduce the implementation-defined values in appropriate places? How does one inform one's users about the values so defined? How does one provide access to these specifications, so that calls to the interfaces can be compiled in application code?

At first, a couple of people applied for, and received, official limited permission at least to make compilable copies for development purposes. Our group was concerned about the reticence, the bother of individual application, and the implications for the ease of use of the standard. Therefore our chair approached the IEEE, explained the situation, and appeared to have reached an amicable agreement.

The IEEE defined a policy of approval to copy the specifications for such use, with requests and automatic approval exchanged by email. Copies
of the correspondence were to be sent to a member of POSIX.5 to monitor the process. To date, upwards of 20 requests have been monitored, but no approval responses have been forthcoming. We have not heard of similar difficulties for other language bindings. Obviously, this is a serious problem, and will be addressed further at the next meeting.

On a more technical level, the April meeting will be spent resolving comments to the mock ballot; defining strategy on the coordination of the POSIX.5 standard with the revised POSIX.1 and the proposed POSIX.20 (Ada binding to the real time extensions), and addressing the issues of interpretations that have been raised with POSIX.5.

Report on POSIX.7: System Administration

Bob Robillard <duke@cc.bellcore.com> reports on the January 11-15, 1993 meeting in New Orleans, LA:

Introduction

Three of the POSIX.7.1 System Administration small groups met during the week:

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

There were also several plenary meetings of the group, and issues were discussed that cut across subgroups.

POSIX.7 — Overall

The first issue discussed by POSIX.7 as a whole was the question of Test Assertions (TA) and Language Independence (LIS). I suspect this issue is discussed at length in another snitch report, so I'll just give POSIX.7's angle. The group had discussed this in the past, and was clearly in agreement with Stephen Walk's movement to drop these requirements. We wrote a letter to the SEC stating our position and POSIX.6 (Security Extensions), POSIX.11 (TP Profile), and POSIX.14 (Multiprocessor Profile) co-signed it.

Since the Test Assertion requirement was suspended by the SEC and the Language Independence requirement is under attack, the group has decided to limit the amount of time spent on these to a bare minimum. If an LIS is still required by the time the Print Group goes to ballot in May, a rough draft of one will be submitted with the real, C language draft.

The second cross-group issue debated was the question of using common command line options for "extended options" (i.e., options that take more than a simple switch to specify). Both the Printing and Software Management command line interfaces (CLI) describe similar files that can contain extended options for commands. It was decided to use the same option letter for these "extended options files" (-X).

Both CLIs also allow these extended options to be passed in a quoted string on the command line, and there was an agreement to use the same letter for this as well (-x). Unfortunately, the groups couldn't agree upon a common syntax for the content of the files and strings.

The last POSIX.7-wide issue was the question of distinguished names. These are names of entities in the network, e.g., machines or print daemons. It was decided that the POSIX.7.X drafts would require that implementations accept Internet style names and can allow any other style they want. The suggestion of requiring X.500 style names (/co=usa/ org=dec/ host=foobar/printer_server) was voted down, mainly because it's not widely used. In fact, even the POSIX X.500 API doesn't use it directly! That API requires applications to parse the name given on the line and fill a C structure, so it is just as happy with Internet names as with the "/" style names.

POSIX.7.1 — Print Administration

The first issue the Printing Group dealt with was the forthcoming new edition of the ISO Document Printing Application (DPA) Draft. The POSIX printing document is based on the ISO DPA. A new version of the ISO DPA is due out in May or June, and the printing group had to decide how to deal with the new document.

One of the members of POSIX.7.1 is also a member of the ISO DPA committee. He went over the changes in the next version, and the group decided that their effects on the POSIX document were small enough that they could be incorporated into the draft during the ballot process.

The group next discussed a number of changes proposed by the Open Software Foundation. None of these were serious changes, and most were adopted. In addition, the section describing the Name Service was extensively rewritten and the programming examples in the rationale for the API were brought up to date.

The printing group started the process of forming a ballot group. Although the dates for ballot are not final, the POSIX.7.1 ballot will probably run from May 10 to July 16. To get on the ballot group, contact:

IEEE Standards Office
Attn: Anna Kaczmarek
PO Box 1331
Piscataway, NJ 08855-1331 USA
Tell her you want to join the "TCOS Standards Subcommittee." [TCOS, the Technical Committee on Operating Systems, is no longer the parent of the POSIX standards subcommittee. TCOS-SS has become PASC, the Portable Applications Standards Committee. – Ed.] Give your IEEE or IEEE Computer Society Number, if you’ve got one. Only IEEE or Computer Society members are eligible balloters on IEEE proposed standards; nonmembers can participate as Parties of Interest, which means they can vote and object, but their vote doesn’t count in the final tally.

Alternatively, you can contact Bob Robillard (908/644-2249, <duke@cc.bellcore.com>).

**POSIX.7.2 — Software Management**

The Software Group had a set of written comments from a number of the members, and they spent the week going through them, improving the draft for their mock ballot. The mock ballot will be conducted from March 1 to March 31. To join in this ballot, contact Jay Ashford, <ashford@austin.ibm.com>, 512/838-3402.

Many of the comments reviewed dealt with cleaning up the command line interface (e.g., determining options names, and so on). There was a long debate on the value of allowing multiple “MIBs” or databases of installed software packages. In the end, the group decided to permit this.

Other details were worked out, such as the use of a name server, the media format (the POSIX pax format was chosen), and use of environment variables. The idea of making everything in the standard optional except for the distribution format was discussed. This would ensure portability of distributions, but wouldn’t do anything toward promoting a common command set. The decision was reached to make the entire draft required, at least for the present.

**POSIX.7.3 — User and Group Management**

The User/Group subgroup made some concrete progress toward a real draft. After reviewing POSIX.1 and FOSIX.2 for any user/group items and meeting with POSIX.6 to learn their concerns, they wrote a scope and picked three base documents to merge into a draft:

- USL’s System V Interface Definition, 3rd Edition (with additions from the new Distributed Manager User management product)
- OSF’s DCE user management
- SCO’s user management product

The Scope and Base document list was given to POSIX.7’s PMC Mentor, who agreed that they would make a good start. [The POSIX Project Management Subcommittee (PMC) assigns someone to act as a mentor or guide to each project, who is supposed to be a shield for some of the procedural work, and help the project keep on track.—Ed.]

POSIX.7.3 will be creating a Project Authorization Request (i.e., permission to start a document) in April. The PMC Mentor is happy with their proposal, and intends to recommend granting approval. In anticipation of that, they will attempt to have Draft 1 of POSIX.7.3 in the mailing before the April meeting.

While it is somewhat premature to work out the details of the command line, POSIX.7.3 contributed to the joint debate on the “extended options” (i.e., -x and -X) and intends to follow the lead of the other two groups.

In addition, they presented the idea of another common option: a “template” object, used as a template from which to create real objects. For example, a typical-user template would have all the information necessary to set up a new typical user, and could be specified in the useradd command (this is similar to inheritance in the object oriented world). There was agreement from POSIX.7.1 and POSIX.7.2 that this could be useful, and will be investigated further.

**Report on POSIX.18: POSIX Platform Environment Profile**

Paul Borman <prb@cray.com> reports on the January 11-15, 1993 meeting in New Orleans, LA:


The title says it all. What is a POSIX Standardized Profile? What does USI-P001 mean? If you can answer these questions – We Want You! Unfortunately, this confusion is carried through the whole draft.

Due to problems of redundancy and obfuscation, the working group unmercifully hacked away at the draft with an axe in the previous meeting. This meeting we took out our Bowie knives to whittle it down further still.

The three major issues discussed were the prose, what was it for, and what new normative text or changes to the normative text should be made.

This discussion of the prose centered around the large amounts of redundant and apparently meaningless text in the draft. Was it boiler plate?
Was it just the previous editor? Did it simply come from Planet X in the middle of the night? Extraterrestrial or not, either the prose was simply removed or we reworded it to be more easily understood.

First on the list of things to be clarified was the introduction, which was determined to be mostly redundant or irrelevant. We did decide to reword it to indicate that POSIX.18 describes UNIX Classic or Version 7, for those who remember it. The profile still will not define administrative interfaces, or even a way to login.

We did lobby the POSIX.1a working group to modify a couple of interfaces to bring them in line with FIPS 151-2. [FIPS 151-2 is the updated NIST specification of the POSIX.1 standard, used in U.S. government procurements where POSIX-like functionality is required. — Ed.] We hope POSIX.18 will mirror this new FIPS. These modifications were:

• When read() or write() is interrupted by a signal, after having read/written any data, then they will return the byte count instead of -1.

• That the group-ID of a file at creation time is that of the directory in which it is created, and not the effective group-ID of the process.

We introduced text in POSIX.18 that requires that CS7, CS8, CSSTOPB, PARODD, and PARENB be supported from the POSIX.1 General Terminal Interfaces section.

We are not clear exactly what NIST was trying to accomplish by this and any comments would be appreciated.

There were several parts of the document that existed to fulfill TR10000 requirements, but as TR10000 is changing, much of this was removed. [TR10000 is the ISO technical report, defined originally in the OSI profile world, and now making itself felt in the POSIX profile space. — Ed.] We are going to lobby for the new TR10000 to require less and make it easier to understand.

We restructured the two or three pages of real normative text in the document in line with our decision in the last meeting to require the C language.

Due to a new SEC ruling, we plan to remove the current, inadequate test assertions in the document, and concentrate on the normative text.

Our major additions to the normative text, aside from the FIPS 151-2 item mentioned earlier, were coherency statements. We have required, for example, that all the base standards that are pointed to by this profile must be implemented with the the same file-system name space and use a consistent byte size.

We also mandated that text files would be usable between all the different base standards and that text files can be used to contain source code that the compilers can compile. Without these sorts of statements it would have been technically possible to have a conforming system in which vi was not capable of creating a file that the C compiler could compile!

Other things were that the shell could execute a program built with the compilers and that the compiler would allow use of the POSIX.1 functions. Pretty straightforward and obvious stuff, but that is the sort of thing a profile must point out to make itself useful.

Overall, I feel that the POSIX.18 draft made a lot of forward progress, but because it now references POSIX.1a it cannot go to ballot. We also feel we need to do a bit more work cleaning up the wording of the draft (and come to grips with what NIST is really asking in FIPS 151-2).

Please note that POSIX.18 is the profile that will more than likely define the basics of a time-sharing UN*X system in the future. If you are concerned about this, you might want to show up at our next meeting, and you will certainly want to join the balloting group.

Report on ANSI X3B11.1: WORM File Systems
Andrew Hume <andrew@research.att.com> reports on the Dec. 14-16, 1992 meeting in Orlando, FL

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 two X3B11.1 meetings. In brief, our ECMA standard has been published, we have entered the fast-track process, and are now DIS 13346!

ECMA -167

I won't describe ECMA-167 again; if you want the gory details, see my last snitch reports. At the time of my last report, the ECMA General Assembly had approved ECMA-167 as a standard and "all" we had to do was publish it. This was not an entirely smooth process, but it could have been worse.

