Predicting the Future

All those software managers who ask for project completion dates are engaged in the difficult practice of "Predicting the Future." As each of us watches the marketplace with announcements and shipdates, we come to our own conclusions about the difficulty of predicting any kind of schedule.

Those who go for longer-term predictions ("futurists") have their own problems. My early grade school years were filled with interesting predictions and fears: gyrocopters for transportation (easy now to see how this wasn't a hot idea), 30 hour work weeks (easy extrapolation—who would have guessed that workers would prefer to work 10 more hours in exchange for more luxuries), fear of people being replaced by computers (well...), and, of course, all kinds of predictions about the space program.

One recent event has heightened my understanding of the difficulty of future prediction. Over the last several years, we have heard about improved eyeglasses and better contact lenses. I don't recall anyone predicting that a cure for near-sightedness (myopia) was possible.

You might have heard of the new operation called "radial keratotomy" (RK), in which incisions change the eye's support structure so that it reshapes itself into the correct form. RK was invented in Russia after a doctor noticed dramatic eyesight improvement in his patient who recently suffered broken glass in his eye.

My roommate, Jeff, is an avid outdoorsman and found his glasses cumbersome and, sometimes, unsafe. Last month his two eyes rated 20/400 and 20/600. Jeff elected to take the risk of RK. Now, his eyes are rated at 20/15. Cool. The cost was reasonable, too.

Knowing the next neat development is always a fine skill (consider its value at start-up companies). It looks like a really difficult task.

The closing date for submissions to the next issue of ;login: is June 28.
General Information

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

;login: (ISSN 1044-6397) Volume 18, Number 3 (May/June 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 ;login:. Subscriptions for nonmembers are $50 per year. Second-class postage paid at Berkeley, CA and additional offices. POSTMASTER: Send address changes to ;login:, USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.

Contributions Solicited

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

Editorial Staff

Rob Kolstad, Editor <kolstad@usenix.org>
Ellie Young, Staff Editor <ellie@usenix.org>
Carolyn S. Carr, Managing Editor and Typesetter <carolyn@usenix.org>
Nick Stoughton, Standards Report Editor <nick@usenix.org>
Bryan MacDonald, SAGE Editor <bigmac@erg.sri.com>
Jamie Marks, Copy Editor and Proofreader

Membership and Publications

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

Copyright © 1993 USENIX Association. The Association acknowledges all trade references made herein.

;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
Lake Forest, CA 92630
Telephone: 714/588-8649
FAX: 714/588-9706
Email: <conference@usenix.org>

Tutorials

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

Supporting Members

ANDATA
Frame Technology, Inc.
Matsushita Electrical Industrial Co., Ltd.
Mtxnix
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 <board@usenix.org>.

President:
Stephen C. Johnson <scj@usenix.org>
Vice President:
Michael D. O’Dell <moo@usenix.org>
Secretary:
Evi Nemeth <evi@usenix.org>
Treasurer:
Rick Adams <rick@usenix.org>
Directors:
Eric Allman <eric@usenix.org>
Tom Christansen <tchrist@usenix.org>
Lori Grob <grob@usenix.org>
Barry Shein <bzs@usenix.org>

USENIX Association

Executive Director
Ellie Young <ellie@usenix.org>
Participants in the third USENIX Mach Symposium enjoyed cool sunny weather at an elevation of roughly 7000 feet in Santa Fe, New Mexico during the third week of April. Over 250 researchers and system programmers who work with Mach attended the day of tutorials and two days of technical presentations, attendance similar to that of the previous gathering held in November, 1991 in Monterey, California.

The initial day of tutorials offered useful instruction in various aspects of Mach 3.0, from general introductions to the microkernel and using its IPC services, to porting issues and the principles of constructing external user-mode memory managers.

Tuesday, April 20

The program chair, David L. Black (OSF) noted in opening that this was gratifying, given that not too many years ago the entire population of Mach-knowledgeable people could (and often did) fit into one room.

Sape Mullender (U. of Twente, Netherlands) began the keynote address by discussing the requirements for an operating system (OS) to support research activity. Support for fast prototyping is necessary and in turn requires that the OS be configurable, have a short learning curve, and hence enable wild ideas to be tried quickly. It should also impose minimal overhead and allow accurate measurements while still supporting realistic loads and testing in an environment of day-to-day usage. Real-time Mach provides a set of mechanisms and policies for implementing real-time servers. His group has been working on server models (and system support for them) providing real-time resource management, improved preemptability, and priority inheritance. They have found that the necessary predictability and analyzability have a cost, and that a fair part of it comes from the mismatch of some existing facilities, such as the Mach interface generator (MIG), to real-time needs.

Sape outlined his current project, a multimedia platform consisting of compute nodes, storage nodes, cameras and workstations with displays, all connected through a modular ATM switch, joking that ATM stands for “Awfully Tiny Messages.”

The first group of four papers discussed some of the structural issues of implementing OS personalities:

Tatsuo Nakajima (CMU) talked about the problems of writing real-time servers which ensure predictable service response times. Real-time Mach provides a set of mechanisms and policies for implementing real-time servers. His group has been working on server models (and system support for them) providing real-time resource management, improved preemptability, and priority inheritance. They have found that the necessary predictability and analyzability have a cost, and that a fair part of it comes from the mismatch of some existing facilities, such as the Mach interface generator (MIG), to real-time needs.

Paul Roy (OSF) presented work done to extend UNIX file services into a multicomputer environment. Several data structures from traditional UNIX implementations (such as open file structures, vnodes, mount points) are represented by ports. A distributed token passing scheme is used...
to provide atomicity, with callbacks from fileservers used to revoke possession as necessary. Several attributes are cached locally for efficiency: file size, seek pointer, and the accessed and modified bits (the actual dates are managed by the fileservers). Reading and writing are implemented using \texttt{vm\_map} in conjunction with \texttt{bcopy}.

Paul then outlined the disk block reservation mechanism used to support UNIX ENOSPC semantics. This led into a discussion of ways that Mach might better support UNIX servers; the problems all stemmed from the decomposition of services into the server/microkernel structure, with the looser coupling leading to loss of information across these boundaries. An example of this is the use of a page fault to bring in the contents of a mapped file: the fault handler has no hint about the intended size of the access, and must deal in single whole pages, making it harder for the server to keep the file size coherent, and slowing large accesses.

Jay Lepreau (U. of Utah) outlined the work done by his group to mitigate the costs of modularity and protection afforded by the traditional microkernel design. Applications pay a substantial cost in performance due to the traversing of multiple layers of software, various protection domains, and the use of an overly general and powerful RPC mechanism. Jay’s group has designed a mechanism to load trusted server code into the microkernel, allowing more efficient communications between the user program, server and kernel. They have eliminated the use of RPC between the application and server if on the same node, using a trap instead. These enhancements have all been hidden behind the interface generator, in the stubs generated by MIG; the same server binary can run in or out of the kernel, and old programs continue to run, though without the improved performance. Though further optimizations and RPC short-circuiting remain to be done, the preliminary speedup is promising.

Simon Patience (OSF) presented work done in Grenoble, France to remove the emulator from the OSF’s microkernel architecture. Originally, the emulator was used to provide compatibility for BSD binaries which made trap-based system calls that weren’t understood by Mach. Other performance optimizations also crept into the emulator over time. Most problems with the emulator stemmed from it living in the address space of the process. This foreign entity wasn’t understood by debuggers or their users, and the emulator couldn’t be trusted by the server since malicious or errant code could overwrite it. This caused the server to grow ever more complicated, since it had to verify every request from the emulator, and much functionality was duplicated in both places. Removing the emulator while remaining at least performance neutral with the older version required several enhancements to Mach. A generalized and more efficient exception mechanism was added to avoid having to make multiple RPCs to the microkernel for each system call. This exception mechanism was then used to build more efficient system call redirection into Mach. Copyin/out functionality had to be changed to directly read/write the process’s address space, rather than relying on information in the request RPC. \texttt{Vm\_read()} and \texttt{vm\_write()} were enhanced to deal with requests that were not page aligned and not an integral number of pages, and to avoid the use of out-of-line data in some cases. Simon showed the new design to be at least performance neutral, though a number of key optimizations have not yet been done.

After lunch, the next block of three papers discussed interprocess communication:

Hilarie Orman (U. of Arizona) talked about implementing Mach interprocess communication (IPC) abstractions over networks, using the x-kernel. She described the protocol stack design, including the detection of node failure using the Easter Algorithm: when a node is reincarnated (i.e., rebooted) the other nodes infer that it must have died at some time in the past. She presented timing data showing the efficiency of the x-kernel design, and pointed out the flexibility inherent in the layering of their protocol.

Kenneth Koontz (John Hopkins U.) told of his efforts to use Mach RPC in an application involving data acquisition, requiring large numbers of small messages. Not surprisingly, he found the performance to be lacking. His solution was implemented purely as a layer of user-mode code wrapped around \texttt{mach\_msg}(); \texttt{Port\_Buffers}. Many messages are packaged together into one large message, which is sent when the buffer is full, when a timeout expires, or by explicit user request. He improved remote message delivery rates by an order of magnitude, local rates by a factor of three, and pointed out that further improvements and tuning should better the results.

Michael Ginsberg (CMU) found that XI1 performed abysmally when it used sockets implemented by the server on top of Mach 3.0. He described the work necessary to present the traditional UNIX interface to X clients, while switching the underlying implementation to Mach IPC or shared memory. He found that by using native Mach IPC he could get performance slightly better than that of a monolithic kernel, but that the
use of Mach shared memory primitives provided speedup in excess of 40%.

The last group of three papers presented on Tuesday revolved around threads and timers:

Stefan Savage (CMU) discussed improvements made to support increased precision in the measurement of time, synchronization, and scheduling. While working on real-time threads, they found the usual timing and synchronization primitives to be lacking; more precision was needed, and the relative nature of time deltas coupled with scheduling uncertainties gave them motivation to invent two new abstractions: clocks and timers. Timers are active objects which allow users to synchronize with time in a variety of ways. Clocks are devices which measure the passage of time and support the use of timers to a particular degree of accuracy. Together, these new abstractions allow various highly-accurate hardware devices to be easily integrated into a Mach system and accessed in a consistent and useful way.

Paul Barton-Davis (U. of Washington) presented the work performed to add scheduler activations to Mach 3.0. Traditional user-level threads packages are implemented on top of kernel threads, and suffer from various problems ranging from poor performance to incorrect behavior due to the mismatch between the two. Blocking kernel operations such as I/O, page faults and processor preemption are difficult for the user-level code to deal with, since much of that information is hidden inside the kernel. Scheduler activations are an alternative to kernel threads that support user-level management of parallelism by providing upcalls allowing the kernel to notify the user-level scheduler of system events that affect the job. Processors are allocated to jobs by the kernel, the user-level thread scheduler controls which threads run on a job's allocated processors, and the user-level scheduler notifies the kernel of changing demand for processors. In return, the kernel never time-slices scheduler activations, and gives the user-level scheduler a chance to deal with events that are normally invisible to it. All of this is hidden behind the C threads interfaces. Paul presented compelling evidence for the increased run-time efficiency this provides to applications.

Randall Dean (CMU) presented his work in using continuations to improve the performance of the C threads library. He found that aggressive saving of thread state before entrance into the C threads package, coupled with flattening of the C threads locking hierarchy increased throughput and decreased context switch latency substan-

tially. In response to questions from the audience, Randy said that his work was complementary to the previously presented work on scheduler activations, not mutually exclusive.

Wednesday, April 21

The first group of papers presented in the morning were from efforts at IBM to support multiple concurrent OS personalities on Mach:

Guy Sotomayor, Jr. (IBM) presented work done at IBM in Boca Raton, Florida and Austin, Texas to make Mach device drivers more general, modular, and to run them in user-level tasks. He stated that the current Mach device drivers, derived from the BSD device drivers used in Mach 2.5, have been widely regarded as one of the weakest features of the system. IBM has followed up on work done at CMU to run device drivers as user-level tasks, and also revamped the device-driver model to separate device access from device operation so that multiple device drivers can cooperate in managing a single piece of hardware. They plan to support multiple cooperating OS personalities on a single platform. The whole scheme is built on a message-based protocol for invoking device-driver services.

Ravi Manikundalam (IBM) spoke about their implementation of Multiple Virtual Machines (MVM), on Mach 3.0. In order to be able to use the extremely large body of legacy software built for PC DOS systems, they have constructed emulation for Windows 3.0 and 3.1, and DOS, including its protected-mode interface. This work required some extension of the Mach microkernel including a revamped fast exception mechanism and a virtual memory interface to allow portions of an address space to be reserved (to avoid having Mach plunk out-of-line messages into an inconvenient area). This paper is one of several to point out the utility of improvements in Mach exception handling and MIG RPC. Though further work remains to be done, both Windows 3.0 and 3.1 work and versions of DOS from 3.3 – 5.0 boot and run.

James Arendt (IBM) spoke about creating an OS/2 flavored server. This work also used the VM reservation interface; otherwise, the Mach VM model seemed a good match for OS/2, which maps some facilities on demand, using guard pages to detect the need to grow a stack or attach to a shared memory region. The Mach scheduler, however, lacks the variety of capabilities of the OS/2 2.x product, requiring them to fold their mapping of scheduler features into a functional subset.
The last group of papers for the morning session concerned memory management:

Inshik Song (Seoul National U.) talked about work done to reduce page-in delay time by prefetching pages based on a task’s page fault history. When a page is requested from the vnode pager, the history buffer is consulted; if another page was requested following the previous time the page was faulted in, the pager prefetches that page in anticipation of its use. Page prefetching was shown to yield good results in most cases, but future work will investigate reducing the prefetch overhead, and evaluate moving the prefetch logic into the microkernel, for closer coupling with VM internals.

Khien-Mien Chew (U. of Texas, Austin) spoke about a collaborative effort to get closer coupling between database management systems and the kernel’s virtual memory support. Databases generally would prefer to manage their pages directly, avoiding the overhead of having the kernel page beneath them, since the database has more accurate knowledge of its own requirements. However, the kernel cannot page directly to/from the database file, since it knows nothing about the consistency and ordering constraints required by the database to be able to cope with system failures. The solution implemented was to allow the kernel to page directly to/from the database files, but to provide flush-locks for intra-page consistency, and give the kernel knowledge of page flush ordering constraints (“page-flush before rules”) to control database consistency.

Philippe Bernadat (OSF) presented work done to adapt Mach to run on machines lacking page-based virtual memory support, such as the segmented Cray vector supercomputers or transputer-based real-memory systems. Mach was designed with the assumption that the underlying hardware includes a page-based memory management unit, enabling the use of a large, sparse virtual address space. Philippe described the machinations gone through to eliminate the use of submaps and minimize or eliminate fragmentation by coalescing address map entries, perhaps extended over what are logically holes in the address space. Copy-on-write cannot be used to lazily evaluate memory copying, because a real memory system cannot assume that write references can be detected and responded to. Performance numbers showed that lack of virtual memory is not always a handicap (every feature has its cost) and that where performance was worse, the extra copying required completely accounted for the slowdown.

The first group of papers after lunch were about distributed systems.

Miguel Castro (INESC) presented “MIKE – A Distributed Object-Oriented Programming Platform on Top of the Mach Microkernel” which allows the use of C++ much the same way it would be used in a nondistributed system. The platform supports fine-grained objects which can be invoked in a location transparent way, and which are potentially persistent. MIKE supports the abstraction of one-level store; persistent objects are transparently loaded on demand when first invoked and saved to disk when the application terminates. The platform also offers distributed garbage collection of nonpersistent objects.

Dejan Milojicic (U. of Kaiserslautern) presented the design and implementation of task migration on top of the Mach microkernel. This was done at user-level in the interest of portability and flexibility, though some modifications to the kernel were needed. The actual migration is accomplished by:

- Suspending the task and aborting the threads to get clean state
- Interposing the task/thread kernel ports
- Transferring the address space
- Transferring threads by getting local and setting remote state
- Transferring the capabilities
- Transferring the other task/thread state
- Resuming the (now remote) task

This work also supported various memory copying strategies, such as eager copying, flushing, and copy-on-reference. Further research will center on load distribution and distributed scheduling problems.

Mark Swanson (U. of Utah) talked about the Schizophrenic Workstation System, which distributes processing load among a network of autonomous workstations in a non-intrusive manner. Besides the technical problems of implementing migration and deciding when to balance load, they set out to build in a solution to the sociological problem: owners of workstations want response-time guarantees, and want the illusion that the machine is entirely theirs to use as needed, allowing them to plan their work around the predictability of computation. This
was accomplished by depressing CPU priority and constraining competition for physical memory by Schizo processes, and by migrating a task to another host when the current host load becomes too high.

The final paper was by Michael Kupfer (UCB) talking about his work to produce a Sprite server. The difficulties encountered involved dealing with exceptions in the server from failed copy-in/os, and problems with signal delivery. The end result was 22% smaller than an equivalent Sprite kernel, and contained almost no machine-dependent code. Though the result ran at 38% of the speed of native Sprite, Michael indicated that many of the reasons for the slowdown were understood and could be fixed.

After all of the papers had been delivered, several people gave ten minute briefings on their work in progress:

Dejan Milojicic plans to investigate using network paging and IPC traffic information to improve load distribution decisions. Initial efforts will be based on periodically disseminated statistics by all nodes involved.

Brent Welch (Xerox PARC) spent his time-slice giving a shameless plug for Sprite, finally tossing the Sprite distribution CD he’d brought with him out into the audience.

Jose Rogado (OSF) spoke about ongoing work on the Cluster Port File System, intended to be a replacement for the existing distributed multi-computer file system, optimized to minimize the number of messages passed.