The source of the draft was a weird form of text that, after processing by several awk and sed scripts, became more or less normal troff -ms input. The ECMA office uses a popular PC pub-
lishing package. The conversion was mostly done mechanically (using RTF as the intermediate form) with our chair Ed Beshore doing the final pass by hand on his PC before sending floppies off to Geneva. A mere three galley proofs later, I (as technical editor) approved the current draft. Proofing galleys is about as tedious as it sounds. (It's good to do while watching Sunday afternoon football.) I was ably assisted by Howard Kaikow, now no longer at DEC. The draft was much improved stylistically by this process, although I personally find the ECMA house format to be visually unappealing.

International Activity

There is substantial 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 were very pleased to learn that ECMA-167 is now DIS 13346. The six-month ballot period will end July 28, 1993 and the special working group meeting that addresses the ballot responses has been tentatively scheduled for October 13-15, 1993 in Geneva, Switzerland. The end is definitely in sight.

The other activity going on in SC15 is work on a reference model for information interchange between open systems by interchangeable storage media. This is similar to the OSI reference model; in fact, rather too similar in my mind. Although reference models can be astonishingly boring, a good one would have helped the development of our standard a little, and a bad one can easily hinder the development of good standards. The current draft of the reference model represents early work and is being commented on by interested parties in our committee and by an ad hoc group in X3B8.

Future Activity

The committee's focus is now split among three areas. The first area is preparing for voting on DIS 13346. This is fairly routine but intricate because of procedural rules and delays within the U.S.; documents have to get passed from ISO to ANSI to X3 to X3B11 and finally to us. We vote on a recommendation for the U.S.'s vote, and then that goes back up the chain. The complications involve meeting schedules, voting deadlines and making sure no one inadvertently says no.

The second area is implementing ECMA-167. I know of five implementation efforts; one commercial implementation is beta testing with customers. As a means of verifying our understanding of the standard, and as a way of improving the level of interchange, Hewlett-Packard organized a meeting on conformance testing for ECMA-167 in February in Fort Collins, CO. This was surprisingly popular, with about 30 companies attending. In brief, the meeting agreed to work on the areas of conformance testing, and the details of how to translate between conforming media and various operating systems' interfaces.

The third area is addressing work for future standardization. This includes specific proposals for issues like compression, which ECMA-167 supports in a generic way, and proposals for niche targets with specific reliability and performance goals. This work is parallel to, and asynchronous with, the progress of DIS 13346. If anyone has specific proposals for things not adequately addressed in ECMA-167, they are invited to make them known to X3B11.1. (If you can't or don't want to attend meetings, I may be willing to be an advocate for you!) Contact Ed Beshore for meeting details.

Electronic Distribution of Standards/Drafts Several

X3B11.1 documents have been available electronically by both ftp and email (netlib) from <research.att.com>. (For ftp, login as netlib.) For details, get index from research/memo. The main documents are:

- The standard itself (121 pages including TOC and index). (This is the actual standard as published; ECMA has approved its electronic distribution.)
- 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

If you would like more details on X3B11.1's work, you should contact either me at <andrew@research.att.com>, 908/582-6262 or the committee chair, Ed Beshore at <edb@hpgla.gr.hp.com>, 303/350-4826). The next two meetings are in Tucson in mid-March and Long Island in mid-July. Anyone interested in attending should contact Ed Beshore.
This month I want to talk about only three books. If I stated that they had occupied all of my reading time since Thanksgiving, I'd be lying; they've occupied an awful lot of it (as Santa Claus brought me a batch of history books, too, I've been alternating my time). But the three volumes at hand strike me as involving the most important questions of the information systems industry today: standards and interoperability, on the one hand, and communications, on the other.

Open, Sesame!

First, I'd like to look at Quarterman and Wilhelm's UNIX, POSIX, and Open Systems. This is a very good piece of work (it even mentions me, for instance). Whether you hate it or not, you are going to be involved more and more as time goes on in the world of standards. If you've been involved in C, you know that the K&R version and the ANSI version are not identical; FORTRAN is now a standard; and soon several more parts of UNIX will be standards, too. This may mean that the battle is over: after all, we have an international standard for 8" floppies, and how many of us can remember them? Arnold Toynbee has reflected that standardization may be the har- binger of decline in civilizations.

Q&W have organized the book using a "puzzle" metaphor, and the volume is divided into four parts: Context, Cutters (the people and organizations who cut the pieces of the puzzle), Pieces and Patterns (base standards, extensions; networking and Internationalization), and Puzzles (profiles). There are also three forewords, a preface, two appendices, and lots more.

Q&W have been active in the standards world for many years and clearly know everyone and have read everything. From their outline of the basic area of the problems inherent in both open systems as a concept and standardization as an international political can of worms, their work will enlighten and inform.

Their outline (pp. 74-76) of the various industry organizations, user groups (EurOpen, JUS, UniForum / /usr/group), and USENIX are the four "traditional" ones, and the relation between standards bodies is excellent. It leads into chap- ters on Formal Standards Bodies and Industry Organizations and User Groups.

In the Pieces and Patterns chapters, I especially liked chapter 8, on OSI and TCP/IP protocol suites. The last section gives a more-than-adequate explanation of TCOS profiles.

Appendix A, Resources, is just superb. If you have any questions on how to get standards, draft standards, industrial consortia specs and white papers, documents from national groups, books on standards, or work in the periodical literature, here it is, in barely over 20 pages.

My compliments to Q&W, but a real tip of the hat to Addison Wesley, for a brilliant beginning to a new "UNIX and Open Systems Series."

TCP/IP, Vol. III

Here comes Doug Comer again! A third volume to the Internetworking with TCP/IP series. This one written with Comer's Purdue colleague David L. Stevens.

If you are unfamiliar with vol.1 (Principles, Protocols, and Architectures) and vol. 2 (Design, Implementation, and Internals), I guess you can begin here: Client-Server Programming and Applications. This volume concerns itself with the design and implementation of new applications for the Internet infrastructure.

This volume states that it is the "BSD socket version." If USL/Novell succeeds in killing off BSD, only the academic and research sites will be employing this, I guess. But as a loyal and true BSD fan, I'm rooting for BSDI and the CSRG.

The real advantage of this volume (and its predecessors) lies in the fact that not only is each "how" that is considered (e.g., Multiprotocol Servers or TELNET clients) fully explained, but also the "whys" of the matter are gone into.

Appendix 1 gives the man pages for the socket system calls and library routines.

Another good volume.

The Internet, Again

Dan Lynch, the founding president of InterOp, and Marshall Rose, author of the four volumes of his internetworking trilogy, have pulled together a massive volume written by 23 contributors (including themselves) on the history of the Internet, its present status, technology and tools, and its future. This is a wonderful, unreadable, indigestible volume.
It took me over six weeks to read the nearly-800 pages because there is just too much here for a mere mortal to absorb quickly. And there is no space here for me to even list each contribution and its contents. As a consequence, I’d like to remark on just a few things.

The best. It’s a bit embarrassing, but perhaps the best and most useful contribution is John S. Quarterman’s “Annotated Bibliography” (pp. 751-775). Between this and the Appendix in Q&W, JSQ has performed extraordinary services to the computing community. Now there is no reason left to yell to the next cubicle or office to find out how to locate information where either the Internet or standards and open systems are concerned. JSQ gets the gold star award for this.

Chapters 1 (on the history of the Internet, by Dan Lynch) and 2 (on the expansion and globalization of the Internet, by Barry Leiner) are both interesting and (perhaps) the most readable. Parts II (pp. 77-466) and III (pp. 467-704) are useful and, as might be expected, hard-to-read and tough-to-digest. As each chapter is largely independent, I’d recommend reading what you are interested in (or what you need), rather than sitting down with the book.

Part IV (“Directions,” pp. 705-749) may be the most disappointing one. Though A. Lyman Chapin’s chapter on the giga-node Internet and Charles E. Catlett’s on evolution and future directions are interesting, they are constrained. For example, I looked in vain in the index for an entry on commercialization of the net. Looking for, say, UUNET or AlterNet or CompuServe was equally futile. PSI gets a mention, but only as “a leader in developing dial-in services” (p. 568) in a section on network services and support.

As I see the global Matrix (nod here to JSQ, again) as integrating the Internet and commercial services in the relatively near future, these are serious omissions. Furthermore, surely the size of the US deficit will constrain government expenditure on the Internet quite severely over the next lustrum. This will push us ever-further towards genuine commercialization. What we need now – and it doesn’t exist among the flurry of Internet books of this past 18 months – is a serious analysis of the economics of the Matrix.

These last two paragraphs aside, this is a real contribution to the literature.

If you buy any or all of these books, lock them in your desk drawer. I’ve lost too many books over the past four years to leave anything important on the shelves in my office. Anyway, you’ll have invested nearly $150.


Yes, this is a review. But I didn’t think it really fitted in with my Bookworm column. Rob Kolstad (have you a Fan Club pin?) agreed. So here we are.

What do you do with a t-shirt, a 1200-page book, and a CD-ROM? Clearly, wear one, read another, and ...

So I’m sitting here in frigid Boston, wearing a UNIX Power Tools t-shirt and writing about this fascinating joint-venture between O’Reilly & Associates and Bantam Electronic Publishing. The book, over 1150 pages, weighs in at around a ton. It is chock-full of useful tips, information, and (occasional) glimpses of a human being behind the text.

UNIX Power Tools is not designed to be read through; it is dedicated to the flitter, the reader who’ll hop through the book pursuing the topic at hand and, perhaps, alighting on a lily-pad that was sighted serendipitously.

The CD-ROM, which was put together by Ready-to-Run Software, is in ISO 9660 format (“High Sierra”), and there are binaries for Sun4 SunOS 4.1.1, Sun3 SunOS 4.1.1, DECstation Ultrix 4.1, IBM RS/6000 AIX 3.2, HP 700 HP-UX 8.07, SCO XENIX 2.3.2, and SCO UNIX 3.2x. If you don’t have one of these platforms, you can still build from source – there are scripts to do this with minimal effort. The install instructions are clear and well-written. Don’t have a CD drive? You can get the equivalent on 3.5” floppies; 5.25” floppies; QIC-150; QUI-24; 8mm; 4mm; and TK50. As “What’s on the disk” runs from page 1022 to 1040, I won’t reproduce it here. But it goes from emacs-ml through agrep to GNU emacs (18.58) to perl (4.035) [two rahs to Larry Wall] through zipcode to zview.

The book itself has a slew of neat features: article numbers, cross-references, summaries, footers, attributions (in the form of authors’ initials), and several lovely icons: a wood screw (= “Be careful ... or you might get screwed”), a bomb with fuse lit (= “a cross-referenced screw”), and a CD-ROM (= on the disk; to install use “the name listed under the icon”).

Bouquets to everyone involved in this: three major authors, the staff at ORA, and the 35 contributors.

Check it out! It’s a bargain.


Understanding DCE


Reviewed by: Choong Huei Seow
Synesort Inc.
<choong@panix.com>

The Distributed Computing Environment (DCE) is a software system designed and implemented by the Open Software Foundation (OSF) as a solution to providing a homogeneous distributed computing environment.

Anyone new to DCE or distributed computing in general should start off with reading Understanding DCE. This book is basically organized into three major sections. Part One gives a very informative discussion of the architecture and principal components of DCE. Examples of the DCE components are the Cell Directory Service, Remote Procedure Calls (RPCs), and the Security Service.

Each chapter in Part One covers a specific component of DCE. The chapter first provides a review of the current problems and issues pertaining to that particular area of distributed computing, and how that DCE component addresses such problems. Examples and informative diagrams are given throughout the entire chapter to help the user understand the concepts and functionality of that particular DCE component. Overall, each chapter shows how each particular component contributes to
concepts and functionality of that particular DCE component. Overall, each chapter shows how each particular component contributes to the overall functionality and architecture of DCE.

As an example to describe the strong point of the book, the following is a summary of the chapter (Chapter 3) which describes the RPC component of DCE. The chapter begins with an overview of the issue of the communication task requirements of distributed applications. This is followed by a description of local procedure calls (LPC), and how remote procedure calls extend the same concept of LPCs. Informative figures are given to show the various software abstraction layers present on RPC models and LPC models. The reader is then presented with a conceptual step-by-step “execution” of an RPC. The explanation of each step is complemented by flowcharts showing the logical flow of control between components. Other chapters in Part One of the book are:

2) Cells: The Domain of the Distributed Environment – the first chapter which covers an architectural component of DCE, the Cell domain.

4) Threads: Improving Program Performance – a discussion of how the implementation of DCE threads improve application performance.

5) DCE Security Service – this chapter covers one of the essential services of DCE which implements security and resource protection services, such as data encryption.

6) DCE Directory Service – explains the workings of the Directory Service and its function as a clearinghouse between application processes and network resources.

7) DCE Time Service – the chapter explains the implementation of network time synchronization, a crucial service needed to provide consistency in file and process sharing.

8) DCE Distributed File Service – this chapter in Part One covers the most important component of the DCE architecture. This component implements a single integrated file system that is shared among all DCE users and host computers in a DCE cell. The chapter discusses the advantages of DFS, and the facilities which it provides to users and applications.

9) Writing DCE applications – the last chapter in Part One covers the entire process of writing a DCE application, in the scope of the flow of applications while requesting network services, etc.

Part 2 of Understanding DCE takes the user away from the technical discussions of DCE components and implementations. This section is geared to the administrative and planning aspects of a DCE environment. The first chapter in this section (Chapter 10) is brief, basically giving the system administrator a feeling for the planning involved in configuring and managing DCE. The issues in planning for a Cell (a DCE domain of computer systems) are covered by Chapter 11. Overall, Part 2 of this book is useful for people who are planning to install DCE on their distributed computer systems.

Several appendices are included in Part 3 of the book. Appendix A includes source code listings of a few simple (and functional) programs. These sample DCE applications are intended to familiarize the reader with the usage of an RPC, thread implementation, and security services. Appendix B is a question-and-answer section that covers the most common questions asked by end users, systems administrators, and application programmers. Appendices C and D contain listings of Coordinated Universal Time (UTC) providers, and contacts for obtaining GDS (Global Directory Service) and DNS (Domain Name Service) cell names.

Understanding DCE is an excellent and comprehensive book on the subject of DCE. The book is very well written, making this technical book relatively easy to read. I would recommend it to anyone who is interested in DCE or who would like to learn about the issues pertaining to distributed computing environments.

Guide to Writing DCE Applications


Reviewed by: Choong Huei Seow
Syncoast Inc.
<choong@panix.com>

Guide to Writing DEC Applications is a book targeted at programmers who develop application programs with the DCE RPC in order to use the remote computing resources in the network. In particular, the material presented covers the development of a client application, an interface, and a server. This book is intended to be used.
together with the DCE technical publications as a reference text. This DCE book is technically very informative in its descriptions of the various component implementations. This book will be useful as a ready reference by the side of the novice DCE programmer.

Source code listings for the different types of DCE applications are included, and are used principally as references throughout the chapters. While the sample applications are useful as teaching tools in the book, I would have preferred additional coding exercises and assignments at the end of each chapter.

The book begins with an overview of an RPC application, describing the components which need to be present in a DCE application program. The different parts of an RPC application are presented briefly, with more details covered in the later chapters of the book. The overview of the RPC application is helpful, as it gives the reader some insight into the later topics in the book. This chapter should be read first (as recommended by the author too) before one proceeds further into the book.

Chapter 2 covers the implementation of the interface definition, which is the first step required in writing a DCE application. This chapter goes over the attributes of the interface definition and how it is implemented using the DCE Interface Definition Language (IDL). Examples of data and constant definitions, procedure declarations provided in the chapter are taken from the inventory application (which is part of the set of sample applications included in the book).

The next chapter of the book covers the topic of writing client programs for the DCE RPC interface. This chapter first focuses on the subject of binding, an important specification in every client/server application. There are three main binding methods available for RPCs, the automatic method, the implicit method, and the explicit method. All three methods are described in detail with accompanying coding examples. The remaining sections of Chapter 3 cover the available methods of locating network servers, protocol sequences through the use of various RPC runtime routines.

Chapter 4 is relatively short, focusing mainly on the different usage and implementations of pointers and arrays in RPC procedure calls. This is mainly due to issues such as data movement between the client and server, varying address space executions between the client/server, etc.

Chapter 5 is quite similar to Chapter 3, except that it deals with the subject of writing server programs. This chapter assumes that the reader has read the previous chapters, and understands the workings of a distributed application, features of interface definitions, and usage of servers by client programs. It steps the reader through the entire process of writing a server application, beginning with the server initialization procedure. Interface registration, server binding, server location advertisement, and endpoint management are all part of the server initialization. The latter part of the chapter covers the issues of memory management, threads, and client binding handles in remote procedures.

Chapter 6 covers a topic which is geared more towards administration, rather than programming. It basically covers the process of creating a server entry name, followed by group entry and profile entry specifications.

Chapter 7 goes over the subject of context handles. Context handles are required in certain client/server applications since the client and the server process execute in different address spaces. Any application which requires information to be maintained between RPCs requires context handles.

The last chapter of the book (Chapter 8) covers the mechanism of pipes in DCE applications. Pipes are most useful in applications which possess large data quantities, data whose size is not initially known, or incremental data. Sections of this chapter show examples of the usage of pipes as input or output, and the management of pipes by the server application.

The latter sections of the book contain:

• a reference list of the IDL (Interface Definition Language) and ACF (Access Control Facility) attributes.

• a reference list of the DCE RPC runtime routines.

• source code listings of applications which are used throughout the book as reference for the various DCE application specifications.

It would have been nice if the book came with a machine-readable diskette which contained all the source code for the sample programs covered by the book. Most programmers who read the book will want to try out the sample applications. There is a lack of programming exercises and assignments, which I think would help application programmers get up to speed with knowledge and expertise in writing DCE applications.

Overall, I recommend Guide to Writing DCE Applications to programmers who are new to writing DCE application programs. The book is a
good reference text on the subject, and is used in conjunction with the standard DCE documentation set. The application examples provided are comprehensive, and cover the different program specifications and requirements of a variety of DCE applications.

New Units!

by Rob Kolstad
<kolstad@bsdi.com>

It's not just every day that the International Units people make up new prefixes. I was surprised and pleased to read in the latest Science News magazine that there are now even bigger and smaller units. No more having to say "teragigabytes." Now you can just say "zettabytes." And 1,000 of those (or is it 1024?) is a "yottabyte." You may laugh at these new prefixes noting their resemblance to the names of some of the Marx brothers, but they are for real. You can even see how the Exabyte company chose their name. Enjoy!

INTERNATIONAL UNIT PREFIXES, 1993

+ 1 deca- -1 deci-
+ 2 hecto- -2 centi-
+ 3 kilo- -3 milli-
+ 6 mega- -6 micro-
+ 9 giga- -9 nano-
+12 tera- -12 pico-
+15 peta- -15 femto-
+18 exa- -18 atto-
+21 zetta- -21 zepto-
+24 yotta- -24 yocto-

CERT Seminars

CERT Security Seminars

Hyatt Regency, Crystal City (in the Washington, D.C. area)

Internet Security for Managers - June 8
Internet Security for UNIX System and Network Administrators - June 9, June 10 (choose either day)

To contact CERT:
Telephone: 412-268-7090
Internet Email: <cert@cert.org>
FAX: 412/268-6989
CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
Cincinnati, Ohio
June 21-25, 1993

Preliminary Technical Sessions Program

Wednesday, June 23, 1993 9:00 am-6:00 pm

9:00 am - 10:20 am [Track 1 & Track 2]
Keynote Speaker
Bruce Tognazzini, SunSoft, Inc

11:00 am - 12:30 pm [Track 1]
Session Chair: M. Kirk McKusick, University of California, Berkeley

Call Path Profiling of Monotonic Program Resources in UNIX, Robert J. Hall, Aaron J. Goldberg, AT&T Bell Laboratories

Computer System Performance Problem Detection Using Time Series Models, Peter Hoogenboom, Jay Lepreau, Center for Software Science, Department of Computer Science, University of Utah

Design and Implementation of a Simulation Library Using Lightweight Processes, Janche Sang, Ke-hsiung Chung, Vernon Rego, Department of Computer Sciences, Purdue University

11:00 am - 12:30 pm [Track 2] Invited Talk
Five Years of Gateways and Hackers
Bill Cheswick, AT&T Bell Laboratories

2:00 pm - 3:30 pm [Track 1]
Session Chair: Matt Blaze, AT&T Bell Laboratories

The Restore-o-Mounter: The File Motel Revisited
Joe Moran, Bob Lyon, Legato Systems Incorporated

The Autofs Automounter, Brent Callaghan, Satinder Singh, SunSoft, Inc.

Discovery and Hot Replacement of Replicated Read-Only File Systems, with Application to Mobile Computing, Erez Zadok, Dan Duchamp, Computer Science Dept., Columbia University

2:00 pm - 3:30 pm [Track 2] Invited Talk
That's Easy with my Editor, Jim Blandy, Free Software Foundation, Tom Christiansen, Consultant, and Rob Pike, AT&T Bell Laboratories

4:00 pm - 6:00 pm [Track 1]
Session Chair: Pat Parseghian, AT&T Bell Laboratories

X Through the Firewall, and Other Application Relays, G. Winfield Treese, MIT Laboratory for Computer Science and Digital Equipment Corporation and Alec Wolman, University of Washington and Digital Equipment Corporation

The Ferret Document Browser, Howard P. Katseff, Thomas B. London, AT&T Bell Laboratories

LADDIS: The Next Generation in NFS File Server Benchmarking, Bruce E. Keith, Digital Equipment Corporation and Mark Wittle, Data General Corporation

Design and Implementation of a Multimedia Protocol Suite in a BSD UNIX Kernel, Raj Yavatkar, K. Lakshman, Giri Kuthethoor, Dept. of Computer Sciences, University of Kentucky

4:00 pm - 5:30 pm [Track 2] Invited Talk
Introduction to Object-Oriented Programming and C++, Roger Sessions, Object-Technology Products Group, IBM Corporation

Thursday, June 24, 9:00 am-5:30 pm

9:00 am - 10:20 am [Track 1 & Track 2] Invited Talk
Ten Problems in UNIX, and How Object Technology Solves Them, Mike Powell, Sun Microsystems Laboratories, Inc.