Mihael Bushnell (FSF) talked briefly about the Gnu Hurd server, confident that his user-level implementation of signals would pass VSE testing quickly. He declined to estimate a possible release date, since “software folks never know how fast they’ll work.” Many of us have yearned for that luxury.

Jay Lepreau spoke about his work in flexible system structuring, designed to get optimum system performance via intelligent and persistent loading of programmable executable objects. Nearly maximal buzzword usage, too!

Santosh Rao (USC) sketched out his ideas on resource management for distributed virtual systems, concentrating on scalable techniques, and coping with heterogeneity through the abstractions of job managers and node managers.

Franklin Reynolds (OSF) outlined recent work done to implement untyped IPC, an attempt to gain performance by simplifying the representation of messages to avoid kernel involvement in having to read and parse each message in its entirety. Some common cases were improved by also performing an eager copy on transmission, page stealing if the deallocate bit is set, and by overwriting an existing buffer (avoiding zero-fill faults for new memory).

---

**A Changing of the Guard**

by Jeffrey S. Haemer
USENIX Standards Liaison
<jsh@canary.com>

Many of you saw the call for applications for a new USENIX Standards Report editor. In fact, many of you applied. Well, after many weeks of black smoke, white smoke is finally rising from the chimney of the USENIX office in Berkeley. We have a new standards report editor: Nick Stoughton (rhymes with “Stoughton”). For history buffs, he’s the fourth in this job, making him ***Shane_McCarron.

Nick tells me that the earliest UNIX version he felt really comfortable with was 6th Edition UNIX, though he did spend time working on 5th Edition UNIX before V6 was available. He also lays claim to having been part of the first 4.1BSD port in the United Kingdom; this latter is surely related to his being English, which means that Carolyn Carr and Sean Eric Fagan, curators of ;login: and comp.std.unix, may have to begin running “spell -b.” You can write to Nick at <nick@usenix.org>.

It is my official job, at this point, to offer Steph Walli thanks from USENIX for a job truly and genuinely well done. I’m personally sorry enough that he’s retiring that I tried at least once to get him drunk enough to agree to continue.

I urge readers who’ve appreciated Steph’s work as much as I have and who wish to stay in touch, to continue writing to him at <stephe@usenix.org>.

The Snitch Editor is dead, long live the Snitch Editor.

May/June 1993 ;login: 7
Good Policy Is Stronger Than Asbestos: What To Do When Your Users Court USENET Flames

by Hal Pomeranz
QMS, Inc.
<pomeranz@imagen.com>

If you have been responsible for a USENET feed for any period of time, chances are that one of your users posted inflammatory material to one or more newsgroups. The Internet is a big place and chances are good that somebody will be offended by every post made. However, we will consider cases here like the recent anti-Japanese polemic that was posted to nearly every newsgroup or the recent posting to rec.pets.cats advocating feline torture. The general response by the USENET community to such postings has been to (a) mail bomb the individual and/or (b) complain loudly to the site administrator that the poster should have his or her privileges revoked. As a system administrator caught in the crossfire, what should you do?

You could do nothing. At the very least this will mean much more careful monitoring of your mail spooling area so that all of that hate mail doesn’t overflow your partition, but that will die off in a few weeks – maybe. On the other hand, the poster is a representative of your institution. It may appear that your organization is tacitly condoning the poster’s actions if you refuse to act. In fact, your organization may have a harassment policy or nondiscrimination statement that may require that some action be taken against the poster.

You could take the frontier justice approach: lock the user out of his or her shrink wrap account, cancel the article (if possible), and post an apology (or force the poster to do it to get the account back). This will quell the hate mail quickly. (It will take up to a week for the apology to propagate out, though.) The downside is that you have taken responsibility for determining that a certain person originated the posting (remember that it is relatively easy to forge news), passed sentence, and delivered punishment. Good for you if you as a system Administrator have that much power in your site (and do you have any job openings?).

Your safest bet may be the “common carrier” tactic that the phone companies use. The phone company must ensure that the obscene phone callers get a dial tone and that their call is connected to your number when they dial it. It is not their responsibility to hunt the caller down and prosecute, much less punish them, when they harass you. Similarly, as a system administrator, it is your responsibility to ensure that email, news, and other data gets to where it is supposed to go, and not to legislate content. It is probably not even feasible for you to do so, but leave that aside.

This does not mean that your job is over once your news feed is working properly. Somebody has to be ultimately responsible for news and email that originates from your site or you’re back to the “do nothing” position. The answer, unfortunately, is something that technical people usually dread: management. Here I mean management in a generic sense – at an educational institution this usually translates to a Dean of Computing or Head of Department for cases like this. Simply saying, “It’s management’s responsibility,” and abrogating any technical input, however, is another way of asking to get your news feed shut off or seriously curtailed.

You need to educate your managers on the benefits of news as well as the problems, and work with them to prepare in advance a complete policy for news at your site. Such a policy should probably include a statement of which newsgroups you will carry (“all,” “all comp and sci groups,” “no alt groups,” etc.), whether or not your users will be allowed to post (and to which groups), an acceptable use statement, possibly the text of a disclaimer to appear in all news articles originating from your site (“the opinions expressed herein...”), and a policy on what to do when somebody from your site posts that “Hitler was right” on soc.culture.jewish. Make sure that all your users sign a copy of the policy, so there will be no question of “I didn’t know.”

As an example, here at QMS our news policy is simple. We carry every group, subject to disk space limitations. (I can’t carry the binaries groups and have to expire news rather quickly because my spooling partition is cramped.) Any user may post to any newsgroup. Questions of acceptable use are decided by the user’s manager. In the event of an inflammatory posting, all mail about the posting is forwarded to the manager in charge of the person who appears to have posted the message. (Remember those forgeries.) I also
reply with a form letter to each person who complains:

We have received your message regarding message ID <insert message ID here> which was apparently posted by a user at our site on or about <insert date/time>. Your message has been forwarded to the appropriate manager for consideration. Any further correspondence should be directed to <manager's name/title/email>.

In the event of an incident, the system administrator tends to be called in as an "expert witness." You will probably have to explain the possibility of a forged message (the seriousness of this threat is determined by the technical expertise of the users at your site and amount of "bad blood") and to produce logging or accounting data to prove or disprove that a certain user was responsible for the posting (your logs are complete and secure, aren't they?). The posting will probably offend you as much as everybody else, but don't let that cloud your technical judgement.

Ridiculous postings appear on the USENET all the time. If one originates from your site, the head-in-the-sand approach reflects badly on your institution and the lack of response only inflames the already offended. Reacting too fast and without authorization can end up getting the system administrator in more trouble than the poster. A policy prepared in advance with the full authority of your management and the consent of your user community will enable you to deal with these situations in an expedient and relatively painless fashion. Ultimately, though, this is an issue for management to resolve, not the system administrator.

More Laws of System Administration

by Paul Evans
SRI International
<ple@erg.sri.com>

Steve Simmons and Elizabeth Zwicky are fond of collecting laws of system administration. Well, I have a new one for them to add to their collection.

If users are made to understand that the system administrator's job is to make computers run, and not to make them happy, they can, in fact, be made happy most of the time.

If users are allowed to believe that the system administrator's job is to make them happy, they can, in fact, never be made happy. Furthermore, in their quest for happiness, they will cause enough resources to be diverted to trying to make them happy that the computers will no longer run.

My first job as a system administrator was in a group with a very strong technical lead. I will call him Don, because that was his name. Don had red hair and a temper to match. He wore the kind of moustache made popular by Josef Stalin. He had a cobra tattooed on one arm. He had hanging in his office a print of Frank Frazetta's "Dark Rider" (a barbarian horseman holding a bloodied battle axe), to which our boss had added a cartoon-style caption saying, "the System Administrator is in ...."

Don did not have what pop business literature would call a "total customer service attitude," and in that regard set the tone for our whole group. He expected users to defer to him, and they did. He had, and was seen to have, total support from the management of the company for which we worked. On the one occasion that a user challenged his authority — he said that Don should look for a new line of work — the user was fired the same day. Because it was made absolutely clear to everyone that Don's job was to make the computers run and not to make the users happy, the computers ran, and the users were happy.

After Don left the company (which sued his new employer in an unsuccessful attempt to get him back), our group was incorporated into a unified engineering services group, which included drafting and document control. Because the other engineering services were explicitly in the business of making their customers happy, users began to draw the thoroughly false and pejorative analogy that the end and highest good of system administration should likewise be user happiness. As more and more system administration resources were poured into the voracious maw of user support in a futile attempt to make the users happy, the reliability of the computers eroded badly, which made the users unhappier still. After fighting this downward spiral for a year, both remaining full-time system administrators resigned in the same week.

User happiness is an important outcome of effective system administration. Paradoxically, however, it is only possible to provide a level of service that will keep users happy when they've been explicitly told that making them happy is not what system administrators are paid to do.
Itty Bitty Boxes All Made Out Of Ticky-tacky
by Elizabeth D. Zwicky
SRI International
<zwickyl@erg.sri.com>

The delivery guy brings me a package from Bodacious Software, Inc.

Knowing that the new version of their WhizBang product is out, I indulge in the following daydream: I take my keys out of my pocket, pick out one that I don't care about, and use it to rip open the top of the box in one smooth motion. Opening the box, I take out a piece of corrugated cardboard on top, and underneath it is the software distribution. It is not shrink wrapped. If it is a CD, it's in a jewel box; if it's a tape it's in a normal tape box. It is not a floppy. The front and spine of the box, and the front of the CD (or the front and spine of the tape) say, in big clear letters "WhizBang Version 2.3 for Sun 4." This is the entire identifier; there are no more coded release numbers to find. If there's a serial number, it's on the media. The most important release notes and installation information are on an insert in the case with the media. Underneath the media is an installation manual, not shrink wrapped, no assembly required. It says "WhizBang Version 2.3 Sun 4 Installation" on the front and the spine. Underneath that is a user manual, labelled "WhizBang Version 2.3 Sun 4 User Manual" on the front and on the spine. It's not shrink wrapped. It is fully assembled and ready to use. The box is now trash; there are no serial numbers and no important information stuck to it anywhere.

What happens when I stop daydreaming and actually open the box? Well, not that — not even once. First off, when I rip the key across the top of the box, half the time it doesn't work. Oops, it's a Sun CD box, and the flaps don't meet in a straight line; better get out the knife and feel for where and how it jogs. Double-oops, there's nothing between the top of the box and the CD, and I've just blunted my knife and scratched the top of the CD (but hey, it makes it much easier to get the shrink-wrap off.) Or maybe it hasn't got a join across the top at all, and I need to figure out which piece of tape is covering up the flap. Better yet, maybe it's Solaris 2.0, which looks like one of those, but there is no flap, and no matter what you do to the top of the box, short of disembowelling it by going right through the cardboard, you can't open it from the top. We used those boxes as a spectator sport, giving each one to a new person and seeing how long it took them to give up on trying to open it from the top and decide to do it "wrong" and rip the bottom open. The winner took over five minutes and mere words cannot capture the expression on his face when he discovered that in fact the designer intended you to open the box from the bottom.

Once I have passed the intelligence test and gotten the box open, we proceed to the test of dexterity and patience. The people who design these things appear to be for the most part frustrated designers of matrioshkas, those nested Russian dolls. Inside the box, possibly nestled in styrofoam peanuts, there is usually a shrink wrapped box. Stripping off the plastic allows me to remove a lightweight glossy slipcase (generally strengthened by an inner or outer corrugated cardboard sleeve). Inside that there is a heavy cardboard slipcase. Inside it are a bunch of shrink wrapped manuals.

If I'm lucky, there's also a separately shrink-wrapped CD in a jewelbox. CD packaging is one of those areas where innovation is popular, though. If I'm not lucky, the CD is inside one of the shrink wrapped manuals. This appears to be a thoughtful move on the part of the vendor, until you realize that I loan manuals to users. I don't want to loan my media to users. Not only are there lots of them who never return them, sometimes they take them home — a not unreasonable thing to do with a manual you want to read — where my media are exposed to the uncontrolled influences of children and pets. I do not want to call vendors to ask for replacement media because their CD was eaten by an iguana. Tyvek envelopes are popular, but impossible to store logically, particularly when innovatively attached to other things. Then there are innovative cardboard boxes that are almost the shape of a jewelbox; it's a real pity that they're about an eighth inch larger than the slots in our CD racks.

Of course, if I'm really unlucky, the manuals are in the form of shrink wrapped binders in which are two separate shrink wrapped sets of pages — the actual text and the index tabs for it. I am expected to assemble these components. A really innovative software company — Hewlett-Packard, for instance — may provide me with an assortment of manuals, in different sizes and bindings, only some of which need to be assembled. (In Hewlett-Packard's case, they accompany this with a packing list that includes item numbers not only for the binder, the binder contents, and the binder labels, but also for the box that the packing list is in, and the labor it required to pack the box.) Or, going for maximum mystification, they may give me unlabeled binders and
slipcases, and put the labels elsewhere. There are two major theories for label distribution; you provide all the labels together, where they’re easy to get at but hard to match to their destinations, or you put them with the documents. The preferred method for providing the labels with the documents is to shrink wrap something (the document or the index tabs for the document), optionally shrink wrap the labels, and then shrink wrap the two together. Now, I don’t need to unpack all the user manuals; that can be left until somebody wants to read them. On the other hand, with the unlabelled binders, I have to open binders until I figure out which one is the one I want. By this point, I'm getting a little hasty about stripping off shrink wrapping, and small items like labels are flying wildly around the room. (This could explain all those slipcases with post-it notes stuck to them because nobody can find the labels.)

On the outer box, there is at least one packing slip (frequently two, one from the software company and one from the shipping company). Inside the box — maybe shrink wrapped to things, maybe floating around loose somewhere or other — are 5 or 6 other pieces of paper and cards. Some of them are valuable coupons, generally for things I don’t want, but occasionally for things that I am passionately interested in. There’s a piece of paper that either has a license, or a way to get a license; there’s a piece of paper that tells me how to get support. There’s frequently yet another packing slip. These little pieces of paper are now inextricably mixed up with the backing for the labels, the slips that were inside the shrink wrap to tell me which labels match which documentation which match which index tabs, and any pieces of paper I might happen to have had already. They will all have to be sorted out and either thrown away or filed.

I have now spent, on average, about 15 minutes getting the software out of the box. My office is covered with removed shrink wrap, extra pieces of cardboard, and small paper shreds. I have sustained at least one paper cut, and I am beginning to feel a bit sullen about the entire enterprise. I decide that perhaps I’ll install the software tomorrow, and put the software carefully on the shelf. I can’t put the CD away until I find a jewel box to put it in so that it will fit on the rack. I have to reassemble the manuals into the slipcase(s) before I put them back. After another 5 minutes of hassle putting all the pieces on the shelf, in the tape or CD rack, or in the trash, as appropriate (after carefully reading all surfaces of the apparently discardable packaging, becomes sometimes the serial number is stuck to them), my phone rings. It’s a user, who wants to know why he can’t use WhizBang 2.3 yet, when they say they shipped it to us last week. When I finish explaining to him that shipping is not an instantaneous process, and that software does not install itself, the phone rings again. It’s our shipping department; they have a pallet containing our other 39 copies of WhizBang 2.3 (shrink wrapped, of course), and they want to know where to put it.

I then go home, where my friends and loved ones start asking questions like “How can you dislike a piece of software when you haven’t even tried to install it yet?” “Did you know that your pants are covered in little tiny cardboard shreds and there’s plastic stuck to your hair?” “How can an otherwise intelligent person want to discuss cardboard boxes?”

The next day I go to get the package down from the shelf and install it. First I have to find it. We have an entire wall and a half of manuals. About a third of the manuals we own are spiral-bound with no name visible once they’re on the shelf. Another 10% are paperbound but have blank white spines, or spines with only the company logo on them. Of the remaining manuals, under 10% have the version number and the platform printed on the spine. At least 10% don’t have it on the cover either; you can usually figure it out by pulling all the manuals that might be of interest and comparing the copyright dates. So pulling the manuals off the shelf may end up being another several minute challenge. If the installation instructions are in the user manual, I’m in real trouble — the user who’s dying for the upgrade will borrow the manual, and I’ll have to track it back down to get the piece I need.

The CD spine may be equally uninformative (especially if the CD came in an envelope and we had to transfer it to an unlabelled box). In some cases, the version information may be on the CD in coded form — oops, I meant to install the version with the rabbit on it, but instead I reinstalled the one with the kangaroo, when we had been running all the way up at penguin. Of course, the version numbers on all three CDs are identical — it’s just the picture that differs. Or, there may not be a CD. Quarter-inch and half-inch tapes are no more problematic than CDs. They don’t stack as well, but they’re harder to lose. (Those little boxes fit under things real well, not to mention that people in a hurry tend to take CDs out of carriers without putting them back in boxes, leaving forlorn little stacks of naked CDs sitting in corners.) The vendor who shipped us its UNIX software on 40 3.5 inch floppy disks, on the other hand, presented whole new problems. There’s nothing like
misplacing floppy #17 of 40 to really enliven your morning, and floppies fit under, behind, and between objects even better than boxed CDs.

Now that I have succeeded in getting the software out of the box, onto the shelf, and back off the shelf, all I have to do is actually install the software. Oh, and open those other 39 boxes. I actually have a theory for that; I bring the whole stack to staff meetings, and we open them while we meet. This does add a certain odd note to the proceedings—it’s not just the background noises provided by tearing cardboard, people struggling to remove shrink wrap, and boxes being thrown into corners, it’s having everybody armed with knives.