11:00 am - 12:30 pm [Track 1]
Session Chair: Steve Kleiman, SunSoft, Inc.

The Spring Nucleus: A Microkernel for Objects, Graham Hamilton, Panos Kougiouris, Sun Microsystems Laboratories Inc.


Anonymous RPC: Low-Latency Protection in a 64-Bit Address Space, Curtis Yarvin, Richard Bukowski, Thomas Anderson, Division of Computer Science, University of California, Berkeley

11:00 am - 12:30 pm [Track 2] Invited Talk
Digital Signal Processing 101: Sound Programming for your Workstation, Stephen A. Uhler, Bellcore
Friday, June 25, 9:00 am-12:30 pm
Session Chair: J. R. Oldroyd, Instruction Set
Fast and Flexible Shared Libraries Douglas B. Orr
Jay Lepreau, John Bonn, Center for Software Science, Department of Computer Science, University of Utah

11:00 am - 12:30 pm [Track 2] Invited Talk
UNIX Documentation: Where are We and How Did We Get Here? Linda Branagan, Convex Computer Corporation

2:00 pm - 3:30 pm [Track 1 & Track 2] Panel on Privacy
For registration and additional information please contact the USENIX Conference Office.

Supporting Members Donations

Supporting Members Donations of Equipment for USENIX Executive Offices

Since membership dues pay for only part of the Association's activities, corporate donations of equipment and services are particularly welcome. We would like express our gratitude to the following companies for their recent contributions:

Network Computing Devices, Inc. in Mountain View, California, donated two X terminals for use in our Berkeley Executive office. This donation will allow us to upgrade our office to an X windows environment. Quality Micro Systems, Inc.'s donation of an 860 print system has enabled us to gain high quality output for producing our publications.

UUNET Technologies in Falls Church, Virginia, has supported us with the donation of alternet service since January 1992, and has offered to donate an IP link which will allow connectivity between our Berkeley and the El Toro offices in 1993.

Mt Xinu, Inc. continues to provide on-call systems support. Frame Technology has donated Framemaker publishing software and technical support which has made it possible to bring the typesetting of ;login: inhouse.

March/April 1993

;login: 53
Much of the growth of UNIX has been due to its support for casual communications, thus fostering cooperative work within a location-independent framework. The latest incarnation of location independence is "Mobile Computing."

Distributed computing, now fashionable in other circles, was pioneered by the UNIX community. Support for Mobile Computing is the next logical step in assuring the role of UNIX as the operating system that offers a rich and complete feature set.

Progress in Mobile Computing is everywhere evident both in academic and non-academic circles. We intend to concentrate on mobile and location-independent computing in a true state-of-the-art symposium. We wish to sponsor a technical free-for-all on what it takes to make Mobile Computing work and work right.

**Symposium Schedule**
- Sunday, August 1, evening
  - Registration and Welcome Reception
  - Keynote by Bob Metcalfe, publisher of *InfoWorld*
- Monday, August 2, all day
  - Technical sessions will follow
- Monday, August 2, 6-10 pm
  - Vendor demonstrations and Birds-of-a-Feather sessions
- Tuesday, August 3, all day
  - Technical sessions
- Tuesday, August 3, noon
  - Hosted Luncheon with speaker

This is a single track symposium offering two days of refereed paper presentations. The symposium will include two panels, Work-in-Progress reports, Birds-of-a-Feather sessions, a series of vendor demonstrations, and a hosted luncheon (included in symposium registration).

Formally reviewed papers, presented during the symposium, will be published in the symposium proceedings, which will also include materials from Work-in-Progress reports and other similar material. Proceedings will be distributed free to attendees during the symposium and later will be available for purchase from the USENIX Association.

**Symposium Topics**
As is usual for a USENIX symposium, we are looking for new and compelling developments in systems that directly contribute to a technical understanding of Mobile Computing. UNIX will be the context of discussion, but we are eager for presentations of progress from other world views as well. The Mobile Computing Symposium will address a wide range of issues and ongoing developments, including, but not limited to:
- Naming (e.g. Prospero or OSF/DCE DNS)
- Wide area information distribution (e.g. WAIS and Archie)
- Security (e.g. authentication based on devices and digital signature services)
- Rendevous (e.g. videoconferencing over the internet and various groupware efforts)

(continued or reverse side)
- Networking and Connectability (e.g. the new IETF routing work, movement of "sockets" from site to site, and the rumored advent of IP connections from airplanes)
- Portable tiny devices (e.g. the various palmtops and personal information assistants)

**Refereed Paper Submission Details**
Submission of an extended abstract of 1500-2500 words (9000-1500 bytes or 3-5 pages) is recommended. Shorter abstracts run a significant risk of rejection as there will be little on which the program committee can base an opinion. Extended abstracts should be sent to Dan Geer at the address below. Those submitting hardcopy abstracts must send five copies.

Please also provide the following information about the author(s):
- name
- title
- affiliation
- daytime telephone
- postal address,
- e-mail address (please)
- FAX if possible
- whether you want a 15, 30 or 45 minute time slot

**Dates for Refereed Paper Submissions**
- April 19, 1993 Extended abstracts due
- May 3, 1993 Notification to authors
- June 14, 1993 Camera-ready final papers due

**For More Program Information**
For questions about refereed paper submissions and other program concerns, contact the Program Chair:
- Daniel E. Geer
  Geer Zolot Associates
  One Main Street
  Cambridge, Massachusetts USA 02142
  Telephone: +1 (617) 374-3700
  FAX: +1 (617) 374-3715
  E-mail: geer@gza.com

**For Registration Information**
Materials containing all details of the symposium program, symposium registration fees and forms, and hotel discount and reservation information will be mailed early June 1993. If you wish to receive the registration materials, please contact:
- USENIX Conference Office
  22672 Lambert Street, Suite 613
  Lake Forest, CA USA 92630
  +1 (714) 588-8649; FAX: +1 (714) 588-9706
  E-mail: conference@usenix.org

USENIX, the UNIX and Advanced Computing Systems Professional and Technical Association.
Proponents of microkernels claim that the use of this kind of technology is the inevitable next step in the engineering of operating systems. Their claim is microkernels bring the ability to support new hardware architectures and applications with no loss of performance. Whether or not this is true, this type of operating system architecture is being increasingly adopted by both industry and research.

Following the success of last year's Symposium, USENIX is pleased to announce the second USENIX Symposium on Microkernels and Other Kernel Architectures. This Symposium is aimed at exploring the different approaches to microkernels and the tradeoffs and benefits associated with each. Of particular interest is the question of whether microkernel architecture does lend itself to the support of new kinds of applications or operating systems which would be difficult or impossible to support under another operating system model.

TUTORIALS
September 20, 1993
The first day of this Symposium will feature a two track tutorial program. Expert-led tutorials will cover topics, such as current and forthcoming microkernels, of interest to the microkernels community.

TECHNICAL SESSIONS
September 21-22, 1993
The next two days will be devoted to presentation of papers from the industrial and research communities. These papers will be formally reviewed and selected by the Program Committee. The papers will be published in the Proceedings, distributed free to technical session attendees and available for purchase after the symposium from the USENIX Association.

SYMPOSIUM TOPICS
Papers are being solicited on microkernels, kernel architectures and what these bring to particular applications. Both positive and negative experiences are welcome. Topics include, but are not limited to:

- Performance and Optimization
- Fault Tolerance and High Availability
- Real-Time on Microkernels
- Scalability
- Distribution
- Evolution of Kernel Architecture
- Positive and Negative Experiences
- Use of Microkernels to Support Non-Traditional Applications
- Embedded or Dedicated Applications
- Applications Supported Directly by Microkernel

SUBMISSIONS
If you are interested in submitting a paper for the technical sessions, please submit an extended abstract. The object of an extended abstract is to convince the Committee that a good paper and 25-minute presentation will result. They need to know that the authors:

- are attacking a significant problem.
- are familiar with the current literature about the problem.
- have devised an original solution.

(continued on reverse side)
have implemented it and, if appropriate, characterized its performance.
- have drawn appropriate conclusions about what they have learned and
  why it is important.

The extended abstract should include the abstract as it will appear in the final
paper, and represent the paper in "short form." Supporting material may be
in note or outline form. Authors should include references. It should be clear
from the abstract whether the paper represents a design, an implementation
or a system that is in wide use.

Note that the Program Committee considers it unethical to submit the same
paper simultaneously to more than one conference or publication, or to sub¬
mit a paper that has been or will be published elsewhere without including
that information with the submission.

HOW TO SUBMIT
Your submission of one copy of an extended abstract will be accepted by fax,
mail, or e-mail. E-mail is greatly preferred.
- E-mail to grob@usenix.org
- Fax to +33 1 30 57 00 66
- Mail to:
  Microkernels
  USENIX Association
  2560 Ninth St., Suite 215
  Berkeley, CA USA 94710

The extended abstract may be no longer than 5 manuscript sides. Only the
first 5 sides of your submission will be sent to the Committee. The schedule
for reviewing submissions doesn't permit reviewers to read full papers. You
may attach the full paper to the extended abstract. It will not be sent to the
Committee but may be helpful during final evaluation.

Every submission should include one additional side stating:
- The name, mail address, daytime and evening phone numbers, e-mail
  address and (if available) fax number of one of the authors, who will act as
  the contact point.
- An indication of which, if any, of the authors are students.
- A list of audio/visual equipment desired beyond a microphone and an
  overhead projector.

Authors of accepted submissions will be notified by May 26, 1993. They will
receive instructions for preparing camera-ready copy of the final paper,
which must be received by July 8, 1993.

Enquiries about submissions may be made by e-mail to grob@usenix.org or
to +33 1 30 64 82 00.

FOR REGISTRATION INFORMATION
Materials containing all details of the symposium program, symposium
registration fees and forms, and hotel discount and reservation information
will be mailed June 1993. If you wish to receive registration materials, please
contact:
- USENIX Conference Office
  22672 Lambert Street, Suite 613
  Lake Forest, CA USA 92630
  (714) 588-8649; FAX: (714) 588-9706
  E-mail: conference@usenix.org
Sponsored by the USENIX Association

In association with:
The Software Engineering Research Center (SERC)

In cooperation with:
ACM SIGCOMM, ACM SIGARCH, SIGOPS and SIGSOFT (Pending)
IEEE-CS Technical Committees on Distributed Processing, Operating Systems, Software Engineering, and Design Automation

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 provides 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.

Papers that have been formally reviewed and accepted will be presented during the symposium and published in the proceedings. Invited talks will complement the peer-reviewed paper presentations. There will also be discussion panels on submitted themes. Extra-long breaks between sessions and works-in-progress reports will be provided to facilitate a workshop-like atmosphere during parts of the symposium.

Refereed Paper Submissions
Six copies of each submission or panel proposal should be sent to the Program Chair (address below) to arrive no later than June 1, 1993. Submissions of full papers are invited on any topics related to the theme of the symposium. The committee will give preferential consideration 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 SEDMS and the qualifications of the participants suggested.

For more program information, contact:
- General Chair: Peter Reiher
  Computer Science Department
  Boelter Hall
  UCLA
  Los Angeles, CA 90024
  (310) 825-8332
  reiher@wells.cs.ucla.edu
- Program Chair: David Cohn
  Computer Science and Engineering Department
  University of Notre Dame
  Notre Dame, IN 46556
  (219) 239-6694
dlc@cse.nd.edu

(continued on reverse side)
FOR REGISTRATION INFORMATION
Materials containing all details of the symposium program, symposium registration fees and forms, and hotel discount and reservation information will be mailed August 1993. If you wish to receive the registration materials, please contact:

- USENIX Conference Office
  22672 Lambert St., Suite 613
  Lake Forest, CA USA 92630
  +1 (714) 588-8649
  FAX: +1 (714) 588-9706
  E-mail: conference@usenix.org

USENIX, the UNIX and Advanced Computing Systems Professional and Technical Association.
January 18, 1993

ANNOUNCEMENT/CALL FOR PAPERS

SPONSORED BY THE USENIX ASSOCIATION
Co-Sponsored by SAGE, the Systems Administrator's Guild
In cooperation with:
The Computer Emergency Response Team (CERT)
The Association for Computer Machinery

The goal of this symposium is to bring together security practitioners, system administrators, system programmers, and others with an interest in computer security as it relates to networks and the UNIX operating system.

This will be a three and one-half day, single-track symposium. The symposium will consist of tutorials, refereed and invited technical presentations, and panel sessions. The first day will be devoted to tutorial presentations. The following two-and-one-half days of technical sessions will begin with the keynote address by Robert H. Morris. There will also be two evenings available for Birds-of-a-Feather sessions and Works-in-Progress sessions.

TUTORIALS
October 4, 1993
This one-day tutorial program will feature two tutorials, designed to address the needs of both management and technical attendees. The tutorials will supply overviews of various security mechanisms and policies. Each will provide specifics to the system and site administrator for implementing numerous local and network security precautions, firewalls, and monitoring systems.

KEYNOTE AND TECHNICAL SESSIONS
October 5-7, 1993
The keynote address by Robert H. Morris, Sr. of NCSC will begin the technical sessions program. Mr. Morris will speak on information security in computing. He will cover a number of subjects that bear directly on security. Principal among these will be the shoddy quality of software. In short, he considers the question "if the program is full of bugs, what can you say about its security?"

The technical sessions program will include refereed paper presentations, invited talks, and panel sessions. The program committee invites you to submit proposals, ideas, or suggestions for these presentations. Papers that have been formally reviewed and accepted will be presented during the symposium and published in the symposium proceedings. Proceedings will be distributed free to technical session attendees during the symposium and after will be available for purchase from the USENIX Association.

SYMPOSIUM TOPICS
Papers are being solicited in areas including but not limited to:
- User/system authentication
- File system security
- Network security
- Security and system management
- Security-enhanced versions of the UNIX operating system
- Security tools
- Network intrusions (including case studies and intrusion detection efforts)
- Security on high-bandwidth networks

(continued on reverse side)
DATES FOR REFEREED PAPER SUBMISSIONS
- Extended abstracts due: June 4, 1993
- Program Committee decisions made: June 30, 1993
- Camera-ready final papers due: August 15, 1993

TO SEND SUBMISSIONS
- Send ASCII or Postscript submissions to: ches@research.att.com
- Send hard copy submissions to the program chair:
  Bill Cheswick
  AT&T Bell Laboratories
  Room 2c416
  600 Mountain Ave.
  Murray Hill, NJ 07974

FOR REGISTRATION INFORMATION
Materials containing all details of the symposium program, symposium registration fees and forms, and hotel discount and reservation information will be mailed beginning July 1993. If you wish to receive the registration materials, please contact:
- USENIX Conference Office
  22672 Lambert Street, Suite 613
  Lake Forest, CA USA 92630
  +1 (714) 588-8649; FAX: +1 (714) 588-9706
  E-mail: conference@usenix.org

USENIX, the UNIX and Advanced Computing Systems Professional and Technical Association.
ANNOUNCEMENT/CALL FOR PARTICIPATION

The annual USENIX Systems Administration Conference provides a forum in which system administrators meet to share ideas and experiences. A growing success for the past six years, the USENIX Systems Administration Conference is the only conference which focuses specifically on the needs of system administrators. Its scope includes system administrators from sites of all sizes and configurations.

TUTORIAL PROGRAM

Monday and Tuesday, November 1-2, 1993
The two-day tutorial program at the conference is divided into three tracks with a total of twelve half-day tutorials. Attendees may move between tracks, choosing which sections most interest them. Tutorials offer expert instruction in areas of interest to system administrators, novice through experienced. Topics are expected to include Networking, Programming in Perl, X and the Administrator, the Domain Name System, Sendmail, and more.

TECHNICAL SESSIONS

Wednesday through Friday, November 3-5, 1993
"The Human Aspect of UNIX System Administration" is the theme of the first track of the conference technical sessions. Although papers of a more traditional technical content are also very welcome, the committee is especially seeking papers on areas such as creating policies and procedures, interacting with management and/or users, or which describe and evaluate methods aimed at improved communication with users and/or management. We are interested in papers which provide freely available or fully described solutions to existing problems.

The second track of the conference technical sessions will be split between presentations on very large installation system administration and presentations of practical, experience-derived material of special interest to new system administrators.

No simple measure defines "large installation." It could be number of hosts, number of users, size of network, amount of on-line storage, or some combination of these. The only certainty is that today's large will be tomorrow's standard. We wish to hear from sites which have unique problems and solutions relating to the management of large installations. We seek proposals for papers, panels, mini-workshops, or similar presentations for this track.

We also seek papers, mini-workshops, or panel presentations of pragmatic material from experienced system administrators who wish to share their experiences, hardships and solutions. It is hoped that administrators at all levels can leverage this track to solve specific problems at their site.

VENDOR DISPLAY

Wednesday, November 3, 1993, 3:00 pm-9:00 pm
Well informed vendor representatives will demonstrate products and services useful to systems and network administration at the informal table-top display accompanying the USENIX Systems Administration Conference.

If your company would like to participate, please contact Cynthia Deno at +1 (408) 335-9445, FAX +1 (408) 335-2163, E-mail: cynthia@usenix.org

CONFERENCE TOPICS

The technical sessions program will include invited talks, panels, Works-In-Progress (WIP) reports, and Birds-Of-a-Feather (BOF) sessions, alongside the refereed paper presentations. The program committee invites you to submit informal proposals, ideas, or suggestions, for the various presentation formats, on any of the following or related topics:

- Implementation, usage, and strategies for Policies and Procedures
- Effects of improved communication with users and/or management.
- Tricks in user education
- How to develop Junior System Administrators

(continued on reverse side)
Program Chair:
Bjorn Satdeva, /sys/admin, Inc.
Brent Chapman, Great Circle Associates
Lee Damon, Castle PALIS
Tina M. Darmohray, Lawrence Livermore National Labs
Janet Frazer, UNIX System Laboratories, Inc.
Helen Harrison, SAS Institute
Bryan McDonald, SRI International
Arch Mott, Cisco Systems, Inc.
Paul Moriarty, Cisco Systems, Inc.
Jeff Polk, Berkeley Software Design, Inc.
Greg Rose, Australian Computing and Communications Institute
Peg Schafer, Bolt Beranek & Newman, Inc.
Steve Simmons, Industrial Technology Institute
Liza Y. Weissler, RAND Corporation
Pat Wilson, Dartmouth College
Elizabeth Zwicky, SRI International

System Security Monitoring
Security issues, especially where multiple people are privileged users
Heterogeneous system administration
Human issues of administration
Graphical User Interfaces for system administration
Distributed system administration
Network growth and performance management
Network management
Network monitoring
Wireless LANs
Evaluating performance of high-end workstations and servers
Integration of heterogeneous systems
Usage monitoring and accounting systems
Administration of remote sites

Refereed Paper Submissions
The committee requires that an extended abstract be submitted for the paper selection process. (Full-papers are not acceptable for this stage; if you send a full paper, you must also include an extended abstract for evaluation.) Your extended abstract should consist of a traditional abstract which summarizes the content/ideas of the entire paper, followed by a skeletal outline of the full paper. We require electronic (nroff/troff, TeX or ASCII) submission of the extended abstract.

Authors of an accepted paper will present their paper and provide a final paper for publication in the Conference Proceedings. Final papers are limited to 20 pages, including diagrams, figures and appendix. Papers should include a brief description of the site (if applicable), an outline of the problem and issues, and details of the solution. Authors may provide electronic versions or camera-ready copy (instructions will be provided upon request) of final papers. Conference proceedings will be distributed to all conference attendees and will also be available from the USENIX Association after the conference.

Address For Submission
For submission of all proposals and of extended abstracts of refereed papers, and for program information, contact:

Program Chair: Bjorn Satdeva
/sys/admin, Inc.
2787 Moorpark Avenue
San Jose, CA 95128
+1 (408) 241-3111
E-mail: bjorn@sysadmin.com

For Registration Information
Materials containing all details of the symposium program, symposium registration fees and forms, and hotel discount and reservation information will be mailed and posted to the net beginning August 1993. If you wish to receive registration materials, please contact:

USENIX Conference Office
22672 Lambert Street, Suite 613
Lake Forest, CA 92630 USA
+1 (714) 588-8649
FAX: +1 (714) 588-9706
E-mail: conference@usenix.org

USENIX, the UNIX and Advanced Computing Systems Professional and Technical Association.
AUUG '93

Preliminary Announcement and Call for Papers
AUUG '93
Darling Harbour
Sydney, Australia
September 27-30, 1993

AUUG, Inc., forum for UNIX Open Systems Users Presents: "Results through Open Systems." Over the past several years we have heard about 'What are Open Systems', and 'Maintaining Control with Open Systems'. Now it’s time to hear about the results which have been achieved. Rapid expansion, the challenge of integration, global networking, and security are all issues of importance and concern to users around the world.

AUUG '93 solicits papers on all aspects of UNIX and open systems, and particularly on successful applications and implementations of open systems technology to age-old and newly emerging problems.

Events: AUUG '93 will be a four day conference, commencing September 27, 1993. The first day will be devoted to tutorial presentations, followed by three days of papers, work-in-progress sessions and BOFs.

Tutorials: Provisions for two full-day tutorials and up to eight half-day tutorials have been made. These sessions, typically in a lecture format, are targeted to educate the audience and arm them with innovative "how to" lessons. Please submit tutorial abstracts, along with preference for a half- or fullday slot to: <ggr@acci.com.au>. Please be sure to include your complete contact information (phone, FAX, postal code and Email addresses) in all correspondence.