SAGE: Solaris Developers Conference Report

Solaris Developers Conference
Santa Clara Convention Center
March 29–31, 1993

Elizabeth D. Zwicky
SRI International
<zwicky@erg.sri.com>

I was sent to represent our division at the Solaris Developers’ Conference, not because I have any pretensions of being a Solaris Developer, but because our division has several projects that are going to be using Solaris machines, and the system administration group is the only group involved with all the projects. Out of our group, I was chosen to go by the usually combination of bribery, threats, and being out of the room at the critical moment.

As a nondeveloper, I wasn’t expecting a lot; in fact, I brought a computer with me so that I could work while I listened (making me one of only two people sitting in a Sun conference using an Apple Macintosh PowerBook — surprisingly, nobody mentioned it to me, although I did hear people behind me snicker occasionally). Actually, I got a lot out of the gathering, after a very unpromising start.

As a USENIX conference goer, I found the makeup of the attendees startling. The ratio of men to women was even higher than I’m accustomed to. (I’m used to there being a line for the men’s room, but none for the women’s. I’m not used to there being a line for the men’s room, and nobody but me in the women’s room.) The ratio of Sun employees to attendees at large was also startling; not only were all relevant developers imported for each talk, which was good, but every talk had a row of other Sun people standing in the back, which was a little weird. The first day, the ratio of press and managers to actual developers was very high, as well — that’s because the first day of the conference consisted primarily of an extremely glorified press conference.

The press conference, which started late and ran long, contributed highly to the unpromising nature of the kick-off. When a conference is running 30 minutes late before the first event, and over an hour late by the end of it, it makes me nervous. The live band, the scripted all-VP, almost all-male presentation, and the overly cute live demos all made me feel even more nervous. Only one of the demos failed, and I considered that one to be poetic justice; the moment the VP made a joke about his wife that was so sexist that he got hissed by the audience, the demo stopped working, and never did restart. (After that, he stuck to Bill Gates jokes, which provoked occasional yawns but no hisses.)

From a system administrator’s point of view, the kick-off was a mixture of good and bad news. There was a section of it devoted to system administration, but it was entitled “The First Hour: Installation and System Administration” and this concept that installation was pretty much all of system administration was pushed hard. There was a lot of hoopla about Sun’s desire to simplify system administration so as to reduce the cost of ownership, and a demo of AdminTool which consisted primarily of putting up a text window and saying “Isn’t that ugly” and then putting up AdminTool and saying “Isn’t that pretty.” This was followed by vague promises and improbable claims. For instance, the claim was made that you could now do all administration remotely, and that Solaris would cut down the number of needed UNIX system administrators by a factor of 5, from 1 per 100 machines to 1 per 500 machines. Estimating this ratio for our site is a little tricky, since we have part-time system administrators, and we don’t separate UNIX system administration from PC and Mac administration. But my best guess is that we currently run 200 UNIX machines with 5 full-time equivalents, throwing some doubt on their initial assumptions.
The kick-off was followed by more vice-presidents with marginally less glitz (they lost the band and most of the stage set, but kept the giant wall-o-monitors). I only saw Adobe's version, which relied heavily on yet another video clip, but it was actually relatively funny. It did appear to assume that the entire world consisted of PC networks which were either not administered at all, or were administered by incompetents; I considered it an advertisement for real administration as much as for Adobe's software. A good PC administrator could have solved most of the problems shown without resorting to Adobe's software (an interchange format and associated programs); a network administered as badly as the one shown would be unlikely to be able to make use of Adobe's software. But it was a funny video.

After the first day, the conference settled down to business. Most of the press and managers went elsewhere, and the presentations were given by project managers and senior developers who for the most part actually knew what they were talking about, and who had a lot of help to fill in when they didn't. Unfortunately for me, somebody had carefully gotten them all to do overheads in exactly the same format and display them using SparcStations and Island Presents. If they'd let everybody pick their own format, at least some of them would have picked a truly high-contrast color set and a font big enough to read. Instead, everybody used a navy blue background, yellow and white lettering, and nothing but the title font was large enough for me to see. Aside from that, the technology was surprisingly reliable; I only saw one presenter crash the display program, and one who put up a slide only to discover in horror that only half the text he'd tried to put on it appeared on the screen.

The talk on AdminTool is adequately summed up by the answers to half of the question-and-answer section afterwards: “No, it doesn't do that.” “No, we have no plans to release that.” “That does sound useful; maybe in a future release.” “Well, we're planning to create another version of AdminTool that will use ToolTalk, and we have to redo all those interfaces then, so we're not documenting them now. No, we have no schedule for doing that.” “No, we have no plans to do that.” “No, there are no hooks for running custom scripts when you add users, but you can edit the default dot files. No, I don't think you can add more files for it to install in new accounts.” “Well, you can do that from the command line, but it's a little baroque; it's really much easier to write a C program.”

Sun is clearly still scrambling with the change to Motif; the biggest laugh anybody got all week was the poor presenter on OpenWindows 4 (I'm sorry, I mean "Sun's next window system product, currently internally code-named OpenWindows 4"), who ended with a summary in which he reiterated Sun's commitment to OpenLook. He meant Motif; his slide said Motif; half the back row practically shouted "MOTIF" the moment "OpenLook" slipped from his lips; but he was undone by years of practice defending OpenLook.

My favorite speaker was Bruce Tognazzini. Part of this was the fact that he was the only speaker with no illegible overheads, which is the sort of nicety one expects from a human factors expert. Also, I was cheering for him to begin with; after all, he has caused Sun to produce instructions for how to write an “Open” dialog box that actually state what order things should be displayed in. I think it ought to specify the same order ls uses, which it doesn't, but I can live with any order at all as long as every application I have to deal with uses the same one. We have Wingz, IslandDraw, and Frame, and they use 3 different sort orders. I'm all for adventure, but not when I'm trying to open a file. In any case, he's an enjoyable speaker in the UNIX tradition; colorful, opinionated and imbued with a True Religion.

Overall, I would go back again, but I think on day one I'd register and then quietly go do something else until the furor died down. On the truly frivolous side, the snacks were good if sometimes peculiar (frozen candy bars?), the toy pickings were slim but high quality (there was a nice floating pen), and the location (the Santa Clara convention center) was excellent. (OK, I'm biased, since I live 10 miles away, but the number of out-of-towners who spent their breaks out soaking in the sunshine seems to indicate that it isn't just me.)

---

Open Meeting with SAGE Board of Directors at Cincinnati

Meet the SAGE Board of Directors at the Summer USENIX Conference in Cincinnati, OH. Help shape the future of SAGE. Bring your questions and ideas to this open meeting on June 23, from 6 – 8 pm. Check the BOF Board for location.
SAGE-AU Inaugural Conference and Annual General Meeting
University of Melbourne, Australia
July 7–9, 1993

Preliminary Announcement

The System Administrators’ Guild of Australia (SAGE-AU) will be hosting a conference in conjunction with its inaugural annual general meeting. The theme of the conference will be “Administering Networked Computers.”

With the coming of age of computer networks, more and more organizations have internal networks of machines sharing information. Administration of these machines involves solving far more complex problems than for standalone computers. System administrators are under increasing pressure to allow more and more interconnection of machines without any loss of reliability or security.

Conference Details

SAGE-AU’93 will be a 3 day conference running from July 7–9, 1993. The first day will be dedicated to tutorials on tools and techniques to aid system administration.

The inaugural AGM will be held at the end of the second day.

All other times will be allocated to presentations. A conference dinner will be held on the second night.

The conference will feature a small trade show focusing on system administration tools.

Tutorials

Tutorial sessions will be either half-day or full-day duration. People wishing to present tutorials should submit an abstract and a preference for a half-day or full-day slot to the address below. Tutorials should be run in a lecture format.

Attendance at tutorials will cost $100 for a half day and $200 for a full day.

Registration and Fees

Registration forms are available from the address below. The registration fee for SAGE members is $100. Nonmembers may register for $150.

The registration fee does not include the conference dinner. Registrants must indicate whether they wish to attend the conference dinner on the registration form and pay an addition cost. In keeping with the low-cost, informal nature of the conference, the conference dinner may be held in a local restaurant, depending on numbers.

The registration fee does not include attendance at tutorials.

SAGE foundation membership is currently being offered for $30 ($20 annual subscription plus a $10 joining fee).

Annual membership fees have yet to be set by the committee.

Addresses

Send all enquiries to regarding the conference to:

Peter Gray
Professional Officer
Dept. of Computer Science
University of Wollongong
N.S.W. 2500 Australia
Phone: +61 42 213770
Fax: +61 42 213262
Email: <pdg@cs.uow.edu.au>

Requests for general information about SAGE-AU and membership applications should be addressed to:

Frank Crawford
Australian Supercomputer Technology
Woods Centre
PMB 1
Menai
N.S.W. 2234 Australia

or emailed to <sage-info@mel.dit.csiro.au>.
Overview

As usual, projects have been plentiful and life has been hectic. I hope your winter has been enjoyable. A lot has happened since the last issue of "What's Out There?" We now have a new President of the United States, Bill Clinton, who seems to be somewhat willing to entertain technological efforts. It's unknown whether all those efforts will be good, but it's definitely a start in the right direction.

In this issue, you'll learn how to find out what's happening within the U.S. Government (including the U.S. Budget!), at least as far as the PR people will allow (the good part); where to find discussions of the Clinton administration's recent support of a NSA - developed encryption chip (the bad part); information on Factsheet Five - Electric, a 'zine of 'zine reviews; and a change (yet again) in the NIC's location.

U.S. Government Information

It is now possible to obtain White House press releases electronically. Also included in these archives are speeches made by the President and other administration officials. This is definitely a step in the right direction from the new administration. You can obtain press releases via a variety of methods - USENET newsgroups, WAIS, gopher, and electronic mail.

The press releases from the White House Communications Office are available on the following USENET newsgroups:

- alt.politics.clinton
- alt.politics.org.misc
- alt.politics.reform
- alt.politics.usa.misc
- alt.news-media
- alt.activism
- talk.politics.misc

The press releases are also available via both WAIS and gopher from

sunsite.unc.edu

If you don't have a WAIS or gopher client, you can access the information by telneting to sunsite.unc.edu – login as the following users:

- For WAIS, login as “swais” (without the quotes, of course)
- For gopher, login as “gopher”

For more information on WAIS, see "What's Out There?" Vol 1, Number 1 in the May/June 1992 issue of this newsletter. Gopher will be discussed in a future issue.

You will be accessing the general WAIS and gopher servers, respectively at sunsite.unc.edu. You can login as the user “politics” to search (via WAIS) various U.S. politically related information. You can also view and search the 1994 U.S. Budget via both WAIS and gopher at sunsite.unc.edu.

If you choose to access the top-level gopher server at sunsite.unc.edu, look under the “Sunsite Archives” section for the press releases, U.S. Budget, and other related information.

The press releases are also on CompuServe, America OnLine, the WELL, MCI, Fidonet, Peacenet, and Econet.

The above are the preferred methods for accessing the White House press releases. If need be, you can also obtain the press releases via email. The current email server is a simple automagic response mail server; it is not designed to handle letters, comments, or requests for specific information. To access the mail server, send email to:

<clinton-info@campaign92.rg>

with a Subject line of:

HELP

To sign up to automagically receive press releases, use a Subject line that starts with:

RECEIVE

followed by one of the following categories:

- ECONOMIC POLICY
- FOREIGN POLICY
- SOCIAL POLICY
- SPEECHES
- NEWS
- ALL

Other commands that you can include on the Subject line are:
STATUS:
list the groups you're signed up to receive

REMOVE category:
stop receiving whatever category you list

WAIS:
to obtain a WAIS query form to search the press releases

Read the HELP message you receive to learn more about this mail server. You can also obtain the press releases via anonymous ftp from the following sites, at least:

sunsite.unc.edu /pub/academic/political-science/white-house-papers
ftp.cco.caltech.edu /pub/bjmccall
cpsr.org /cpsr/clinton

The White House Electronic Publications and Public Access Email Frequently Asked Questions (FAQ) list should be available at each of the archive sites for the White House press releases.

The FedWorld BBS, operated by the National Technical Information Service, also makes the press releases available. You can reach The FedWorld BBS at +1 703/321-8020, with communication settings of 8 data bits, no parity, and 1 stop bit.

Email to the White House

Another cool thing that is slowly starting to happen is sending email to the White House. This system is still being set up and will take some time to fully develop. There's a lot of thought and work that needs to be done to do this properly. But, it is a start. Right now, it is not possible to receive electronic responses to your email letters. So, if you do send email to the White House, include your U.S. Postal address for replies. In the future, there will hopefully be a way to get electronic responses, also. Use one of the following addresses to send email to the White House:

<clinton-hq@campaign92.org>
<75300.3115@compuserve.com>
<clintonpz@aol.com>

You can also send email to the White House directly from CompuServe, America OnLine, MCI, and Fidonet.

All of this definitely seems to be a step in the right direction towards electronic access to the U.S. Government. Sure, it doesn't cover all the different areas where we might want access, such as FOIA and other publicly-available governmental records, but it is a good thing to start. Check it out!

White House and NSA (Encryption) Clipper Chip Announcement

On April 16, 1993, the White House announced the development of an encryption chip for voice communications developed in conjunction with the National Security Agency (NSA) called the Clipper Chip, along with an initiative regarding telecommunications and privacy which could literally affect almost every citizen in the United States. On the same day, AT&T announced a “secure” phone which incorporated this chip. Some important things to point out:

- The encryption algorithm is remaining classified. In the cryptography community, an encryption algorithm is only considered secure after it has been examined extensively and independently by a wide array of experts around the world. With an algorithm which is kept secret, there is no guarantee that it is secure and that the encryption method has no “back door” (allowing easy decryption for those, such as the NSA, that know the “back door”)

- Although the government has announced plans to use the chip in their own phones, they do not plan to use it for classified information, only for unclassified information.

- This chip has been in the making for 4 years. It would seem that the Clinton administration has already made plans to use the chip, without public comment or discussion on a matter which is so important to the privacy of that same public.

- It would seem that the Government might be granting a monopoly to Mykotronx, Inc. and VLSI Technology. It’s unclear whether each company makes the entire chip or just parts thereof.

- The key, which allows the information encrypted with this chip to be decrypted, is embedded in the chip. This means that once the key is known, the chip needs to be replaced to maintain private communications. This would usually mean replacing the entire device (e.g. telephone), anytime that the key was divulged, whether legally or not. The key is also transmitted along with your encrypted data, so that law enforcement can obtain it, which would allow them to decrypt your data without your knowledge.

- The 80-bit key is split into two (2) 80-bit pieces and kept in databases at two different escrow agencies. It’s not clear how the key databases will be kept secure. It is also unknown if the classified encryption algorithm is any less
secure to brute-force attacks, once half the key is known.

A successor chip has already been announced, called the Capstone chip. The Capstone chip is supposed to be a “superset” of the Clipper chip and will include the “digital signature standard” (DSS), which many in the cryptography community seem to consider insecure, as I recall. The NSA also developed DSS, which wasn’t disclosed until CPSR filed a FOIA request with NIST (the National Institute of Standards & Technology).

This announcement, in one way, is a step in the right direction – privacy and encryption technology are important to the general public and for international economic competitiveness. An inquiry on whether export restrictions on encryption technology is good or bad is also a good thing. Currently, companies that want to include encryption as part of their products need to make two versions – one for domestic distribution and one for international distribution.

On the other hand, there are too many things about the announcement which are bothersome and need to be discussed publicly.

Both the Electronic Frontier Foundation (EFF) and the Computer Professionals for Social Responsibility (CPSR) have made public statements against the announcement. The CPSR has filed Freedom of Information Act (FOIA) requests regarding the plan.

Online discussions of the announcement have been occurring all over the Net in various USENET newsgroups and mailing lists. Here’s a sample of where you might find discussions of the Clipper Chip:

USENET newsgroups:
alt.privacy.clipper
sci.crypt
alt.security
alt.privacy
comp.org.eff.talk
comp.security.misc
comp.society.cu-digest
comp.risks

Mailing lists:
<cypherpunks-request@toad.com>

Also, check the archives for the various groups listed above, as things may have changed by the time this newsletter comes to print in June 1993.

The official White House press release of the Clipper Chip can be found via anonymous ftp from:

csrc.ncsl.nist.gov in /pub/nistnews

or via the NIST Computer Security BBS at +1 301 / 948-5717. It should also be available with the rest of the White House press release archives mentioned above.

The EFF comments were first published in the EFFector Online Issue 5.06, which is available via anonymous ftp from:

ftp.eff.org in /pub/EFF/newsletters

Information from CPSR is available online via anonymous ftp from:

ftp.cpsr.org in /cpsr

The cypherpunks mailing list also maintains an archive. Information on the Clipper Chip can be found via anonymous ftp from:

soda.berkeley.edu in /pub/cypherpunks/clipper

Please read the announcement of the Clipper Chip encryption technology, think about it, and discuss the implications of this with your friends, congresspeople, and anyone else.

**Factsheet Five – Electric**

*Factsheet Five* is a hardcopy magazine which reviews ’zines. A ‘zine is typically a small press publication, many times put out by individuals. The quality of ’zines, as with most things, varies greatly. And, that’s where *Factsheet Five* comes in. Within *Factsheet Five*, you’ll find reviews of ’zines on every subject imaginable. Issue #46 had 1,460 reviews. They’re hoping to break 2,000 for issue #47. *Factsheet Five* is highly recommended; you’re liable to find at least something of interest in each issue. And, in the worst case, you can find out about the wide variety of stuff that people write about.

So, why am I telling you about a paper publication? For one, it’s a worthwhile publication to pick up. And, second of all, there’s now *Factsheet Five – Electric* (F5-E), an online version of the reviews in *Factsheet Five*. You can access F5-E via:

- The WELL, a public access computer system; for more information, call +1 415/332-4335. You can also connect to them from the Internet via well.sf.ca.us.

- Anonymous ftp
  - from ftp.msen.com in /pub/newsletters/Zine-Reviews/F5-E
Miscellany

Call for Summarizers
by Rob Kolstad
@login: Editor
<kolstad@bsdi.com>

I'm very excited about an idea of publishing microsummaries and micro-reviews of various journals, magazines, technical rags, and other widely distributed technical media for @login:. I think it would be of tremendous benefit to our members, and would allow them to keep up with a headline-oriented summary of recent technical news.

If you're a subscriber to a technical journal (e.g., JACM, etc.) or a magazine or tabloid (CACM, Open Systems Today, Byte, EE Times, VAR Business, etc.) and would volunteer to summarize their exciting points, please contact me at <kolstad@bsdi.com>. I'll be happy to share a much-expanded documentation of this idea with you. I anticipate that writing the summaries would take only a small amount of time to prepare per issue of any periodical you already read. Of course, appropriate recognition and fame would be accorded to volunteers.

Correction

Peter H. Salus reports: In the January/February 1993 issue of this newsletter I inadvertently omitted crediting Tony Mason as one of the authors of the lex & yacc book in my column, "The Bookworm."

So, start using rs.internic.net to search for sites and people registered in the directory. On a UNIX-like system which has the whois program, you can tell whois to search the new NIC via:

whois -h rs.internic.net NAME

where NAME is what you're searching for.

What's Next?

In future issues, I will discuss such topics as travel-related archives, astronomy, gopher, and whatever else comes to mind.

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

Jeff Kellem is, among other things, a musician, a programmer/developer, an Internet explorer and consultant, a system administrator, and occasional writer. He is the founder of Beyond Dreams, an organization which promotes information exchange and communication. He can be reached via email: <composer@beyond.dreams.org>.

Annual Board Meeting

The Annual Meeting of the Association will be held during the Open Meeting of the Board of Directors which is scheduled for June 22, 1993 from 7–8:00 pm at the Hyatt Regency Hotel in Cincinnati, Ohio (site of the USENIX Summer '93 Conference).

Errata A La Binderia

As some of you may have already noticed, the bindery made an error in some copies of the March/April issue of @login:. The missing pages are 23–30 and 47–54. If you haven't looked at your copy yet, please do. Let us know if you want to receive a new and complete copy of the issue. Please send your requests to <office@usenix.org>.

Curly Quotes

Some readers have expressed their annoyance with the quotation marks that are used in the typesetting of this newsletter. The double quotation marks currently used here are Palatino's "curly quotes." The Palatino family of type does not make standard curly quotes. Theirs are angled and only appear "curly" when enlarged to a high degree. Thus, they appear to be incorrect to some eyes. This may cause the typographically-sensitive among us to rethink the fonts used in this newsletter. We'll see... Carolyn Carr, Font Maven

18 @login: May/June 1993
At the 1993 World Conference on System Administration, Networking, and Security (SANS II), I spoke briefly during the tips and techniques discussion about the "universal" automounter, AMD. This article, along with two future ones, will present the features and techniques used here at the Numerical Aerodynamic Simulation (NAS) Facility, compare the features to Sun's automounter, and most importantly, offer locations where more information can be obtained.

One major design principle followed at the NAS is to port a common set of source files to the architectures that the resulting binaries will use. Since we had no wish, or legal right, to port Sun's automounter to SGI IRIX, Convex ConvexOS, or Cray Unicos we had one alternative: a publicly available automounter. Our current environment includes AMD on Sun Sparc (SunOS 4.1.1 or later) and SGI (IRIX 4.0.5 or later), and the future holds Cray and Convex.

For the 400+ workstation at the NAS, the workstation team maintains six Sun 4/490 fileservers that serve both the Sun and SGI architectures. Going back to the "common source" principle, the fileservers and all workstations have one map file which states which machines they mount and which machines are "backups" should their primary go down. A primary could go down in either of the following ways: the server itself is off-line (for whatever reason) or the network link to the primary server has broken. In both cases, after a certain (configurable) delay time, a machine will notice that it cannot obtain its primary server, and — through heuristic logic built into the mapfile — decide which machine to contact next.

The map files contain logic designed around three parameters: the client's architecture (SGI or Sun), the architecture's subnet, and a "group." The subnet might be any one of the many subnets in our various buildings; the group is configurable. When a primary server is unavailable, AMD attempts to mount a server on the subnet that exports that client's architecture.

The "group" is a controlling group; it is one of a set that determines which lines in the mapfiles AMD looks at when mounting a remote partition. Here at the NAS, workstations in the AMD "admin" group mount fileservers read-write, while the AMD "workstation" group mounts fileservers read-only. Both group entries and their respective options are in the same mapfile; the machine "group" determines which line in the mapfile to consult.

A key feature in AMD's maps is the management of a machine using itself as a server. Three of our fileservers export the TeX and clients partition for the workstations and other fileservers. If a user on the fileserver wants to run TeX, the fileserver may have it locally or the fileserver may have to mount it from a remote machine. In the local case, AMD knows that it does not need to mount; rather, it creates a symbolic link from the "mount" area to the locally installed TeX package. In the latter, AMD mounts another fileserver in the "mount" area. This reduces unnecessary NFS activity and is easy to administer.

Since our installation is large and presents great difficulty for management teams to notice whether or not a client's AMD was/is having difficulties, AMD uses syslog through a local channel. This enables centralized administration, logging, and record keeping for AMD's performance in our environment. AMD also has the capability to log through a logfile (avoiding syslog) which, in some cases, will prove more appropriate.

The last key feature in AMD is its capability to mount different machines in different trees, yet still maintain a "common" point of access. Case in point: when a fileserver goes down, the old NFS partition is unmouted and the new one is mounted. The mount locations for the two (old and new) are different, which avoids problems with stale filehandles, problems during the unmout, and other anomalies. However, though both machines mount the filesystems in different places, a symlink will automagically change allowing the transparency we needed.

My next article will discuss the important features lacking in Sun's automounter and how AMD answered our needs. Watch the next issue of this newsletter.

You may obtain AMD at the usc.edu ftp repository in /pub/amd. You may join AMD's discussion list at <amd-workers-request@acl.lanl.gov>.
The TCP/IP Bakeoff

by Pace Willisson
<pace@bsdi.com>

The 1993 TCP/IP Bakeoff was held the week of February 8th at the offices of FTP Software in North Andover, Massachusetts. The Bakeoff brought TCP/IP developers from many companies together to hash out real world compatibility problems with the low level parts of TCP/IP implementations and with the basic applications such as telnet, ftp, mail, and printing. While there have been many testing events for the higher level standards such as SNMP, NFS and X, it has been 12 years since anything similar has been held for TCP/IP itself.

The organizers of the event were very successful in creating an environment of cooperation among competitors. The participants agreed informally not to disclose details of the testing results in ways that would help or hurt individual companies. Therefore, everyone in the room felt free to help one another. Every vendor left the event with ideas for improving its product, and those who brought source code were able to create and test fixes on the spot.

The bugs uncovered ranged from system crashes caused by pathological packets to garbled screens caused by invalid assumptions of terminal capabilities. Some testing was aimed at finding out which options make a pair of systems work best together, rather than merely determining whether a particular option was properly implemented. For example, one tester set out trying to discover which system-level commands were needed to establish an eight-bit clean telnet connection to every other system.

The testing lab consisted of a large room with three or four tables for each vendor. There were two main networks for testing, the “stable net” and the “battle net,” both implemented as thin coax Ethernet. The stable net was connected through a router to the Internet so that the participants could reach their home machines easily. The battle net was isolated and was used to test reaction to pathological packets. Each participant was asked to bring an MS-DOS machine, and FTP Software provided MS-DOS-based packet sniffing software. When a question arose concerning what was transmitted on a connection, it was easy to get a definitive answer from a nearby machine.

On the stable net, each vendor tried using the servers of every other vendor. Some testers simply looked for basic functionality, such as logging in with telnet, displaying files, interrupting processes and so on. Others wrote elaborate scripts that exercised each ftp command. After the first few days, every server had been tested in a dozen different ways, while every client had faced a dozen different servers.

The battle net was used to see how the different hosts responded to unusual packets. Most battle net tests were created by capturing a regular packet, then using a packet editor to create various test cases. Then another packet editing program created a list of packets, one targeted to each test machine. Finally, a central machine repeatedly transmitted the packets on this list.

Battle net tests included ARP packets with broadcast sources and illegal fields, IP packets with invalid header lengths, and strangely-fragmented IP packets. Surprisingly, one battle net test revealed that half of the vendors didn’t check the IP version number field. This is likely to cause trouble as new protocols are introduced.

In addition to generating ideas for product improvements, the Bakeoff spurred many ideas for improving future Bakeoffs. The battle-net testing concept was refined into practice during the event, and one battle-net test was written from scratch during the week. Testers also discussed ideas for making multiple-packet battle-net tests, such as a test that could open a series of TCP connections and take each one through a different set of closing states. Detailed descriptions of the tests that were run will be placed on anonymous ftp servers by the organizers (once they catch up with their regular work).


All of the participants were pleased with the results of the Bakeoff, and planning for another one has already begun. For more information write to <bakeoff-request@tgv.com>.
I saw the “tenth” anniversary map of USENET in the November/December 1992 issue of this newsletter.

USENET was first announced at the Boulder USENIX meeting in January, 1980. The first software was released at the USENIX Delaware meeting in Summer, 1980 and there were about 15 sites at that time (sorry, no maps).

Having UC Berkeley join the net in late 1980 or early 1981 was the big catalyst. The explosive growth started about then.

Here are some older maps than the ones previously published. Sorry, I have no idea who to credit (Bill Shannon maybe? Tom Truscott maybe?). I saved them away a long time ago.

It’s fun to note that USENET was running on the Internet in June 1981.

It’s weird. I still recognize almost all of the site names!
Big Iron

by Gunter Ahrendt
<gunter@tartarus.uwa.edu.au>

[Editor’s note: Gunter Ahrendt posts his list of the
world's most powerful computing sites every week.
This is the April 13, 1993 posting. I find it fascinating
that so much computing power exists.]

BT ratings are from the NASA AMES NAS CFS
BT Benchmark Report of May, 1992, ratios are to
a Y-MP/1, others are Manufacturer’s Peak fig¬
ures, ~ denotes approximations, ? denotes
guesses. Sites are listed in order of available BT
figures.

Planned installations are listed with dates follow¬
ing equipment in brackets. Ratings for “pro¬
posed” equipment and similar, i.e., [1Q94], are
not included in the total ratings for a site until it
has actually arrived “on-site.”

111.64 - (19-JAN-1993)
National Security Agency, Fort Meade, Mary¬
land, US
1) 4 * Cray Y-MP-C90/16-1024 111.64 BT

~+101.65 - (26-JAN-1993)
Cray Research, Minnesota, US
1) 27 * Cray Y-MP/2 46.98 BT
28) 2 * Cray Y-MP8/8-128 13.9 BT
30) 2 * Cray Y-MP4/4-64 6.96 BT
32) Cray Y-MP/8-32 6.95 BT
33) Cray Y-MP8/6-128 5.21 BT
34) 3 * Cray X-MP/2 ~3.69 BT
37) Cray 2/4-128 ~3.08 BT
38) 2 * Cray 2 ~3.08 BT
40) Cray X-MP/4-8 ~2.46 BT
41) Cray Y-MP2/2-1024 1.74 BT
42) Cray Y-MP2/2-32 1.74 BT
43) 5 * Cray Y-MP/EL2 1.65 BT
48) Cray 2/1-16 ~1.54 BT
49) Cray 2/1 ~1.54 BT
50) Cray X-MP/2/4 ~1.23 BT
51) Cray A-PP 6,720 Mflops Peak

~+79.95 - (19-JAN-1993)
Los Alamos National Labs, New Mexico, US,
<dwil@acl.lanl.gov>

1) TMC CM-5/544VU 52.8 BT
2) 2 * Cray Y-MP8/8-128 13.9 BT
3) 2 * TMC CM-200/64K ~3.84 BT
4) 2 * Cray Y-MP8/4-128 6.96 BT
5) Cray X-MP/4-16 ~2.45 BT
9) 16 * IBM RS/6000-560 1,600 Mflops Peak?
25) Cray T-3D/256 [1Q94] 40,960 Mflops
Peak?

~+64.76 - (13-JAN-1993)
Lawrence Livermore National Labs, Liver¬
more, California, US, <moe@jette.ner$c.gov>

1) Cray Y-MP-C90/16 27.91 BT
2) Cray 2/8 ~12.3 BT
3) Cray Y-MP8/8-128 6.95 BT
4) Cray Y-MP8/8-64 6.95 BT
5) Cray 2/4 ~ 6.15 BT
6) Cray X-MP4/4-16 ~ 2.45 BT
7) Cray X-MP/2-2 ~ 1.22 BT
8) BBN TC2000-128 .66 BT
9) BBN TC2000-32 .17 BT
10) 18 * IBM RS/6000-550 1,320 Mflops
Peak?
28) Cray T-3D [2Q93] 40,960 Mflops Peak?

~42.06 - (19-JAN-1993)
Minnesota Supercomputing Center, Minne¬
sota, US <consult@msc.edu>

1) TMC CM-5/544VU 28.05 BT
2) Cray 2/4-512 ~ 6.15 BT
3) Cray Y-MP/4-64 ~ 2.45 BT
4) Cray X-MP4/EA64 ~ 2.45 BT
5) Cray Y-MP92/2-1024 2 BT
6) TMC CM-200/32K ~ .96 BT
7) Cray Y-MP-C90/8 [2Q93] 13.96 BT

35.27 - (19-JAN-1993)
Pittsburgh Supercomputing Center, Pitts¬
burgh, Pennsylvania, US
<balog@bandit.psc.edu>

1) Cray Y-MP-C90/16-256 27.91 BT
2) Cray Y-MP/8-32 6.95 BT
3) TMC CM-2/32K .41 BT
4) Cray Y-MP-C90/8-128 6.95 BT
5) Cray Y-MP-C90/8-64 6.95 BT
6) Cray T-3D/512 [1Q94] 81,920 Mflops Peak?
7) Cray Y-MP-C90/8 [2Q93] 13.96 BT

34.86 - (10-JAN-1993)
European Center for Medium Range Weather
Forecasts, Shinfield Park, Reading, England

1) Cray Y-MP-C90/16 27.91 BT
2) Cray Y-MP/8-64 6.95 BT
-34.1 - (29-JAN-1993)

National Center for Supercomputing Applications, University of Illinois, Champaign, US <consult@ncsa.ncsa.uiuc.edu>

1) TMC CM-5/512 25.4 BT
2) Cray 25/4-128 ~6.15 BT
3) Cray X-MP/4-8 ~2.45 BT
4) TMC CM-2 .1 BT

-18.51 - (29-JAN-1993)

Numerical Analysis Simulator, Ames Research Center, NASA, Moffett Field, California, US <jet@nas.nasa.gov>

1) Cray Y-MP8/8-256 6.95 BT
2) TMC CM-5/128-32/256 6.6 BT
3) 2 * Intel Gamma iPSC/860-128 3.82 BT
5) Cray 2 ~1.54 BT
6) Intel Paragon L-15 [1Q93] 2.6 BT?
7) Cray Y-MP-C90/16 [1Q93] 27.91 BT
8) Cray Y-MP-C90/16-1024 [2Q93] 27.91 BT

-15.09 - (MAR-1993)

National Center for Atmospheric Research, Boulder, Colorado, US <jsloan@niwot.scd.ucar.edu>

1) Cray Y-MP/D8 6.95 BT
2) Cray 3/2 ~6.16 BT
3) Cray Y-MP/D2 1.74 BT
4) TMC CM-200 ~ .24 BT
5) TMC CM-5 [2Q93] ~ .83 BT

10.28 - (FEB-1993)

San Diego Supercomputing Center, University of California, California, US <consult@sdsc.edu>

1) Cray Y-MP8/8-64 6.95 BT
2) Cray 3/2 ~6.16 BT
3) Cray Y-MP/64 1.74 BT
4) TMC CM-200 ~ .24 BT
5) TMC CM-5 [2Q93] ~ .83 BT

-10.26 - (25-JAN-1993)

Caltech, Pasadena, California, US <heirich@cco.caltech.edu>

1) Intel Touchstone Delta 512 ~8.04 BT
2) Intel Gamma iPSC/860 64 1.11 BT
3) Intel Paragon 64 1.11 BT?
4) Multicomputer 16K/40 40,960 MFlops Peak?
5) Multicomputer 512/32 1,024 MFlops Peak?

~10.25 - (19-JAN-1993)

Oak Ridge National Lab, Tennessee, US <hks@ornl.gov>

1) Intel Paragon XP/S-35 6.07 BT?
2) KSR1-64 3.31 BT
3) Intel Paragon XP/S-5 .87 BT?
4) 6 * Cray A-PP [1Q93] 40,320 MFlops Peak
10) Intel Paragon XP/S-150 [1Q94] 26.02 BT?