Papers: AUUG '93 provides dual Technical and Management tracks for the presentations. To share your innovative implementations, applications, and similar areas submit your abstract for the technical track. We are also interested in your experiences, case studies, strategic issues, and the like. If your topic better fits these areas submit your abstract for the Management track. The above should not, of course, discourage papers which are appropriate for both audiences at once. Vendor product announcements will be automatically rejected unless specifically submitted for the special advertising stream.

Prize for the Best Student Paper: A cash prize of $500 will be awarded for the best paper submitted by a full-time student at an accredited tertiary educational institution. In addition, the ten 'runners-up' will be rewarded with free registration.

Work-in-Progress and Advertising Sessions: These brief 15 minute sessions are designed to report on current work with fundamental aspects highlighted. New to the AUUG conference are the Advertising sessions. These are devoted to new products only. Product specification sheets should be submitted with your abstract.

Speaker Incentives: Presenters of papers are afforded free conference registration. Tutorial presenters will receive 25% of the profit for their session and a free conference registration.

Form of Submissions: Please indicate whether your submission is relevant to the technical or management audiences, or both. In either case, submissions are required to be in the form of an abstract and an outline. Please provide sufficient detail to allow the committee to make a reasoned decision about the final paper; of course a full paper is also perfectly acceptable. For more information on submissions please contact the address below.

Programme Committee:
Piers Lauder – Sydney University (Chair)
Liz Fraumann – AUUG
Ian Hoyle – BHP Research Labs
Hugh Irvin – connect.com
Rolf Jester – Digital Equipment Corporation
Bob Kummerfeld – Sydney University
Phil McCrea – Softway P/L
Andrew McRae – Megadata P/L
Greg Rose – Australian Computing and Communications Institute

Relevant Dates:
Abstract and outlines due: April 6, 1993
Notifications to authors: April 26, 1993
Final Papers due: July 26, 1993

Email: <auug93@cs.su.oz.au>
Phone: +61 2 361-5994
FAX +61 2 332-4066

AUUG '93 Programme
P.O. Box 366
Kensington, NSW 2033
Preliminary Announcement and Call for Papers
UKUUG LISA '93 – “Coping with Change”
London, June 30, 1993

by Neil Todd
<neil@pio.gid.co.uk>

The UKUUG announces that the theme of this year’s System Administration and Management conference will be “Coping with Change,” with an informal subtitle of “Strategies for hitting a moving target.”

With the increasing complexity of the working environment and the rate of change of that environment, in terms of both the variety of hardware platforms and associated operating systems, as well as the increasing number of third party software products available for those platforms, the task of the System Manager has become increasingly difficult.

How does one encapsulate the differences in System Administration procedures? How does one evolve strategies for supporting complex third party tools and their varying licensing methods? How does one set about evaluating new software and hardware?

The requirements of quality certification procedures mean that much better system operation documentation is required, and must be produced. How can this documentation be generated, and then be kept up to date with the minimum of manual intervention?

SAGE/UK

It is not only the working environment that is changing, the very role of the System Administrator is changing. System Administrators are now being recognised as being more than just skilled operators. They need to be involved in the planning process when new systems and networks are being ordered. However, at the same time organisations often make inexperienced people take on the job of System Administrator without adequate training or support.

In order to help raise the standard of System Administration and to advance it as a profession, this conference will see the creation of SAGE/UK—a UKUUG Special Technical Group for System Administrators. Successful SAGE groups are already running in the USA and Australia and the UK group will keep in close contact with their activities. The group will provide a focal point for System Administration activities and will organise workshops and tutorials as well as develop guidelines for the proper management of systems.

Call for Papers

Papers are requested on topics relating to the broad themes outlined above. Submissions on other System Management themes are also welcomed.

All accepted authors will be expected to submit a paper in electronic form conforming to the conference guidelines. Copies of the guidelines are available from the UKUUG Secretariat.

You do not have to be a member of UKUUG to submit a paper. Submissions from speakers from outside of the UK are welcomed.

Significant dates

Closing date for abstracts: April 2, 1993
Accepted authors notified: April 7, 1993
Final papers due: May 15, 1993

Method of submission

Potential authors may request further information by sending Email to <ukuug-lisa-93@bnr.co.uk>, or may contact a member of the programme committee directly.

Initial abstracts should be sent either electronically to <ukuug-lisa-93@bnr.co.uk>, or in hard copy format to the UKUUG Secretariat. Electronic submission is preferred. All submissions will be acknowledged.

Programme Committee

Neil Todd, Chair
GID Ltd.
Captain’s Gorse
Upper Basildon
Reading
Berks RG8 8SZ
United Kingdom
+44 491 671 964
<neil@pio.gid.co.uk>

Andrew Macpherson
BNR (Europe) Ltd.
London Road
Harlow
Essex
CM17 9NA
United Kingdom
+44 279 402423
<a.macpherson@bnr.co.uk>

Bill Barrett, UKUUG Sec.
UKUUG
Owles Hall
Buntingford
Herts
SG9 9PL
United Kingdom
+44 763 273475
+44 763 273255 (FAX)
<bill@ukuug.ucp>

Lindsay Marshall
Dept. of Comp. Science
Univ. of Newcastle
Newcastle upon Tyne
NE1 7RU
United Kingdom
+44 91 222 8267
<lindsay.marshall@newcastle.ac.uk>

March/April 1993
Support GNU Development

Support GNU Development – Get the FSF GNU CD-ROM

by Richard Stallman
<rms@gnu.ai.mit.edu>

The Free Software Foundation now offers a CD-ROM which contains sources to the full distribution of the GNU Project. This is a way of getting GNU software in convenient form; it is also a way to help fund the continued development and maintenance of GNU software.

The FSF source code CD-ROM contains nearly everything on the six main FSF distribution tapes, including Emacs, GCC, G++, GDB, Bison, GAS, make, GAWK, Texinfo, the GNU Utilities, RCS and CVS, f2c, gnuplot, Ghostscript, tar, diff, and BASH, as well as the MIT X Window System, MIT Scheme, the Andrew Toolkit, and TeX. The CD contains more than 46,000 files. Versions are now current as of November 1992; newer editions will be made from time to time.

Also, the CD contains packages which run on 80386 or 80486 machines under MS-DOS, including: Demacs (GNU Emacs for MS-DOS), DJGPP (a port of GCC 2.2.2), and MIT Scheme 7.2. In addition, it contains Mtools, which allows UNIX systems to read, write, and manipulate files on an MS-DOS filesystem (typically a diskette).

The CD is in ISO 9660 format. You can mount it as a read-only file system on most operating systems. If your system supports symbolic links, you can build most of this software without needing to copy the sources off the CD; you need only enough free disk space for the object files and the intermediate build targets. Except for several of the MS-DOS packages, there are no precompiled programs on this CD. You will need a C compiler (programs which need some other sort of interpreter or compiler normally provide the C source to a bootstrapping program).

The basic price of the CD is $400; this is the contents of six FSF distribution tapes for the price of two tapes. Companies should have no trouble affording this; but for individuals, we offer a lower price: $100. This is not a license fee; it is the price for the disk. In either case, the contents are free software, so you may redistribute copies to anyone.

In the long run, it makes sense to choose the FSF as your source for distribution copies of GNU software, because only that way supports GNU software development significantly. The other ways you can get copies provide little or no funds for free software development.

The FSF also gratefully accepts donations of any size; as we are tax exempt, your donations are tax-deductible. But usually the easiest way to support the FSF is by ordering a software distribution, such as a CD-ROM.

Like listener-supported radio, we depend on you to continue our work. If you use GNU software, and you have not supported the FSF recently, isn’t it time?

For more information on GNU and the Foundation, contact us at: <gnu@prep.ai.mit.edu> or the postal address on the order form.

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

Prices and contents may change without notice after June 30, 1993. All software and documentation is distributed with permission to copy and to redistribute. Texinfo source for each manual is included. All Free Software Foundation software is provided on an "as is" basis, with no warranty of any kind. This order form expires June 30, 1993.

You can also order tapes and printed manuals from the FSF. Contact <gnu@prep.ai.mit.edu> for more information. 

____@ $100 = $______ GNU CD-ROM for individuals.

____@ $400 = $______ GNU CD-ROM for businesses.

Subtotal $______

+ $_____ In Massachusetts: add 5% sales tax, or give tax exempt number.
+ $_____ In Alaska, Hawaii, or Puerto Rico for shipping: $5 for each CD.
+ $_____ Outside of U.S., Canada & Puerto Rico, for shipping: $30 for one CD, and $5 for ea. addl. CD.
+ $_____ Optional tax-deductible donation.

TOTAL $______ (We pay for shipping via UPS ground transportation – in the contiguous 48 states and Canada.)

Orders filled only upon receipt of check or money order in US dollars.

Please make checks payable to the Free Software Foundation.

Name: ________________________________

Mail Stop/Dept. Name: ________________________________

Organization: ________________________________

Street Address: ________________________________

City/State/Province: ________________________________

Zip Code/Postal Code/Country: ________________________________

Please provide the following information in case of a problem with your order, or to enable customs agents to contact you:

Telephone number: ________________________________ FAX number: ________________________________

Email address: ________________________________

Do you want to receive future GNU announcements? (Yes No)

For orders from outside the US: Orders must be paid in US dollars. You are responsible for paying all duties, tariffs, and taxes. If you refuse to pay the charges, the shipper will return or abandon the order.

To order in Japan, please contact the FSF for more information by FAX at 0066-3382-0158 or send Email to <japan@gnu.ai.mit.edu>.

Please mail orders to:
Free Software Foundation
675 Massachusetts Avenue
Cambridge, MA 02139 USA
617-876-3296

This order form expires June 30, 1993
That's right, you'll receive your personal copy of the *1993 Open Systems Products Directory* **FREE** with your paid membership in UniForum: The International Association of Open Systems Professionals.

Acclaimed by *UNIXWorld* as “the most comprehensive guide to UNIX products,” the 1993 Directory contains more than 7500 products and services in one easy-to-use volume. Well designed indexes and tabs guide you right to the information you want. That's why *UNIXWorld* calls the Directory “Old Reliable,” and why you'll call it “indispensable.”

Boasting an extensive roster of 2000+ vendors, the Directory offers 1500 pages of unmatched coverage of:

- Horizontal and Vertical Software
- Hardware Systems From PCs to Supercomputers
- Peripherals From Modems to Printers
- Services From Employment Search to Training
- XPG and POSIX Conformance Information

Priced to sell to non-members at $95.00, the *1993 Open Systems Product Directory* is a real value. But why spend a penny extra when it's yours free when you join UniForum?

Becoming a member is a smart move. By joining you get the **Directory plus all these member benefits:**

- Subscription to *UniForum Monthly* — the colorful authoritative voice of the UNIX and Open Systems industry.
- Subscription to *UniNews* — the biweekly official newsletter of the Association containing insightful information you can rely on.
- UniForum Technical Publications — precise and credible, written by today's leading experts on systems and standards.
- Member discounts to all UniForum conferences and trade shows.
- Discounts on publications and services from publishers and vendors — available exclusively to UniForum members.