8 - (31-JAN-1993)

Arctic Region Supercomputing Center, University of Alaska, Fairbanks, Alaska, US <floyd@ims.alaska.edu>

1) Cray Y-MP-M98 8 BT

7.95 - (07-JAN-1993)

Navy Supercomputing Facility, Stennis Space Center, Missouri, US <finnegan@invader.navo.navy.mi>

1) Cray Y-MP8/8-64 6.95 BT
2) Cray Y-MP 1 BT

7.95 - (29-JAN-1993)

Ohio Supercomputing Center, Columbus, Ohio, US <oschelp@osc.edu>

1) Cray Y-MP8/8-64 6.95 BT
2) Cray Y-MP 1 BT

7.57 - (29-JAN-1993)

KFA Research Center, Julich, Germany

1) Cray Y-MP 8 6.95 BT
2) Cray X-MP 62 BT
3) Cray Y-MP-M94 [1Q93] 4 BT

~7.42 - (03-FEB-1993)

Center for Theory & Simulation in Science & Engineering, Cornell National Supercomputing Facility, Ithaca, New York, US <consult@tc.cornell.edu>

1) KSR1-128 ~6.62 BT
2) iPSC/860-32 .56 BT
3) TMC CM-200/8K ~ .24 BT
4) IBM ES/9000-900 2,664 MFlops Peak
5) 16 * IBM RS/6000-550 1,173 MFlops Peak

[800 more lines available on request]

If you have any changes, additions or deletions to make, please mail them to <gunter@uwa.edu.au>. You can remain anonymous if you wish.
An Update on UNIX-Related Standards Activities

by Nicholas M. Stoughton
USENIX Standards Report Editor
<nick@usenix.org>

This report is somewhat shorter than usual, not least because as login: goes to press, the IEEE POSIX Spring meeting is taking place, and few of our regular reporters have had time to put finger to keyboard. Stephe Walli, standards report editor for USENIX for the last year, has stood down, and a new fool has been found, too innocent to realize what he was taking on, in me, Nick Stoughton.

I am unashamedly British, and if some of the spelling, punctuation, and use of English looks strange to you, that's because it is correct! Otherwise, I hope none of you notices much change in the style of what is reported here. Having been in the UNIX field since the early days of 6th edition (remember the Lions book?) in 1976, I have been involved with almost every aspect of the operating system, both from an academic point of view, at Queen Mary College (as it was then) of the University of London, and latterly in a commercial services point of view, with Hoskyns Group, also in London.

First Impressions

However you look at it, Stephe is a hard act to follow. Those of you who have been following his comments recently on Test Methods and Language Independent Specifications will know what I mean!

The next installment in these great battles dominated the April meeting in Irvine for many.

As a newcomer on the scene, many people asked me about my first impressions of a POSIX meeting. Imagine two great armies facing each other across a battlefield. Rather than let each side attack the other till one is the clear winner, the whole encounter is far more formalized, more like a game of chess. Of course, this drags out the battle, but is much fairer (and of course politically correct!)

How many standards committees does it take to change a light bulb? Forty two ... one to specify the mechanism for light bulb exchange, one to write a language-independent specification, one to develop test methods, a screw-cap language binding committee, a bayonet-cap cap language binding committee, a subgroup on bulb-exchange methodology, a working group on bulb disposal, then of course there's real-time bulb exchange ... hey, this sounds like a great new area for POSIX!

LIS Wars III

The real battle of the week was the Language Independent Specification (LIS) issue. In January, an Ad Hoc Committee, chaired by Stephe Walli, was established to research the whole issue of LISs. The Ad Hoc had spent considerable time between the meetings canvassing opinions, and it was clearly a hot potato. Some countries really believed them a major issue, even if it took twenty years to get there! On the other hand, the real users of POSIX standards, people like you and me, want standards as soon as possible, and we want them written in a language that we can go ahead and use.

The Ad Hoc met frequently during the Irvine meeting, and was well attended. All views were aired, and a consensus reached. Even writers of existing bindings in other languages (FORTRAN and Ada) recognized that the existing, non-LI, methods of deriving their work had worked, and worked well. So a very carefully worded report was constructed to present to the SEC. With Stephe chairing it, what would you expect?

It recognized that the goals of a LIS were good, but that the tools were not yet appropriate for the task. I have lost count of the number of times in my working career where I have been asked, metaphorically, to use a wrench to undo a screw. LIS is just another wrench, a programming language version of Esperanto. Indeed, although it wasn't suggested, if we are to write a truly language-independent standard, the portions currently in English should be rewritten in Esperanto. Actually, thinking about it, I believe it was suggested once, but never mind. A large amount of time and effort has been spent in POSIX over the last three years trying to get LI to work, but we are still not much further ahead then when we started.

The report went on to make the explicit resolution to drop the requirement for LIS, allowing working groups to use it only if they thought it appropriate. It further proposed a resolution to provide an alternative means of ensuring semantic equivalence between different language bind-
ings. The original version, written in a specific programming language (normally C, but not necessarily), should contain conformance rules for future language bindings.

The SEC debated this report at length. Issues such as what to do with existing LI work, in particular the POSIX.1 LIS work, would need very careful consideration.

The chair of the United States Technical Advisory Group to ISO Working Group 15 (better known as the WG15 US TAG), asked that the decision on LIS be deferred until July. WG15 itself meets in Heidleburg in May, and it was WG15 that placed the LIS requirement on the US National Body in the first place. If the US TAG were to present a unilateral fait accompli, their position would be much weakened. Several members of WG15 had made their views known to the Ad Hoc, and were aware of the likely outcome. It was quite clear that in July, the weight of opinion would be behind accepting the report of the Ad Hoc, and that SEC would cease to require LISs. The US TAG delegation were made aware of this. In the end, by the narrowest of majorities, it was agreed to defer the decision. Whether you view this as three more months, or just another three months depends on you point of view.

It is clear from the falling attendances at the POSIX meetings (down to approx 220 from over 300 last year) that many other people feel that far too little happens far too slowly at these events.

The Little Red Standards Group

The following text appeared anonymously on a notice board at the Irvine meeting during the "great LIS debate." It is reproduced here for your enjoyment. Although it represents one person's view, other conclusions are possible!

"The Little Standards Group"

Once a hard-working little standards group lived with a group of international standards organizations, WG15, SC22, and JTCl. One day they decided to make a standard for them all to share.


When she had cut the interface, she asked "Who will help me edit the standard?" "Not I," growled WG15. "Not I," smiled JTCl. "Not I," snorted SC22. "Then I will do it myself," said the little standards group.

Soon the wonderful smell of a freshly baked standard filled the community. "Now," said the little standards group, "who will help me use the standard?" "Not so fast," cried WG15. "Before we can use our wonderful standard, we must first have an LIS for the specification." "Yes!" cried JTCl. "Absolutely true," said SC22. "I can understand that," said the little standards group, "So, who will help me create this LIS?" "It's your standard, so you must do it," said WG15. "This is reasonable," said JTCl. "Naturally it is your task," said SC22.

The little standards group worked on the problem for a while, but was not able to make progress. So the little standards group went back to the others. "I don't think I can do this LIS thing," said the little standards group. "And I didn't even want to do this LIS thing in the first place. Even though you want it done, you are unwilling to help. Why should I do this thing for you?" asked the little standards group. "You cannot call it a standard until we say you can," said WG15. "Regretably true," said JTCl. "We agree," chortled SC22.

The little standards group said "Fine. Then it will not be a standard for you; it will be a standard for me. When you are ready, I will work with you to create this LIS. Until then, I guess you'll just have to do without." So the little standards group produced useful and portable applications, while WG15, SC22, and JTCl hungrily watched.

Developing Standards in the Absence of Test Methods – or – What if you threw a party and nobody came?

Many years ago, the POSIX gods decided that it was important for POSIX standards to have test assertions. The purpose of these test assertions was to ensure that implementations of the POSIX standards could be fairly evaluated by potential system purchasers. Unfortunately, the task of creating test assertions, much like the task of testing software and hardware in every organization, was viewed as an onerous, menial task that was just not exciting. At the April POSIX meetings, the POSIX gods rescinded their test assertions
requirement because of considerable pressure from the community that develops the POSIX standards. In this article, I will attempt to evaluate the impact of this fundamental change in the way in which POSIX standards are developed. When reading this article, please keep in mind that it is an editorial, and is one person's opinion. There are a number of other, perhaps better informed opinions in this area. With luck they too will write about this topic.

**Why did POSIX develop Test Methods?**

Testing is essential to determining the quality of any product. That is not under dispute. While things like Total Quality Management are currently under close scrutiny to determine their overall benefits, the fact remains that without testing it is all but impossible to be confident that the product you are distributing will work to the satisfaction of your customers.

In the open systems community, testing usually takes place during the development cycle. However, large procurement groups (like the US federal government) may require additional, post-production testing to ensure that systems conform to open systems standards. This “conformance testing” is usually done via some sort of test suite that the procurement group has chosen.

This need for conformance testing within the open systems movement has caused a segment of the test development community to focus on developing tests against open systems standards. In fact, the success of POSIX has caused some testing organizations to come into being. These organizations develop conformance tests against the open systems standards, many times in competition with one another for a very limited market. In order to evaluate the quality of a test suite, a potential test suite customer must have some metric for determining that the suite actually tests the standard in question. Hence, test assertions.

Test methods are, in general, simple statements that describe the behavior of an interface under certain conditions. (Note that this is an over-simplification, but is generally accurate.) By agreeing on the test methods for a given interface, the open systems community in effect gives guidance to test suite developers as to the tests that should be performed to ensure conformance to the interface. As mentioned above, these methods can also be used by the community to evaluate the quality of the tests that the test suite developers produce. This is particularly useful for organizations who, like the US federal government, must use objective metrics to evaluate products prior to procurement. Just as computer systems must be tested to see if they conform to standards, the tests to test the systems must be tested.

**So what's the problem?**

So far this seems pretty straightforward, right? Unfortunately, the model breaks down because of a market that is too small. While the market for open systems is roughly really big, the market for test suites is pretty small. Further, the market for test suites that have to be objectively evaluated against other, competing test suites in the same space is even smaller yet! Consequently, while the market pressure to develop open system standards is very great, the market pressure to develop test assertions is almost nonexistent.

What does this mean? Let’s consider an example that typifies traditional POSIX committee patterns, as opposed to ideal behavior. When POSIX committee A starts their work, they have a base interface specification. There is considerable enthusiasm for the work, and it proceeds quickly (in standards terms). Eventually, the draft standard is complete, and is sent out to a broader group of people for formal balloting and approval as an industry standard. This standard represents an interface that is important to the community, and in particular important to the companies that funded their employees to attend the standards meetings and work between meetings to complete the document. It is, in turn, probably important to those companies’ customers, since the cost of participation is high enough that it usually requires some sort of higher-level management approval.

At this point in the process, the members of the group that did the work are effectively out of the loop. Balloting is a process that does not generally involve group activity. However, there is more work to be done. The standard must have test assertions developed. Therefore, the members of the group that have just completed work on the draft standard need to start developing a new document that is effectively a detailed design description of the document they just finished. Moreover, this description has to be made in a somewhat formal manner, using stilted, precise, language. On top of this, the document they now must develop is one that most of their management and customers don’t need. Consequently, many participants who worked on the initial document suspend their participation until there is some other task in which their sponsor is more interested.

**What was the solution?**

As I mentioned above, that was an example of a more traditional POSIX activity. Recently the POSIX working groups have come to grips with...
this trend by changing their development model to one in which the test assertions are developed at the same time as the interface specification. This results in a higher-quality specification, fewer changes in the specification during balloting, and of course, the test assertions being available in the same time frame as the specification itself. That's the good news. The bad news is that the standard itself takes somewhat longer to develop in the first place, and in the end has all of this stuff in it that wasn't really needed by the majority of the standard's users.

Other groups have addressed the problem by finding some industry group with deep pockets to do the primary development work on the test assertions, and then ratified the work through the working group process, but in a rapidly accelerated fashion. In the three examples of this within recent POSIX memory, the work has resulted in all the benefits of the previous example, but without the disadvantage of taking significantly longer than the development of the base standard. (To be fair, the base standard in these three cases was also developed by an external industry group.)

In both these cases the working groups managed to retain most of their participants throughout the process, and to produce standards and test methods that are reasonably high-quality without delaying the introduction of their standards overly much. So, with the introduction of these more ideal working models, the POSIX community has discovered a way to develop the required test assertions, albeit with some cost to the industry.

What are the implications of not requiring test assertions?

So, what have we come to? Test methods are useful tools for improving the quality of a standard under development. However, test assertions are expensive, and the formal standard version of the test assertions only serves a very small community of "users" — those who develop test suites for open systems standards and those who need to do objective procurement of open systems test suites to use as tools for evaluating other objective procurements.

Given these assumptions, what does the POSIX gods' decision to no longer require test assertion development do to the quality or availability of POSIX standards? My guess: not a lot. POSIX has traditionally provided a useful service, and has not yet succeeded in making itself irrelevant. (Many other standards groups have.) Consequently, POSIX will continue to produce standards for which the market exhibits a need. This development may, or may not, involve the specification of test assertions along the way. Those groups that see utility in developing the test assertions will do so. If the market can agree on the need for test assertions for a given standard, the market may cause those test assertions to be developed. If a large procurement group needs to procure a test suite that tests a standard for which there are no test assertions, it may define the test assertions itself. In any event, the community that has the need will fund the development. In the interim, the standards the market requests will continue to be developed to the level of quality that the industry has come to expect.

Report on POSIX.0: POSIX Guide

Kevin Lewis <klewis@gucci.ent.dec.com> reports on the April 19 - 23, 1993 meeting in Irvine, CA:

Let me say right up front that this was quite a productive week for the group. Our primary goal was to achieve in excess of 75% in our affirmative ballots so as to move into recirculation prior to the next meeting in July. The group was successful in achieving this goal. The chronology is as follows:

- Initial number of affirmative ballots received was 28 out of 58, placing the percent affirmative at 48%.

- The group converted 7 ballots to affirmative prior to the April meeting.

- The group converted 13 ballots to affirmative during the April meeting.

- This places the current percent affirmative at 82% (48 out of 58).

The issue of public specifications, expected to be a highly significant issue, proved to be exactly that for only a small number of balloters (5 out of 58, 3 of whom could be considered negotiable).

The POSIX.0 group conducted a Birds-of-a-feather (BOF) session on this issue of public specifications to ensure that any balloter and any one else with a concern in this area had an opportunity have a dialogue with Dot Zero to ensure an effective exchange of information. Our main concern prior to this BOF was that the way POSIX sees public specs and their use and the way everyone else sees this issue might be at odds.

It became evident after the BOF that the understanding on this was mutual. In fact, there were a very small number of people (3 out of about 20
that attended) who had any major concern with this topic.

The ISO WG15 Rapporteur and the SC22 Secretariat representative were present during the BOF. I queried them on whether or not there was any concern expressed on this issue at their respective levels within ISO. They both indicated that each group was aware of the presence of this topic in the POSIX.0 guide and no one had expressed any concern.

Given that POSIX.0 has reached the goal enabling a recirculation, this will be coordinated with the IEEE with the objective of having this next phase start prior to the July meeting in Denver. The group will be in recirculation during the July meeting. So our discussions will focus on possible future projects, to include a guide in the area of Transaction Processing and an IEEE video conference that would serve as a tutorial about the use of the guide.

The Old Stuff: POSIX.4.

POSIX.4 is the base, real-time standard. This document is so close to finishing, we can taste it! On the last ballot, we achieved an 83% affirmative approval rating. By that metric, we’re done. On the other hand, some of the remaining balloters are vociferous and represent large, existing bases of UNIX functionality. For this reason, in New Orleans the technical reviewers were still addressing the remaining objections, trying to remove as many of these as possible. At the close of the New Orleans meeting, the ballot resolution process wasn’t completed. However, since then, the resolution process has been finished, and we have in fact sent out POSIX.4 for its final recirculation. Two large changes were made from draft 13 to draft 14, along with several minor changes. The major changes are:

1. The real-time files chapter has been removed from the document. This chapter had become the lightning rod for the majority of the remaining objections, most of which objected to the fact that the facility only seemed able to do contiguous, pre-allocated disk files. More capability and extensibility was desired. A competing proposal, for a thing called fadvise, has been made by one of the objectors, and it seems like a better solution to the whole issue of real-time files. We will probably be addressing this area in future work of the POSIX.4 working group. Unfortunately, for now there is no proposal in place for real-time files. I was personally uncomfortable with this action, it being late in the balloting process. At the same time, though, I do like the fadvise interface more than the POSIX.4 Draft 13 interface, and some of the Draft 13 facilities were, frankly, incomprehensible to me. Basically, I think this action was OK.

2. Some of the POSIX.4 interfaces feature the capability to set up “something” that will cause the application to be asynchronously notified in the future. For instance, a timer expiration, or asynchronous I/O completion, both in asynchronous notification. As of Draft 13, “asynchronous notification” meant a signal was delivered. Several objectors wanted the ability to replace signals with other, presumably more lightweight methods of asynchronous notification. (Simply calling a function is a possibility.) Draft 14 has been accordingly modified to support extension to new methods of asynchronous notification. Signals are currently the only known method of asynchronous notification, but other, future
mechanisms can now be implemented in a portable fashion under the POSIX.4 facilities. I am quite in favor of this change, as everyone knows there are faster ways of doing asynchronous notification than signals!

In addition, there are numerous, very minor changes to Draft 14 of POSIX.4. At this writing, POSIX.4, Draft 14 has been sent out for one last recirculation. Draft 14 is not a complete draft; it is merely a set of changes from Draft 13. There are about ten pages of changes. The balloting period has already closed, and we are in the process of totaling up the responses to this last recirculation.