Your membership entitles you to all of this, and the *1993 Open Systems Products Directory* is yours **FREE**, when you join. Membership is only $100.00 per year and ordering is easy, too. In the U.S. call toll free **1-800-255-5620**; calling from outside the U.S. dial **408-986-8840**. Have your major credit card ready and we'll start your membership at once.

You won't find a better offer anytime soon. So call now and become part of the largest group of open systems professionals in the world. Join UniForum, today!

---

2901 Tasman Drive, Suite 201, Santa Clara, CA 95054 (800) 255-5620 (408) 986-8840 FAX: (408) 986-1645
## Publications Order Form

### Conference & Workshop Proceedings

<table>
<thead>
<tr>
<th>Qty</th>
<th>Proceedings</th>
<th>Member Price</th>
<th>Non-Member Price</th>
<th>Subtotal</th>
<th>Overseas Postage</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>WINTER/SUMMER CONFERENCES</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>San Diego</td>
<td>Winter '93</td>
<td>33</td>
<td>40</td>
<td>$</td>
<td>$25</td>
</tr>
<tr>
<td></td>
<td>San Antonio</td>
<td>Summer '92</td>
<td>23</td>
<td>30</td>
<td>$</td>
<td>$14</td>
</tr>
<tr>
<td></td>
<td>San Francisco</td>
<td>Winter '92</td>
<td>30</td>
<td>39</td>
<td>$</td>
<td>$22</td>
</tr>
<tr>
<td></td>
<td>Nashville</td>
<td>Summer '91</td>
<td>32</td>
<td>38</td>
<td>$</td>
<td>$22</td>
</tr>
<tr>
<td></td>
<td>Dallas</td>
<td>Winter '91</td>
<td>28</td>
<td>34</td>
<td>$</td>
<td>$18</td>
</tr>
<tr>
<td></td>
<td>Anaheim</td>
<td>Summer '90</td>
<td>22</td>
<td>22</td>
<td>$</td>
<td>$15</td>
</tr>
<tr>
<td></td>
<td>Washington, DC</td>
<td>Winter '90</td>
<td>25</td>
<td>25</td>
<td>$</td>
<td>$15</td>
</tr>
<tr>
<td></td>
<td>Baltimore</td>
<td>Summer '89</td>
<td>20</td>
<td>20</td>
<td>$</td>
<td>$15</td>
</tr>
<tr>
<td></td>
<td>San Diego</td>
<td>Winter '89</td>
<td>30</td>
<td>30</td>
<td>$</td>
<td>$20</td>
</tr>
<tr>
<td></td>
<td>San Francisco</td>
<td>Summer '88</td>
<td>29</td>
<td>29</td>
<td>$</td>
<td>$20</td>
</tr>
<tr>
<td></td>
<td>Dallas</td>
<td>Winter '88</td>
<td>26</td>
<td>26</td>
<td>$</td>
<td>$15</td>
</tr>
<tr>
<td></td>
<td>Phoenix</td>
<td>Summer '87</td>
<td>35</td>
<td>35</td>
<td>$</td>
<td>$20</td>
</tr>
<tr>
<td></td>
<td>Washington, DC</td>
<td>Winter '87</td>
<td>10</td>
<td>10</td>
<td>$</td>
<td>$15</td>
</tr>
<tr>
<td></td>
<td>Atlanta</td>
<td>Summer '86</td>
<td>37</td>
<td>37</td>
<td>$</td>
<td>$20</td>
</tr>
<tr>
<td></td>
<td>Denver</td>
<td>Winter '86</td>
<td>25</td>
<td>25</td>
<td>$</td>
<td>$15</td>
</tr>
<tr>
<td></td>
<td>Portland</td>
<td>Summer '85</td>
<td>45</td>
<td>45</td>
<td>$</td>
<td>$25</td>
</tr>
<tr>
<td></td>
<td>Dallas</td>
<td>Winter '85</td>
<td>15</td>
<td>15</td>
<td>$</td>
<td>$10</td>
</tr>
<tr>
<td></td>
<td>Salt Lake City</td>
<td>Summer '84</td>
<td>29</td>
<td>29</td>
<td>$</td>
<td>$20</td>
</tr>
<tr>
<td></td>
<td>Washington, DC</td>
<td>Winter '84</td>
<td>25</td>
<td>25</td>
<td>$</td>
<td>$15</td>
</tr>
<tr>
<td></td>
<td>Toronto</td>
<td>Summer '83</td>
<td>32</td>
<td>32</td>
<td>$</td>
<td>$20</td>
</tr>
<tr>
<td></td>
<td>San Diego</td>
<td>Winter '83</td>
<td>28</td>
<td>28</td>
<td>$</td>
<td>$15</td>
</tr>
</tbody>
</table>

### LARGE INSTALLATION SYSTEMS ADMINISTRATION

<table>
<thead>
<tr>
<th>Qty</th>
<th>Proceedings</th>
<th>Member Price</th>
<th>Non-Member Price</th>
<th>Subtotal</th>
<th>Overseas Postage</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LISA VI</td>
<td>Oct. '92</td>
<td>23</td>
<td>30</td>
<td>$</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>LISA V</td>
<td>Sept. '91</td>
<td>20</td>
<td>23</td>
<td>$</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>LISA IV</td>
<td>Oct. '90</td>
<td>15</td>
<td>18</td>
<td>$</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>LISA III</td>
<td>Sept. '89</td>
<td>13</td>
<td>13</td>
<td>$</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>LISA II</td>
<td>Nov. '88</td>
<td>8</td>
<td>8</td>
<td>$</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>LISA I</td>
<td>April '87</td>
<td>4</td>
<td>4</td>
<td>$</td>
<td>5</td>
</tr>
</tbody>
</table>

### C++

<table>
<thead>
<tr>
<th>Qty</th>
<th>Proceedings</th>
<th>Member Price</th>
<th>Non-Member Price</th>
<th>Subtotal</th>
<th>Overseas Postage</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C++ Conference</td>
<td>Aug. '92</td>
<td>30</td>
<td>39</td>
<td>$</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>C++ Conference</td>
<td>Apr. '91</td>
<td>22</td>
<td>26</td>
<td>$</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>C++ Conference</td>
<td>Apr. '90</td>
<td>28</td>
<td>28</td>
<td>$</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>C++ Conference</td>
<td>Oct. '88</td>
<td>30</td>
<td>30</td>
<td>$</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>C++ Workshop</td>
<td>Nov. '87</td>
<td>30</td>
<td>30</td>
<td>$</td>
<td>20</td>
</tr>
</tbody>
</table>

### SECURITY

<table>
<thead>
<tr>
<th>Qty</th>
<th>Proceedings</th>
<th>Member Price</th>
<th>Non-Member Price</th>
<th>Subtotal</th>
<th>Overseas Postage</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UNIX Security III</td>
<td>Sept. '92</td>
<td>30</td>
<td>39</td>
<td>$</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>UNIX Security II</td>
<td>Aug. '90</td>
<td>13</td>
<td>16</td>
<td>$</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>UNIX Security</td>
<td>Aug. '88</td>
<td>7</td>
<td>7</td>
<td>$</td>
<td>5</td>
</tr>
</tbody>
</table>

### MACH

<table>
<thead>
<tr>
<th>Qty</th>
<th>Proceedings</th>
<th>Member Price</th>
<th>Non-Member Price</th>
<th>Subtotal</th>
<th>Overseas Postage</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mach Symposium</td>
<td>Nov. '91</td>
<td>24</td>
<td>28</td>
<td>$</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Mach Workshop</td>
<td>Oct. '90</td>
<td>17</td>
<td>20</td>
<td>$</td>
<td>9</td>
</tr>
</tbody>
</table>

March/April 1993
<table>
<thead>
<tr>
<th>Qty</th>
<th>Proceedings</th>
<th>Member Price</th>
<th>Non-Member Price</th>
<th>Subtotal</th>
<th>Overseas Postage</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>DISTRIBUTED &amp; MULTIPROCESSOR SYSTEMS (SEDMS)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SEDMS III Mar. '92</td>
<td>30</td>
<td>36</td>
<td>$_____ 20</td>
<td></td>
<td>$_____</td>
</tr>
<tr>
<td></td>
<td>SEDMS II Mar. '91</td>
<td>30</td>
<td>36</td>
<td>$_____ 20</td>
<td></td>
<td>$_____</td>
</tr>
<tr>
<td></td>
<td>SEDMS Oct. '89</td>
<td>30</td>
<td>30</td>
<td>$_____ 20</td>
<td></td>
<td>$_____</td>
</tr>
<tr>
<td></td>
<td><strong>GRAPHICS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Graphics Workshop V Nov. '89</td>
<td>18</td>
<td>18</td>
<td>$_____ 10</td>
<td></td>
<td>$_____</td>
</tr>
<tr>
<td></td>
<td>Graphics IV Oct. '87</td>
<td>10</td>
<td>10</td>
<td>$_____ 10</td>
<td></td>
<td>$_____</td>
</tr>
<tr>
<td></td>
<td>Graphics III Nov. '86</td>
<td>10</td>
<td>10</td>
<td>$_____  5</td>
<td></td>
<td>$_____</td>
</tr>
<tr>
<td></td>
<td>Graphics II Dec. '85</td>
<td>7</td>
<td>7</td>
<td>$_____  5</td>
<td></td>
<td>$_____</td>
</tr>
<tr>
<td></td>
<td><strong>OTHER WORKSHOPS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>File Systems May '92</td>
<td>15</td>
<td>20</td>
<td>$_____  9</td>
<td></td>
<td>$_____</td>
</tr>
<tr>
<td></td>
<td>Micro-Kernel &amp; Other Kernel Arch.</td>
<td>30</td>
<td>39</td>
<td>$_____ 20</td>
<td></td>
<td>$_____</td>
</tr>
<tr>
<td></td>
<td>UNIX Transaction Processing May '89</td>
<td>12</td>
<td>12</td>
<td>$_____  8</td>
<td></td>
<td>$_____</td>
</tr>
<tr>
<td></td>
<td>Software Management Apr. '89</td>
<td>20</td>
<td>20</td>
<td>$_____ 15</td>
<td></td>
<td>$_____</td>
</tr>
<tr>
<td></td>
<td>UNIX &amp; Supercomputers Sept. '88</td>
<td>20</td>
<td>20</td>
<td>$_____ 10</td>
<td></td>
<td>$_____</td>
</tr>
</tbody>
</table>

Discounts are available for bulk orders. Please inquire.

**Payable in US Funds Only**

*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

Please allow 2–3 weeks for delivery. Overseas orders are shipped via air printed matter.

If you wish to receive our membership information packet, please check this box. □
FL - Melbourne

Meets the 3rd Monday of every month.

Space Coast UNIX User’s Group
Steve Lindsey (407) 242-4766
lindsey@vnet.ibm.com

FL – Orlando:

Meets the 3rd Thursday of each month.

Central Florida UNIX Users Group
Mikel Manitius (407) 444-8448
mike@aaa.com

FL – Western:

Meets 1st Thursday of each month.