More Old Stuff: POSIX.4a

POSIX.4a is the threads proposal. This document is probably of greater interest to a lot of the USENIX members than POSIX.4 itself. Here’s a shocker: POSIX.4a has gone out for a recirculation! After a long while in ballot resolution, the threads proposal was just recently sent out again. It should be in the hands of the balloting group for another recirculation before this report is published. With any luck, this recirculation will go more smoothly, and more swiftly, than the previous two recirculations have. The good news is, this time, POSIX.4a is being recirculated, not reballoted. The previous time around, POSIX.4a was reballoted. What that means is, the entire draft was open for comment. In a recirculation, by contrast, only the changed parts of the proposal can be commented upon. A reballot is required when not enough people deliver an affirmative ballot. Thus, if you need a reballot, you’re in trouble. The fact that we are recirculating POSIX.4a is a good sign.

Yet more Old Stuff: POSIX.13

The profiles document, POSIX.13, is currently still in ballot resolution. It seems to be following the POSIX.4a model, wherein the technical reviewers have very little time for technical review. The issues for POSIX.13 are fewer than for POSIX.4a – that’s good. On the other hand, the one big issue, subsetting of POSIX.1, is still to be addressed. That’s bad. (For those of you who are just learning about the POSIX.4 world, POSIX.13 consists of four profiles, ranging from supercomputer real-time all the way down to tiny, embedded systems. These embedded systems generally run on hardware that is not capable of supporting all of POSIX.1 and POSIX.2 (no disk, no MMU, very little memory, etcetera). For that reason, there is a need for an embedded profile to call out a subset of POSIX.1: it needs a lot of POSIX.1, but other parts are simply irrelevant or impossible. Stay tuned for more information on how POSIX.13 is doing!

POSIX.4c

Some progress has been made towards a language-independent specification for POSIX.4. However, given the current instability of the whole LI thing, I wouldn’t be surprised if we were able to drop this work later on. For now, work continues on LI.

New stuff: POSIX.4b

POSIX.4b is a collection of fairly exotic stuff: timeouts on blocking functions, direct application access to interrupts and different types of memory, sporadic server scheduling, and so on. The facility that is of interest to the mainstream community here is a thing called spawn. Spawn combines fork and exec into a single call that spawns a new process. This is useful for systems (see above) that cannot support a separate fork (because they have no MMU for duplicating an address space), but which can support separate processes (because they have enough physical memory for it). Spawn is interesting to the mainstream community because, in general, a fork is followed by an exec anyway. Spawn would be a nice way of combining things.

Other interesting work is devctl. Devctl is really ioctl, but we had to rename it so as to not break existing systems.

During the week in New Orleans, small groups addressed various issues in the chapters of POSIX.4b. An interface for various sorts of typed memory (static RAM, different buses to access the same memory, and so on), was part of the proposal, but due to lack of interest and commitment in this proposal, it was removed from the proposal. Other changes were less drastic. We hoped, at New Orleans, to take the document out for balloting after the Irvine meeting. I personally have my doubts that this will happen. I think it will take a couple more meetings to mature the POSIX.4b functionality sufficiently. On the other hand, POSIX.4b has been substantially written (in Language Independent form plus C bindings, event!), so I could be mistaken.

Report on POSIX.9: FORTRAN Bindings

Michael J. Hannah <mjhannah@sandia.gov>, chair of POSIX.9 reports on the January 11–15, 1993 meeting in New Orleans, LA:

The POSIX.9 (FORTRAN bindings) group, though sparsely attended, did meet in January and made progress towards their next project. While other IEEE standards have been drafted by three people, this is uncommon for POSIX. A committee of such small size implies a very significant reliance on the ultimate balloting group. There is nothing wrong
with a small group doing the draft, but before such a draft becomes a standard it must be reviewed and examined by a much larger, representative, balloting group. While there may be nothing improper or illegal with this approach, I would certainly like to see more participation in our meetings.

IEEE Std POSIX.9-1992 is approved and was available for purchase at this meeting. This standard completely defines how to access all functionality of ISO/IEC 9945-1:1990 from FORTRAN 77, as well as defining a generalized way to access any operating system structure and defining byte-stream I/O for FORTRAN 77. Since FORTRAN 77 is essentially a subset of the new Fortran 90 standard, IEEE Std POSIX.9-1992 is completely usable in a Fortran 90 environment. Like any standards committee that just completed a standard, the committee discussed how to convince vendors to implement this standard, and how to convince users to demand this standard from the vendors.

Actual work was begun on draft 0 of the next project for this committee, POSIX.19, which is a binding between the Language Independent Specification (LIS) of POSIX.1 and the new Fortran 90 language standard. This Project Authorization Request (PAR) was approved by the IEEE standards board in Sept 1992, though approved over a year ago by the SEC. Part of the delay was ensuring complete liaison with X3J3, the FORTRAN Language committee. At their August 1992 meeting, the X3J3 approved an official resolution endorsing the scope of the POSIX.19 PAR and agreed to active liaison with the POSIX.9 committee. This is significant since the POSIX.19 PAR includes in its scope the ability to define extensions to the base language standard of Fortran 90. One of the major unresolved objections in balloting IEEE Std POSIX.9-1992 was that many of the functions could have been defined as simple extensions to the base standard syntax. For example, mode bits could be included as an extension to the FORTRAN OPEN statement, etc. Such an approach is planned for the POSIX.19 work.

The committee began by discussing the overall approach to the new work. In addition to examining the new language features in Fortran 90, the committee discussed how this new binding should relate to the POSIX.1 LIS and its companion POSIX.1 C binding. While this POSIX standards body is focused on portability of applications, as an end user I am particularly concerned about portability for application writers. Whether to point to the LIS or point to the “historical practice” of C is a contentious issue. For example, the LIS describes a procedure called change_file_mode, while the traditional C function is called chmod. In IEEE Std 1003.9-1992, because any function/subroutine name in FORTRAN 77 was exposed to the loader, and since the IEEE Std 1003.9-1992 routine to change file mode had to be slightly different than chmod, we had to give it a distinct name, PXFCCHMOD. Because of the new features of Fortran 90, module names used by an application are not necessarily exposed to the loader. Thus we could now call the Fortran 90 routine chmod without fear of conflict with a different C library routine of the same name. Using chmod as the name is intuitive to any programmer coming from the C world, but is not intuitive to a strictly FORTRAN programmer. While the argument in this example may be stronger for chmod since there is also a POSIX.2 command by that same name, such an argument does not apply to all POSIX.1 functions. If you really believe in the concept of the LIS, and especially if the new C binding is “thin,” then a name that is the same as the LIS change_file_mode might be more appropriate. A Fortran 90 bindings reader should need at most the POSIX.19 binding and the POSIX.1 LIS. However, many LIS names are more than 31 characters, so some mapping may be required, or an extension to Fortran 90 to recognize uniqueness in names longer than 31 characters. This seems to be something like a religious issue, where parties on each side are certain that their position is correct, and the only intelligent position. The current committee default is to use the long LIS names, not the familiar C names, but this may change.

One item of interest is that new constructs in Fortran 90 seem to permit the binding to be completely specified as Fortran 90 code fragments. Whether the IEEE and ISO document structures could be accommodated by this is unclear. For an implementation, you would like to give an electronic copy of the binding (code fragments) to the implementors so they could simply add the rest of the code necessary on their implementation. Since the code fragments completely define the application interfaces, this would be a boon for everybody. However, such a scheme also raises the question among the lawyer folk as to what this would mean with regard to the IEEE copyright of the binding.

Finally, the committee was actively involved in the hot debate concerning whether the POSIX Executive Committee should rescind its mandatory requirement for base POSIX standards to be developed as Language Independent Specifications (LIS). As a language binding committee we
are viewed as the direct beneficiaries of the work to produce an LIS. The members of the bindings committee of both the POSIX.5-Ada and POSIX.9-FORTRAN have strong opinions on this issue.

The POSIX.9 committee is scheduled to meet the first three days of the April POSIX meeting.

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

*Paul Borman <prb@cray.com> reports on April 19 – 23, 1993 meeting in Irvine, CA:*

**The Reduction Continues**

At the April meeting in Irvine, the POSIX.14 group dedicated one day to the POSIX.18 Draft. It was much easier to work on the draft this time, mostly due to its reduced size. As before, the major work done on the draft was reducing the number of words.

First of the areas we attacked during this meeting was the introduction and scope. We decided that even though the content was basically hidden in there somewhere, we would do best by just rewriting the introduction and scope instead.

The next section of the document looked at was the definitions section. After reviewing the definitions of conformance, we moved them to the conformance section of the document. We also removed several definitions that were either defined in one of the base-level standards we reference, or were actually simply definitions of English words, such as "portability."

In the actual normative text, we moved some of the pieces in the "Language Interoperability" section to our "Coherency" section. This was done to clarify and not change content. The only actually substantial change was to remove the FIPS 151-2 requirements for CS7, CS8, CSTOPB, PARODD and PARENB, which were added at the last meeting. It was brought to our attention that this was required by NIST to facilitate their RS-232 loop back test. We decided it was not appropriate for this profile to require a particular hardware interface.

We had some discussion on the Fortran Language option when we realized that as the document stood we referenced FORTRAN 77, specified Fortran 90 and required a binding to FORTRAN 77 (POSIX.9). Although we are not sure what to do for the final draft, we are sure that it should be consistent. The issues include:

1. POSIX.9 currently refers to an ANSI standard (FORTRAN 77) and is not an international standard.
2. The International standardized version of FORTRAN is Fortran 90, which is not as widely used as FORTRAN 77.
3. This profile is intended to be forwarded to ISO.

The options that we see ahead of us include:

1. Drop the Fortran Language option (not desirable).
2. Have two Fortran Language options (though only the Fortran 90 option would probably make it through ISO).
3. Wait for a resolution between POSIX.9 and Fortran 90, then do what seems appropriate.

Finally, we actually removed all references to test methods from the document. The SEC has rescinded the requirement for test methods and we had been spending too much time on it without having satisfactory results. This also saves 5 full sheets of paper (10 pages).

Once the new editing job has been done, we will probably be basically ready to go to ballot, but we will have to wait because our document depends on both POSIX.4a and POSIX.1a.

**Report on POSIX.19: Fortran-90 Bindings**

*Michael J. Hannah <mjhannah@sandia.gov> chair of POSIX.19, reports on the April 19 – 23, 1993 meeting in Irvine, CA:*

This was an important meeting for anyone following the subject of FORTRAN language bindings to POSIX. After two years of effort to drum up interest in a Fortran 90 binding, the POSIX.9 Working Group proposes to call it quits. The few folks who are left believe that there is an insufficient body of knowledge, practice, or users of ISO/IEC 1539:1991 (Fortran 90) to sustain the effort of producing a POSIX binding at this time, especially in light of a number of outstanding technical issues. Many of these issues concern trying to determine how best to use the new features of the Fortran 90 language in a POSIX binding. However, in a spirit of one last time, the Working Group postponed until July the final act that would disband the Working Group. At the July POSIX meeting, the Working Group will meet for one day only, Monday, July 12. Barrering a large turnout desiring to retain the Working Group, the Working Group will recommend that the executive committee of POSIX withdraw sponsorship of the Fortran 90 binding project and disband the group.
The group is circulating a draft of a final report among members who have participated in the effort so far, and will present the completed final report at the July meeting.

The probable demise of this working group raises several questions about POSIX and language bindings:

1. How many different languages are likely to bind to POSIX? If this is only a few, does this imply that POSIX is less valuable?
2. Is POSIX just for C and Ada, and all other languages should simply figure out how to call system routines in those languages? Does this make those other languages second class citizens in a POSIX world?
3. If there are to be future language bindings, should the IEEE with its POSIX steering committee sponsor that work, or should the committees that define and standardize the language (usually part of ANSI X3) define those bindings? There is some discussion of Modula 2's and COBOL's doing this, and the Fortran 90 project might be resurrected in X3J3 rather than IEEE POSIX. Is this good or bad?
4. Is the lack of interest in Fortran 90 simply a timing issue due to the lack of widespread access to Fortran 90 compilers, or is it due to a lack of interest in Fortran 90 itself? or is this just another victim of the economy?

There are undoubtedly a wide variety of reasons why there is insufficient support at this time for this work. There could be considerable debate over which reason was the most significant. Some would argue that the group should never have tried to start. However, it is clear that there is inadequate support to continue. I believe it is the responsible thing to disband this working group. If you don’t agree, now is the time to speak up.

---

**Automatic Information Server**

The Association now has a mail server that will answer requests received through electronic mail. Information is available about USENIX membership, conferences, and publications.

Please send all requests to: <almanac@usenix.org>

To receive the Almanac Primary Services Catalog, the body of your mail message should contain only the line:

    send catalog

This catalog outlines accessible information about the USENIX Association's services. You can get more detailed listings for sub-topics in each service category by sending the line:

    send service catalog

where “service” is one of the primary topics listed below. Thus:

    send conferences catalog

will result in your receiving a listing of available announcements for upcoming USENIX Conferences and Symposia.

All files are in plain text format.

Questions, comments, suggestions: please contact <almanac-admin@usenix.org>.

---

**Service:** conferences
**Description:** Conferences contains all the current announcements for USENIX Conferences and Symposia. This includes all available calls-for-papers and registration information (technical sessions, tutorials, fees, and accommodations) for upcoming events. To get conferences catalog:

    send conferences catalog

**Service:** publications
**Description:** Publications contains all ordering information for our conference proceedings, back issues of our quarterly technical journal, Computing Systems, as well as publications announcements, and calls-for-papers for special issues of Computing Systems. To get publications catalog:

    send publications catalog

**Service:** membership
**Description:** Membership contains information about the USENIX Association, the System Administrators’ Guild (SAGE), and an application form. To get membership catalog:

    send membership catalog
Back to Basics

The last few weeks have seen a flood of "beginning" books hit my mailbox. Though most readers of this column won't need them, I'm sure that many of you get asked for pointers (or are looking for possible textbooks).

Among these books are three one-volume guides to UNIX: Modern UNIX, UNIX for Programmers and Users, and A Student's Guide to UNIX. While the first and last of these are useful, I found Graham Glass' "complete guide," less so. (The reason for this may be the publisher's fault, not Glass': Bach and Leffler, McKusick, Karels & Quarterman each run to 400-500 pages. The notion that we can have a "complete guide" in just over 600 is a bit silly. Glass' early chapters, on UNIX philosophy and UNIX for nonprogrammers, are quite well-done. The reference list (of only 19 items) is just weird in form and content.)

Southerton's Modern UNIX is less pretentious, but still leaves me with an unhappy feeling. I think this was because Southerton, who is Product Reviews Editor for UNIX World, is trying to do too much in a small compass. The blurb tells me that this volume is for "people who need to become 'power users' quickly." It ain't.

Lest all of you think that I'm just a grouch this month, let me set up as Exhibit A Hahn's Student's Guide to UNIX. At just over 600 pages, this is nearly the same size as Glass' book, but it is a model of presentation and limitation. For example, .forward files are listed in the chapter on dot files in general and considered part of mail. Neither Southerton nor Glass think forwarding is useful enough to talk about. Hahn's chapters on shells are quite good. He also appears to be the only one of the three authors who has heard of bash and Zsh. Without being overly detailed, let me also compliment Hahn for having the good sense to include a section on games. While we all know that (a) the person in the next office wastes his/her time playing games, we also know (b) that we have never played games. This is the perfect book for newcomers to UNIX: whether they are raw newcomers or transferees from some other OS. After you've zipped through ORA's wonderful "Single Session Overview," Hahn's is the book to turn to. And a compliment to McGraw Hill: the book is well laid-out and the inside rear cover features a "Quick Index" to UNIX Commands.

Yet more Internet books

Prentice Hall has taken over the publication of the "Internet Information Series" from SRI. The volumes I've got are Internet: Getting Started and Internet: Mailing Lists. Getting Started is a useful volume that folks intending to use the Clinton-Gore information highway should acquire. I see a period in the near future when folks who know next to zilch about computing/routing/sendmail/filenames/uucp will be sending notes to their friends and family and trying to glean information from remote sources. As more and more folks who have trouble setting the time on their VCRs get access, books like this become more important. Mailing Lists contains a list of lists of over 800 mailing lists. I don't know how many such lists there are. I do know that only two of the five noncomputing lists I'm on appear (medtext, gerling, and disc-nordic are missing). I hope the score will improve in the future: the editors do request additions and corrections.

Insecurity

In August 1988, Matt Bishop chaired the first USENIX Security Workshop in Portland, OR. He also chaired the second one there in 1990. The third Security Workshop was held in Baltimore last September, cohosted by CERT, and chaired by Ed DeHart. If Ferbrache and Shearer had been aware of these, they might have mentioned Dan Klein's work on password-cracking (since he also gave a paper at the UKUUG in 1990, they might have noticed that).

I am going to try and control myself where Ferbrache and Shearer's sloppy yet expensive book is concerned. I hesitate to call them plagiarists, as some items might be inadvertent: like printing nearly all of Gene Spafford's Purdue report on the Internet worm, without a note at beginning nor end that this isn't their work (Appendix A); or "paraphrasing" much of Farmer & Spafford's paper on COPS from the Anaheim (1990) USENIX Proceedings.

But let's ignore the stuff that ain't theirs: it's good stuff, anyway. But what about ignoring a vast amount of other material: Bill Cheswick's, or some of the Kerberos things other than Steiner et al., or de Alvare, or some of the European work.
Moving right along, what I meant as sloppiness
may be exemplified by on the one hand, the
chapter on CERT without an address in the list of
"Contact Addresses;" and, on the other, by the
fantastic contempt that the authors seem to have
for peoples' names (Merrit for Merritt; Mockpet-
ris for Mockapetris [several times]; etc. Enough,

enough.

Not recommended. If you need a security book,
buy Curry or Garfinkel & Spafford.

Take Vitamin C

To get rid of the bad feeling that Ferbrache and
Shearer give you, get a copy of the second edition
of Tizard's book on C. I liked the first edition of
this back in 1987. But that was pre-ANSI. Tiz-
izard has gone ahead and really rewritten his
book in a useful and practical way. This one is rec-

ommended. But it's not for beginners.

TLI and Sockets

Mike Padovano has churned out a massive book
that's decidedly not for beginners. It is, however,
the full answer to a question in Vol. I of Comer's
Internetworking ... series: "What are the major con-
ceptual differences [between TLI and sockets]?"
Half of Comer's answer is in Vol. III [the BSD/
sockets part]; most of Padovano's 500+ pages are
devoted to TLI. I have to admit that he's also got
a respectable amount of BSD material, but it
appears to be in there only to show how much
better (and better-integrated) SVR4 TLI is. (As
SVR4 incorporated sockets into the AT&T code,
one can understand this.)

Don't get me wrong: this is a very good book. I
think that it will end up on most shelves next to
the Comer [& Stevens] trilogy (soon to become a
tetralogy as C&S keep up with Marshall Rose in
volume production) and Rich Stevens' networking
book. But as I said, this is not for the uninini-
tiate.

Padovano has a good discussion of TCP/IP
inside SVR4, including the implementations of
ftp and telnet (Chapter 3). He talks about the
mapping of the socket routines into TLI routines,
referring to his chapter 10 for full details. And
then he gets into ports. What would the tyro
make of:

Every system offering the ftp service has it
available on port number 21. The TCP/IP
specification states that port numbers over
255 are not reserved. However, SVR4
restricts these port numbers by using the fol-
lowing BSD conventions: Port numbers 1
through 1024 are considered "privileged

ports." Your application can use these ports
only if your effective user-id is 0. (p. 65)

There's a lot more. Flipping to chapter 10,
"Applications using Sockets" (pp. 449-536) is,
in fact, helpful. Padovano clearly knows his
stuff. The only problem with this book, how-
ever, is that not only are there a lot of applica-
tions using sockets around, there are a lot of
folks who prefer BSD to USL (and I am not
referring to legal strife). Relegating BSD mate-
rial to a few scattered remarks and the final
chapter (though a long one) does a disservice.

Padovano's prejudices show through.

On the other hand, Padovano's sections on the
architectures and goals of RFS and NFS are
really very fine (Chapter 5). The two brief
expositions on goals (pp. 133-134 and 147) are
the answers to all those interminable discus-
sions of which is better, a skillet or a pencil.
Check it out.

The X Window System

Issue five of The X Resource contains the Pro-
ceedings of the 7th Annual X Technical Confer-
ence (Boston, January 1993). I'm not going to
itemize papers, but if you work with/program
in X and don't get this quarterly, you ought to.
I hope that with Bob Scheifler stepping down
from the X Consortium, the Conference and the
Resource don't vanish, too.

---

Graham Glass, UNIX for Programmers and Users.

Alan Southerton, Modern UNIX. John Wiley &

Harley Hahn, A Student's Guide to UNIX. McGraw

A. Marine, S. Kirkpatrick, V. Neou, & Carol Ward,
Internet: Getting Started. SRI International/Prentice

E. Hardie & V. Neou, Internet: Mailing Lists.

David Ferbrache & Gavin Shearer, UNIX Installa-

Keith Tizzard, C for Professional Programmers. 2nd

Michael Padovano, Networking Applications on
544pp. $44.

The X Resource. O'Reilly & Associates. ISSN 1058-
5591. $65/year [in USA].

---

May/June 1993
You say this doesn’t sound like a UNIX book? Especially since the programs are written in a revolting language that’s basically a relic of the past? So what on earth is a review of it doing in ;login:, you say?

True, this is not a UNIX book, and it’s on a subject – numerical computing – that most UNIX systems programmers would walk miles to avoid. However, now and then you don’t get a choice. This is one of those rare books that is both an authoritative overview of a field and a down-to-earth practical reference; it is \textit{the} book you want to have handy just in case you do need to know how to do some numerical computing.

Heaven knows, I’m not much into the numerical stuff myself, but when I suddenly needed to write a numerical program, I happened to have the previous edition of this book on my shelf, and it told me everything I needed to know. I originally thought I had a reasonable idea how to tackle the problem, maybe not great but okay. When I read the relevant chapter, one of the first things in it was a discussion of why my intended algorithm was a bad choice. The final program was far superior to anything I could have come up with on my own. When the new edition of the book came out, I considered it a smart investment, and it’s already paid for itself.

If you came through a computer science background, like me, you probably have less-than-fond memories of the numerical-analysis course that CS departments like to inflict on their victims. Let me assure you that this book is nothing like that course or the textbook you used. There is a little bit of theory here and there, because the book spends just as much (if not more) paper on explaining the algorithms as on presenting the code for them. However, the emphasis is on practical advice: what works and what doesn’t, what precautions to take, how the various methods compare on accuracy and speed. The book is full of references for further reading, but its own discussion concentrates firmly on the most useful methods for people who have to get some actual computing done.

While \textit{Numerical Recipes} does cover some of the same ground as a traditional numerical-analysis course, it also ranges further afield. It talks about systems of linear equations, numerical integration, root finding, integration of differential equations. It also covers evaluating various unusual functions, generating random numbers (including a simple routine to “clean up” the output of the rather poor random-number generators found in typical subroutine libraries), Monte Carlo methods, sorting, statistics and modeling of data, the Fast Fourier Transform, etc. There’s even a routine to compute the phase of the Moon, in case you really want your software to depend on it!

Overall, the discussion ranges from quite elementary to moderately advanced. The book can’t introduce you to \textit{everything}; in some areas, you need a general idea of what’s going on before the book’s discussion makes sense. In general, though, it does a commendable job of filling in the background of the topics it discusses, which is important if you don’t know very much about a subject and need to decide which method to use. The language throughout is informal and readable; the authors are programmers, not mathematicians, and it shows. There is even occasional humor: “If you know what bubble sort is, wipe it from your mind; if you don’t know, make a point of never finding out!”

The one criticism of this book that I’ve heard occasionally is that its treatment is sometimes too simplistic, and you can get into trouble with it. This is usually accompanied by comments, or at least implications, like “if you don’t know what you’re doing, you have no business trying to do numerical computing.” That’s all well and good, but a lot of us have to try anyway. Expertise in numerical computing is relatively scarce and expensive, either in money (to hire it) or time (to acquire it yourself), and all too often you simply cannot afford it. This book is not as good as ten years of training and experience in numerical computing, but if you don’t happen to have such
expertise on tap, using this book is a whole lot better than digging out your old numerical-analysis textbook or blundering ahead on your own.

In case I haven't made it clear enough already, I highly recommend this book. If you ever need to do any numerical computing on short notice, you will be very happy that you bought it. It's a priceless reference.

At this point, I should explain that this book exists in several different versions. Folks who don't like FORTRAN may prefer this book's brother: Numerical Recipes in C, 2nd ed. The old first edition, still very useful though not as good as the second, also came in Pascal and BASIC versions, and those are still available. There are also example books (showing how to use the routines from the main books), and the code is available on floppies.

So why did I get the FORTRAN version, given that I'm primarily a C programmer and don't much like FORTRAN? Well, the authors of this book are FORTRAN programmers first and foremost, and it shows in their C code. I'd rather read the programs in the original and not have to puzzle over odd idioms created by the translation. FORTRAN is not that hard to follow, especially since the second edition of the book formats the programs for readability rather than worrying about whether FORTRAN compilers would approve. For example, the programs are mostly in lowercase letters rather than the traditional EVERYTHING IN UPPERCASE style that makes FORTRAN so painful.

But isn't it annoying having to rethink the algorithms to express them in C? Well, yes, but this brings up one final topic that I ought to mention. The programs in this book and on those floppies that I mentioned are not public domain or freely redistributable. Buying the book (as opposed to borrowing your friend's copy!) does give you reasonable rights to personal use (as explained in a "Legal Matters" preface), but more serious use needs to be licensed. If you don't want to hassle with that, you can't use their code (or a direct translation). But the real value of this book is in the algorithms, not the code, and it's only the code that's copyrighted.

I haven't bothered buying the floppies. The book is enough. I can handle the programming side of it, if I know what I should be programming. When it's something numerical, Numerical Recipes is where I go to find out.

Inside Windows NT


Reviewed by Bob Birss
SunPro
<bob.birss@sun.com>

Lots of people want to read this book. Lots of people have already bought it. It is, after all, "the inside story behind the design, philosophy, architecture, and future of Microsoft's next generation operating system," as the blurb on the inside of the two-part front cover tells us. But will those people find it satisfactory? I doubt it.

Inside Windows NT tries to be almost all things to almost all people - it is "not written for operating system designers...[but] for the rest of us, those who know something about computers and who want to understand the internal design of this system." A tall order, and one which it doesn't quite fulfill.

The book invites comparison to Tracy Kidder's Soul of a New Machine, and, in fact, Custer says she reread that book "for inspiration and a sense of kinship." And, she says, "the construction of Windows NT was a software version of the hardware construction documented in Kidder's book." I'm sure that's true. But this book isn't the story of the development of NT, except for occasional mentions that "this person designed that" or "that person, who spent years at DEC, was the architect of this." These glimpses are annoying, because they're not enough for the person who really wants to know the story of NT development and too much for the person who simply wants technical details.

The book also invites comparison to Maurice J. Bach's The Design of the UNIX Operating System. Both books contain a wealth of technical detail. And Bach is addressing a fairly wide audience: students, systems programmers, and "programmers on UNIX systems" (p. xii). But Bach's book is more satisfying, perhaps because it's about a
system that had actually been shipped.

That NT hasn’t shipped yet is probably one of the biggest reasons why this book is unsatisfying. It is certainly hard to be accurate about the precise details of a system months before it ships. And it is certainly a good marketing ploy to have released this book ahead of shipment. But if you want to find out what’s really going on inside Windows NT, it won’t be easy to do so from this book.

Take, for example, the point made by David Fiedler in his article “Microsoft Sets Its Own Standard” in the March 15 Open Systems Today. Fiedler notes that Microsoft has recently been conducting a campaign to convince us that NT is “open” and has an “open architecture.” But, as he says, if we consult Custer’s book, we see that the term “open architecture” refers only to interoperability across networks (p. 304) and that openness per se is not listed as one of NT’s design goals (p. 5), though compatibility “with existing Microsoft systems” is.

That sort of inconsistency might bother you, too, if you’re one of those folks that the back cover tells us is “making decisions about how to proceed into the future of personal computing.” You might want a detailed comparison to UNIX, for example. You won’t find it. You might want a description of what hardware you’ll need to run NT. You won’t find that, either.

There is certainly a wealth of technical detail in this book, and many excellent illustrations. But it feels like only half the story. Perhaps someone is already at work on “Outside Windows NT.”

UNIX Power Tools


Reviewed by Rob Kolstad
<kolstad@bsd.com>

The first thing one notices about UNIX Power Tools is its size. It is large. It is hefty. It weighs in at about four pounds (even in paperback). It’s almost two inches thick. There’s a lot of it: 32 pages in the table of contents alone; 1,059 pages of terrific text, an 11 page glossary, and a 47 page index!

Bigger is not always necessarily better, but in this case bigger is fantastic.

If you have ever been the guru at a site or the “hot shot” system administrator then you’ve answered dozens of user questions like “How do I move lines from here to the middle using vi?”, “How do regular expressions work?”, “Why does my shell start so slowly?”, and dozens more – the answers to which are learned only in the school of hard knocks (along with the subtle complexities behind the first-order answers). This book answers those questions and hundreds more.

Want a quick intro to source-control? See pages 366-375 (both SCCS and RCS). Want to be an expert on the find command? See page 302 and 303. Want to see clever shell tricks you can use today? – check out Chapter 11. The list, while not endless, is nearly so.

This book has all the “tricks” in it that one accumulates after years of trial-and-error in learning how to get real leverage out of using UNIX-style systems. If you’re a user who is not intimidated by UNIX’s building block philosophy, this is the book for you. It includes tricks and explains their derivations. It includes warnings on how the tools will abuse you if misused (warnings in boxes with a lovely rendering of a screw). It even includes a CDROM with all the scripts from the book! Who could ask for more?

I liked the way the authors gave not only their own points of view, but also quoted those who dissented (whether from personal correspondence or from the net). I particularly enjoyed their typographical conventions: the margins include cross-references and CDROM-references; they use blue type inside the text for cross-references. This book is clearly lovingly crafted and must represent years of gathering tricks, tips, and hints.

I’ve asked a dozen people how they liked the book – the answer was universally positive. I love the book. I learned a couple tricks on my first browse through it. If you use UNIX frequently, you can increase your leverage the first time you read one of the book’s 54 chapters. I keep it in my reading room for those times when one wishes to check out a quick 2-3 minute section of a book or magazine.
If you’re a UNIX guru, you can probably learn new things too. And if you’d like to be a UNIX guru, this is the book for you. If you’re intimidated by command line and the “UNIX Philosophy,” this book will seem esoteric; skip it in favor of something more your flavor.

Obtaining GNU Software

by <gnu@prep.ai.mit.edu>

The GNU (GNU’s Not UNIX) Project of the Free Software Foundation (FSF) is developing a complete UNIX compatible software system with freely redistributable source code. The rationale for GNU is explained in the GNU Manifesto. Copies are available in the GNU Emacs Manual and sources, or ask <gnu@prep.ai.mit.edu>.

You can get GNU software from (or with) others, by ftp on the Internet, and by uucp from uunet and other sites. You can also order the software from the FSF. Ask gnu@prep for details.

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

FSF also gratefully accepts donations of any size; as FSF is tax exempt (in the US), your donations are tax-deductible. But usually the easiest way to support the FSF is by ordering a software distribution.

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

Since this article was last published in the September/October issue of ;login: there have been several additions and changes to the available GNU software and the media FSF distributes it on. The major ones are detailed in the following paragraphs. Almost all of the other older software on FSF distribution media has been updated to newer versions that are more bug free and support more UNIX systems.

• Several new manuals have been published: Using and Porting GNU CC, which explains how to run, install, and port the GNU C compiler (editions are available for both GCC 1 and GCC 2); Flex, a lexical scanner generator; GNU Calc, an extensible, advanced desk calculator and mathematical tool that runs under GNU Emacs.

• Two new quick reference cards have been published: Bison, a parser generator; Flex, a lexical scanner generator.

• FSF has produced its first CD-ROM, which contains sources to the GNU Project distribution and other free software. See pages 66 and 67 in the last issue of ;login: for full details and an order form.

• FSF is now offering its software on 8mm Exabyte cassettes.

• FSF’s Deluxe Distribution Package includes executables and source for all of GNU software in a choice of formats, as well as a printed copy of each of our manuals.

• The new quarterly Subscription Service provides four new versions of the tape of your choice for the price of three. It is offered only for tapes that change frequently.

• FSF is now distributing diskettes with some of the software that has been ported to MS-DOS: Demacs, a port of GNU Emacs, in two versions (one which handles 8-bit character sets, and one, based on Nemacs, which handles 16-bit character sets, including Kanji); DJGPP, a complete port of GCC, libraries, development utilities, and a symbolic debugger, for Intel 80386 and 80486; Selected Utilities which include: Bison, RCS, flex, GAWK, cpi, diff, MicroEmacs, find, some file utilities, gdbm, grep, libc, ptx, indent, less, m4, make, sed, shar, sort, and Texinfo; and Programs for Windows, GNU Chess and gnuplot.

• The gzip utility is the GNU replacement for compress and is on all media that contain gziped files. It is currently in beta release. gzip compresses much more than compress does; a file compressed with gzip is usually two thirds the size of a file compressed with compress. Additionally, although gzip is slower than compress, gunzip is faster than uncompress.
New software on the Emacs Tape includes:
* CUSP, a Common Lisp interpreter and compiler;
* PCL, a large subset of CLOS, the Common Lisp Object System.

New Software on the Languages Tape includes:
* GDB 4, the GNU source-level C and C++ debugger. GDB 4 is out of beta test and has moved from the Experimental Tape.

New Software on the Utilities Tape includes:
* autoconf, produces shell scripts which automatically configure source code packages;
* fax, Group 3 fax transmission and reception services for a networked UNIX system;
* mtools, programs to allow UNIX systems to read, write, and manipulate files on an MS-DOS file system;
* recode, converts files between character sets and usages;
* wdiff, compares two files, finding which words have been deleted or added to change one into the other;
* screen, allows several independent screens (ttys) on a single physical terminal;
* termcap, an improved drop-in replacement for libtermcap.a.

New software on the Experimental Tape includes:
* Binutils 2.0, this release uses BFD (Binary File Descriptor) library;
* Oleo, a spreadsheet that is better for you than the more expensive spreads. A runtime system for the Objective C language is now available. As of version 2.3, GCC can run Objective C programs on any of the supported target machines.

And finally, FSF is now offering GNU T-shirts. They are 100% cotton and are available in two colors with a picture of an intensely hacking gnu on the front.

Many of the programs in the GNU Software distribution are covered by either the GNU General Public License or the GNU General Library Public License. These Licenses permit everyone to have and run copies of the programs, at no charge, and to redistribute copies under certain conditions designed to make sure that all modified versions remain as free as the versions GNU distributes. The Licenses are usually in files named COPYING and COPYING.LIB. They can also be obtained from <gnu@prep>.