Florida West Coast UNIX Users Group
Richard Martino (813) 536-1776
Tony Becker (813) 799-1836
mcros@tony
Ed Gallizzi, Ph.D. (813) 864-8272
e.gallizzi@compmail.com
Jay Ts (813) 979-9169
uunet!pdn!scs!metran!jan
Dave Lewis (407)242-4372
dhl@ccd.harris.com

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 or csufres@brent

Commercial institutions or individuals:
Gordon Crumal (209) 251-2648
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 frug-info@frug.org.

Front Range UNIX Users Group
Software Design & Analysis, Inc.
1113 Spruce St., Ste. 500
Boulder, CO 80302
Steve Gaede (303) 444-9100
gaede@frug.org

D.C. – 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

FL – Coral Springs

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

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

KS or MO – Kansas:

Meets on 2nd Monday of each month.

Kansas City UNIX Users Group (KUUG)
813B Street
Blue Springs, MO 64015
(816) 235-5212
mlg@cstp.umkc.edu

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
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
uunet!gallist!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
mccp!phil

NM – Albuquerque:
ASIGUNIX meets every 3rd Wednesday of each month. Phil Hertz 505/275-0466.

NY – New York City:
Meets every other month in Manhattan.
Unigroup of New York City
G.P.O. Box 1931
New York, NY 10116

OK - Tulsa:
Meets 2nd Wednesday of each month.
Tulsa UNIX Users Group, $USR
Stan Mason (918) 560-5329
tulix!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@ca.tus.org
Tom Painter (512) 835-5457
president@ca.tus.org

TX – Dallas/Fort Worth:
Meets the 1st Thursday of each month.
Dallas/Fort Worth UNIX Users Group
P.O. Box 867405
Plano, TX 75086
Evan Brown (214) 519-3577
evbrown@dsccc.org

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

CANADA – Toronto:
143 Baronwood Court
Brampton, Ont. Canada L6V 3H8
Evan Leibovitch (416) 452-0504
evan@telly.on.ca

CANADA – Ottawa:
The Ottawa Carleton UNIX Users Group
D.J. Blackwood (613)957-9305
dave@revcan.rct.ca

March/April 1993
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
<jr@inset.com>

Mailing list: <bblisa@inset.com>
List Requests: <bblisa-request@inset.com>

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> 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, (3b1amiga.games/misc, reviewed,sun_unix,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, HyperNeWS, 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/2. We hope to include them on a future issue, once the dust has settled a bit.

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

USENIX members may purchase the issue for $50. 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
415-112 N. Mary Ave., Suite 50
Sunnyvale, CA 94086 USA
+1 408/738-4832 (Voice), -2050 (FAX)
<ptf@cfcL.com>

Issue 2-1 is in production at press time. Call for details.
### Calendar of Events

<table>
<thead>
<tr>
<th>Year</th>
<th>Month</th>
<th>Event</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1993</strong></td>
<td>Apr</td>
<td>19-21</td>
<td>Mach III, Santa Fe, NM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19-21</td>
<td>SANS II – Washington, DC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19-23</td>
<td>IEEE 1003, Irvine, CA</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>20-22</td>
<td>UniForum NZ, New Zealand</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>25-27</td>
<td>NeXT WORLD, San Francisco, CA</td>
</tr>
<tr>
<td></td>
<td>Jun</td>
<td>5-11</td>
<td>DECUS, Atlanta, GA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>21-25</td>
<td>USENIX, Cincinnati, OH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30</td>
<td>UKUUG LISA, London, UK</td>
</tr>
<tr>
<td></td>
<td>Jul</td>
<td>12-16</td>
<td>IEEE 1003, Denver, CO</td>
</tr>
<tr>
<td></td>
<td>Aug</td>
<td>1</td>
<td>ACM Siggraph, Anaheim, CA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2-3</td>
<td>Mobile &amp; Location Independent Computing, Cambridge, MA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10-13</td>
<td>FIRST Computer Security Incident Handling, St. Louis, MO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>23-27</td>
<td>Interop, San Francisco, CA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20-22</td>
<td>Microkernels II, San Diego, CA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>23-24</td>
<td>SEDMS IV, San Diego, CA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>22-24</td>
<td>EurOpen/GUUG, Wiesbaden, Germ.</td>
</tr>
<tr>
<td></td>
<td>Sept</td>
<td>26-</td>
<td>ACM OOPSLA, Washington, DC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Oct 27-30</td>
<td>AUUG, Sydney, Australia</td>
</tr>
<tr>
<td></td>
<td>Oct</td>
<td>4-6</td>
<td>UNIX Security Symposium IV, Santa Clara, CA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18-22</td>
<td>IEEE 1003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25-29</td>
<td>Interop '93 Europe, Paris, France</td>
</tr>
<tr>
<td></td>
<td>Nov</td>
<td>1-5</td>
<td>LISA VII, Monterey, CA</td>
</tr>
<tr>
<td></td>
<td>Dec</td>
<td>4-10</td>
<td>DECUS, San Francisco, CA</td>
</tr>
<tr>
<td><strong>1994</strong></td>
<td>Jan</td>
<td>17-21</td>
<td>USENIX, San Francisco, CA</td>
</tr>
<tr>
<td></td>
<td>Mar</td>
<td>23-25</td>
<td>UniForum, San Francisco, CA</td>
</tr>
<tr>
<td></td>
<td>Spring</td>
<td>*</td>
<td>C++ Conference</td>
</tr>
<tr>
<td></td>
<td></td>
<td>*</td>
<td>UNIX Application Development</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>7-13</td>
<td>DECUS, New Orleans, LA</td>
</tr>
<tr>
<td></td>
<td>Jun</td>
<td>6-10</td>
<td>USENIX, Boston, MA</td>
</tr>
<tr>
<td></td>
<td>Sep</td>
<td>12-16</td>
<td>Interop, San Francisco, CA</td>
</tr>
<tr>
<td></td>
<td>Oct</td>
<td>23-27</td>
<td>ACM OOPSLA, Portland, OR</td>
</tr>
<tr>
<td></td>
<td>Nov</td>
<td>12-18</td>
<td>DECUS, Anaheim, CA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Year</th>
<th>Month</th>
<th>Event</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1995</strong></td>
<td>Jan</td>
<td>16-20</td>
<td>USENIX, New Orleans, LA</td>
</tr>
<tr>
<td></td>
<td>Feb</td>
<td>21-23</td>
<td>UniForum, Dallas, TX</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>13-19</td>
<td>DECUS, New Orleans, LA</td>
</tr>
<tr>
<td></td>
<td>Jun</td>
<td>19-22</td>
<td>USENIX, San Francisco, CA</td>
</tr>
<tr>
<td></td>
<td>Nov</td>
<td>2-8</td>
<td>DECUS, San Francisco, CA</td>
</tr>
<tr>
<td><strong>1996</strong></td>
<td>Jan</td>
<td>22-26</td>
<td>USENIX, San Diego, CA</td>
</tr>
<tr>
<td></td>
<td>Mar</td>
<td>12-14</td>
<td>UniForum, San Francisco, CA</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>18-24</td>
<td>DECUS, Orlando, FL</td>
</tr>
<tr>
<td></td>
<td>Nov</td>
<td>16-22</td>
<td>DECUS, Anaheim, CA</td>
</tr>
</tbody>
</table>

This is a combined calendar of planned conferences, symposia, 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>. Please provide your information in the same format as above.

* = events sponsored by the USENIX Association.

<table>
<thead>
<tr>
<th>ACM: Association for Computing Machinery</th>
<th>AUUG: Australian UNIX Users Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECUS: Digital Equipment Computer Users Society</td>
<td>EurOpen: European Forum for Open Systems</td>
</tr>
<tr>
<td>FIRST: Forum of Incidence Response &amp; Security Team</td>
<td>IETF: Internet Engineering Task Force</td>
</tr>
<tr>
<td>IEEE: Institute of Electrical and Electronics Engineers</td>
<td>INET: Internet Society</td>
</tr>
<tr>
<td>Interex: Intl. Assoc.–Hewlett-Packard Comp. Users</td>
<td>JUS: Japan UNIX Society</td>
</tr>
<tr>
<td>LISA: USENIX Systems Administration Conference</td>
<td>OOPSLA: Object - oriented Programming Systems, Languages, and Applications</td>
</tr>
<tr>
<td>SANS: Conf. on Tools &amp; Techniques for System Admin., Networking &amp; Security</td>
<td>SEDMS: Symposium on Experiences with Distributed and Multiprocessor Systems</td>
</tr>
<tr>
<td>UKUUG: United Kingdom UNIX Systems Users Group</td>
<td>UniForum: International Association of UNIX and Open Systems Professionals</td>
</tr>
</tbody>
</table>

March/April 1993
UPCOMING SYMPOSIA AND CONFERENCES

JUNE 21–25, 1993
SUMMER 1993
TECHNICAL CONFERENCE
Program Chair: David S. H. Rosenthal,
SunSoft, Inc.
Cincinnati Convention Center,
Cincinnati, Ohio

SEPTEMBER 23–24, 1993
EXPERIENCES WITH DISTRIBUTED
& MULTIPROCESSOR SYSTEMS
(SEDMIS IV)
General Chair: Peter Reiner, UCLA
Program Chair: David Cohn,
University of Notre Dame
Hilton Beach & Tennis Resort,
San Diego, California
In cooperation with: ACM SIGARCH, SIGCOMM, SIGOPS and SIGSOFT
and IEEE-CS Technical Committees on Distributed Processing,
Operating Systems, Software Engineering, and Design Automation

AUGUST 2–3, 1993
SYMPOSIUM ON MOBILE &
LOCATION-INDEPENDENT
COMPUTING
Program Chair: Dan Geer,
Geer Zolot Associates
Vice-Program Chair: Clement Cole,
Locus Computing Corporation
Marriott Hotel,
Cambridge, Massachusetts

OKTOBER 4–7, 1993
4TH
UNIX SECURITY SYMPOSIUM
Co-sponsored with The Computer
Emergency Response Team (CERT)
Program Chair: Bill Cheswick,
AT&T Bell Laboratories
Santa Clara Marriott Hotel,
Santa Clara, California

SEPTEMBER 20–22, 1993
2ND SYMPOSIUM ON
MICROKERNELS &
OTHER KERNEL ARCHITECTURES
Program Chair: Lori S. Grob,
Chorus systèmes
Hilton Beach & Tennis Resort,
San Diego, California

NOVEMBER 1–5, 1993
7TH SYSTEMS ADMINISTRATION
CONFERENCE
(LISA VII)
Co-sponsored with SAGE,
the Systems Administrators’ Guild
Program Chair: Bjorn Satdeva,
lsyladmin, Inc.
Marriott Hotel,
Monterey, California

TO RECEIVE FULL INFORMATION
Please contact: USENIX Conference Office, 22672 Lambert St., Suite 613, El Toro, CA 92630 USA
+1 (714) 588-8943; FAX: +1 (714) 588-9706; e-mail: upcoming@usenix.org