[The USENIX Association is printing this information as a service to the user community; no endorsement of GNU software is implied.]
Mobile & Location-Independent Computing

Symposium: Preliminary Program
Cambridge, Massachusetts
August 2–3, 1993

Like transistors replacing tubes, OSs being written in HLLs, high-speed local networking, or distributed computing, achieving the mobile or location-independent computer will profoundly change the manner in which we compute. The introduction of the light-weight mobile computer with plenty of main and virtual memory, megapixel display, backing store, and communications hardware is a seminal advancement in technology.

It is this conclusion, reached by a number of folks independently and collectively within the USENIX community, which led to the organization of the USENIX Mobile and Location-Independent Computing Symposium. The purpose of this Symposium is to explore the questions and challenges that we, as developers and researchers, must answer at the dawn of mobile computing. We encourage your participation in examining the implications of this soon to be pervasive technology.

How will mobile computing actually be implemented? What, for instance, happens to filesystems? How do we achieve “automagic” reconciliation of changed data files, possibly of changed system configurations, after a mobile system has been operating (as the buzzword goes) “detached” and is reattached to a network?

What do we need to change and what is essential in the structures and operations of our systems? For instance, is Kerberos “good enough” or do we need new, different security models? What must happen to UNIX to meet the new way of computing?

At this Symposium we hope to accomplish a first discussion of the issues. Possible approaches and actual experiences will be shared. Two keynote addresses from knowledgeable watchers of new technology, refereed paper presentations, panel discussions, Work-in-Progress Reports, technology demonstrations, and Birds-of-a-Feather Sessions will facilitate this discussion. A panel discussing what we have learned and where we need to go will conclude the Symposium.

Monday, August 2
Keynote Address by Bob Metcalfe, InfoWorld

Dr. Bob Metcalfe is currently Publisher/CEO of the InfoWorld Publishing Company. He graduated from MIT, received his Ph.D. in CS from Harvard in 1973, and was a professor at Stanford University. In 1988 he received the Alexander Graham Bell Medal for his contributions to the invention, standardization and commercialization of local-area networks. He also invented Ethernet at Xerox Palo Alto Research Center in 1973 and founded 3Com Corp. in 1979.

Connectivity

The Qualcomm CDMA Digital Cellular System
Phil Karn, Qualcomm, Inc.

An Infrared Network for Mobile Computers
Norman Adams, Bill N. Schilit, Rich Gold, Michael Tso, Roy Want, Xerox PARC

UNIX For Nomads: Making UNIX Support Mobile Computing
Michael Bender, Alexander Davidson, Clark Dong, Steven Drach, Anthony Glenning, Karl Jacobs, Jack Jia, James Kempf, Nachiappan Periakaruppan, Gale Snow, Rebecca Wong, Nomadic Systems Group, Sun Microsystems

Protocols And Messaging

Providing Connection-Oriented Network Services to Mobile Hosts
Kimberly Keeton, Bruce Mah, Srinivasan Seshan, Randy Katz, Domenico Ferrari, University of California - Berkeley

A Mobile Networking System Based on Internet Protocol (IP)
Pravin Bhagwat, University of Maryland; Charles Perkins, IBM T.J. Watson Research Center

Agent-Mediated Message Passing for Constrained Environments, Andrew Athan, Daniel Duchamp, Columbia University

Panel Discussion

Active Badges, Civil Liberties, Security, and Privacy
Panelists TBA
Technology Demos

A special feature of the Symposium will be demonstrations of some of the latest products and still-in-development technology. Six to eight demonstrations will be provided by vendors and by researchers, who will then be available to answer your questions.

If you are interested in demonstrating relevant technology, please contact Cynthia Deno, USENIX Consultant, (408) 335-9445 or email <cynthia@usenix.org>.

Tuesday, August 3

Work-in-Progress Reports – TBA

Filesystems And Naming

Disconnected Operation for AFS
Larry B. Huston, Peter Honeyman, University of Michigan

File Access in Mobile Computing Environments
M. Satyanarayanan et. al., Carnegie Mellon University

Using Prospero to Support Integrated Location Independent Computing
B. Clifford Neuman, Steven Seger Augart, Shantaprasad Upasani, University of Southern California

Hosted Luncheon With Keynote Speaker - TBA

Experience

BNU: Experiences from an Operating Systems Course Project in Mobile Ubiquitous Computing
Terri Watson, Brian Bershad, Carnegie Mellon University

Experiences with X in a Mobile Environment
Christopher A. Kantarjieff, Xerox PARC

Customizing Mobile Applications
Bill N. Schilit, Marvin M. Theimer, Brent B. Welch, Xerox PARC

Panel Discussion

Reprise: Do we need to start over, or are we on the right track?
Panelists TBA

Program Committee

Dan Geer, Program Chair
Geer Zolot Associates

Clement Cole, Vice-Program Chair
Locus Computing Corporation

Ed Gould, Digital Equipment Corporation
Mike Kazar, Transarc Corporation

Jeff Kellern, Beyond Dreams
Alan Nemeth, Digital Equipment Corporation

Tom Page, UCLA
Charlie Perkins, IBM T.J. Watson Research Center

Dave Presotto, AT&T Bell Laboratories
Jim Rees, University of Michigan

A limited number of student scholarships are available for full-time students. For further Symposium, registration and scholarship information, please contact the USENIX Conference Office.
FOR COMPLETE CONFERENCE INFORMATION AND TO REGISTER:

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

CONFERENCE SCHEDULE

- **Pre-Registration Savings Deadline**
  Tuesday, June 1, 1993 PRE-REGISTER TO SAVE $100

- **Hotel Discount Reservation Schedule**
  Friday, May 28, 1993

- **Vendor Displays**
  Wed. & Thurs., June 23-24, Noon-6 pm

- **Birds-of-a-Feather Sessions**
  Tues.-Thurs. evenings, June 22-24

- **USENIX Conference Reception**
  Thurs., June 24, 6-8 pm

- **On-Site Conference Registration**
  June 20, 4-9 pm, June 21-24, 7:30 am-6 pm

TUTORIAL PROGRAM

**Monday and Tuesday, June 21-22, 1993**

The USENIX Association's well-respected tutorial program offers you intensive and practical tutorials in essential areas of UNIX-related technology. Tutorials are taught by experienced instructors, experts in their field.

- How Networks Work NEW!
- OSF's Distributed Computing Environment (DCE)
- The Kerberos Approach to Network Security
- Essential UNIX Programming
- Process & Virtual Memory Systems and MP Support NEW!
- Threads, POSIX PThreads & OSF/DCE Threads UPDATED!
- UNIX Power Tools: Getting the Most out of UNIX NEW!
- Security & The X Window System NEW!
- Topics in System Admin – Part 1 NEW!
- Managing the Domain Name System
- Topics in System Admin – Part 2 NEW!
- Symmetric Multiprocessing and Caching
- DCE Remote Procedure Call System NEW!
- Sendmail: Inside and Out NEW!
- UNIX Network Programming
- SRV4.2 Internals: File Systems, I/O & STREAMS NEW!
- Windows NT Architecture NEW!
- Achieving Security in an Internet Environment NEW!
- Tcl & Tk: A New Approach to X11 & GUI Programming
- Installing, Configuring & Administering X Systems NEW!

**Wednesday, Thursday, and Friday, June 23-25, 1993**

Parallel tracks of presentations of refereed papers, published in Conference Proceedings, and talks by invited experts.

KEYNOTE:

Evolving New User Interface Technologies for UNIX
Bruce Tognazzini, SunSoft, Inc.

- Call Path Profiling of Monotonic Program Resources in UNIX
- Computer System Performance Problem Detection Using Time Series Models
- Design and Implementation of a Simulation Library Using Lightweight Processes
- Invited Talk: Five Years of Gateways and Hackers
- The Restore-o-Mounter: The File Motel Revisited
- The Autosom Automounter
- Discovery and Hot Replacement of Replicated Read-Only File Systems
- Invited Talk: That's Easy with my Editor
- X Through the Firewall, and Other Application Relays
- The Ferret Document Browser
- LADDIS: The Next Generation in NFS File Server Benchmarking
- Design and Implementation of a Multimedia Protocol Suite in a BSD UNIX Kernel
- Invited Talk: Introduction to Object-Oriented Programming and C++
- Invited Talk: Ten Problems in UNIX, and How Object Technology Solves Them
- The Spring Nucleus: A Microkernel for Objects
- “Stacking” Vnodes: A Progress Report
- Anonymous RPC: Low-Latency Protection in a 64-Bit Address Space
- Invited Talk: Digital Signal Processing 101: Sound Programming for your Workstation
- Integrating Handwriting Recognition into UNIX
- Optimizing UNIX Resource Scheduling for User Interaction
- AudioFile: A Network-Transparent Audio Server
- Invited Talk: Highlights from the 1992 USENIX UNIX Security Symposium
- Work-In-Progress Reports
- Invited Talk: Highlights from the 1993 USENIX Mach Symposium
- Fast and Flexible Shared Libraries
- High Performance Dynamic Linking Through Caching
- The Shell as a Service
- Invited Talk: Analysing Backup Systems
- A User-Level Replicated File System
- sfs: A Parallel File System for the CM-5
- Adaptive Block Rearrangement Under UNIX
- Invited Talk: UNIX Documentation: Where are We and How Did We Get Here?
- Panel on Privacy
A Special Invitation to Full-time Students

The USENIX Association is making a special invitation to full-time students the USENIX Summer 1993 Technical Conference from June 21-25, 1993 at the Cincinnati Convention Center in Cincinnati, Ohio.

ATTEND THE KEYNOTE FOR FREE
Wednesday, June 23, 9:00 am

Evolving New User Interface Technologies for UNIX, Bruce Tognaazzini, SunSoft, Inc.

Our keynote speaker, Bruce Tognaazzini, has long been a customer of the operating system support for the user interface. He has been designing human interfaces for better than 30 years. He spent the last 14 years at Apple where he led at various times both the Apple II and Macintosh human interface efforts before moving to SunSoft last year. During his most recent tenure in the Evangelism group at Apple, he wrote a major new publication in the field of human-computer interaction, "Tog on Interface."

ATTEND THE VENDOR DISPLAYS FOR FREE
Wednesday and Thursday, June 23-24
Noon - 6:00 pm

At the USENIX Vendor Display in Cincinnati, vendors will demonstrate advanced computing and networking products and services. Here, in a relaxed environment, you can get real answers from technically-savvy vendor representatives. Go ahead, kick the tires, and be sure what's on display really does what it's said to do.

ATTEND A TUTORIAL FOR $50
Monday and Tuesday, June 21-22
9:00 am-5:00 pm

A limited number of spaces in each tutorial are reserved for full-time students at the special fee of $50.00 per tutorial. Telephone the USENIX Conference Office, +1 714/588-8649 during office hours of 8:30 am-5:00 pm, Pacific Time, Monday–Friday to confirm availability of the tutorial of your choice and to make a reservation. All tutorials offer printed materials. Tutorial fee includes a box lunch.

ATTEND THE TECHNICAL SESSIONS FOR $75
Wednesday through Friday, June 23–25
Sessions begin at 9:00 am

A program of parallel tracks of presentations of refereed papers and tutorial-style talks by invited experts is rounded out by Work-in-Progress Reports (students: schedule your WIP report by sending email to <WIP@usenix.org>), a panel discussion on privacy, and highlights from recent USENIX single-topic symposia. The conference reception is included in the fee. Also included is a copy of the Conference Proceedings and a copy of the Invited Talks Submitted Notes.

USENIX SUPPORTS STUDENT PARTICIPATION AND PROFESSIONAL DEVELOPMENT

• The USENIX Educational Outreach Program provides full-time students access to full conference scholarships and USENIX publications.

• A $500 cash prize is awarded for the Best Student Paper at USENIX annual Winter and Summer Conferences. (Students are eligible also for Best Paper and other awards.)

Membership in USENIX for full-time Students is only $20. As a member, your benefits include:

• Free subscription to ;login:, bimonthly newsletter with articles, community news, calendar of events, book reviews, Standards Activities Reports, Systems Administrators' Guild News, and more

• Free subscription to Computing Systems, refereed technical quarterly published with the University of California Press

• $75 technical sessions fees vs fees of $275 to $340 for as many as ten technical conferences and symposia each year and $50 tutorials at USENIX annual Winter, Summer and Systems Administration Conferences

• Discounts on conference/symposia proceedings, other technical publications, and books from the new series on advanced computing published with MIT Press

• Eligibility to join SAGE, the Systems Administrators' Guild

And maybe most important, participation in leadership in the UNIX community.
IMPORTANT DATES

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

Registration Materials Available:
- July, 1993

ANNOUNCEMENT/CALL FOR PAPERS

Sponsored by the USENIX Association
- Co-Sponsored with The Computer Emergency Response Team (CERT)
- In cooperation with The ACM SIGSAC

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)
**Program Committee**

- **Program Chair:**
  - Bill Cheswick, AT&T Bell Laboratories
  - Steve Bellovin, AT&T Bell Laboratories
  - Matt Bishop, Dartmouth College
  - Ed DeHart, CERT, Carnegie Mellon University
  - Jim Ellis, CERT, Carnegie Mellon University
  - Marcus Ranum, Trusted Information Systems

**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

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 Chain

Bjorn Satdeva, /sys/admin, Inc.
Brent Chapman, Great Circle Associates
Lee Damon, Castle PAUS
Tina M. Darmohray, Lawrence Livermore National Labs
Janet Frazer, UNIX System Laboratories, Inc.
Helen Harrison, SAS Institute
Dinah McNutt, Tivoli Systems
Bryan McDonald, SRI International
Arch Mott, Cisco Systems, Inc.
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.
HOW TO SUBMIT A REFEREED PAPER

As at all USENIX conferences, papers that analyze problem areas and draw important conclusions from practical experience are welcome. Note that the USENIX conference, like most conferences and journals, considers it unethical to submit the same paper simultaneously to more than one conference or publication or to submit a paper that has been or will be published elsewhere.

Authors of papers to be presented during the conference technical sessions and published in the Proceedings must submit an extended abstract to the Program Committee by July 13, 1993. The object of an extended abstract is to convince the reviewers that a good paper and 25-minute presentation will result. They need to know that authors:

- are attacking a significant problem;
- are familiar with the current literature about the problem;
- have devised an original solution;
- have implemented it and, if appropriate, characterized its performance;
- have drawn appropriate conclusions about what they have learned and why it is important.

An extended abstract should be about 5 pages in length, or about 2500 words. It should represent the paper in “short form.” Please include the abstract as it will appear in the final paper. The body of the extended abstract should be complete paragraphs, not just an outline of the paper. (Sections present in the full paper but omitted from the abstract may be summarized in terse form; this will help reviewers to understand what material will be present in the final paper.) Authors should include full references, and, if appropriate, performance data to establish that they have a working implementation and measurement tools. Figures should be included when available.

Authors may, at their option, submit a full paper in addition to the extended abstract. Since the schedule for reviewing submissions is short, however, most judgements will be made based on the extended abstracts.

Every submission should include one additional page containing:

- The name, surface mail address, daytime and evening telephone numbers, e-mail address and (if available) fax number of one of the authors, who will act as the contact to the program committee;
- An indication of which, if any, of the authors are full-time students;
- A list of A/V needs beyond a microphone and an overhead projector.

Authors of accepted submissions will be notified by August 30, 1993. They will promptly receive instructions for preparing camera-ready copy of an 8–12 page final paper, which must be received by November 2, 1993.

WHERE TO SEND SUBMISSIONS

Please submit one copy of an extended abstract via at least two of the following methods:

- (Preferred method) e-mail to: SF94papers@usenix.org
- FAX to: the USENIX Association +1 (510) 548-5738
- Mail to: USENIX Winter 94, USENIX Association, 2560 Ninth St., Suite 215, Berkeley, CA USA 94710

Inquiries about submissions to the USENIX Winter 1994 Conference may be made by e-mail to SF94info@usenix.org or by telephone to +1 (510) 528-8649. Potential authors of technical papers are strongly encouraged to send us electronic mail. This will allow us to notify you of any important changes and you will receive additional information about the submission and reviewing process. You may request a sample extended abstract by telephoning +1 (510) 528-8649, by fax to +1 (510) 548-5738, or by e-mail to sample-abstract@usenix.org
For many years, UNIX and its derivatives have been the only widely available "open" operating systems to support modern software technology. The next few years promise to change that, as many new PC operating systems are reaching the market. These systems will compete with UNIX, but they will also broaden the set of systems that can support advanced applications, high-performance computing, novel user interfaces, and improved network communication. The question is not "will UNIX survive," but rather how will UNIX and other systems evolve together to improve our computing environments.

As usual at USENIX Conferences, we are interested in papers describing new and interesting developments in open operating systems. Our traditional focus on UNIX remains, but this includes lessons learned from work on UNIX that can be applied more broadly, and lessons from other kinds of systems that can be applied to the continuing evolution of UNIX.

CONFERENCE SCHEDULE IN BRIEF

Unusual for a USENIX Conference, at San Francisco, the 3 days of technical sessions precede the 2-day tutorial program. Other conference events are scheduled accordingly.

TECHNICAL SESSIONS:
MONDAY THROUGH WEDNESDAY JANUARY 17-19, 1994

TUTORIALS:
THURSDAY AND FRIDAY, JANUARY 20-21, 1994

BIRDS-OF-A-FEATHER SESSIONS:
MONDAY THROUGH THURSDAY EVENINGS, JANUARY 17-20, 1994

USENIX RECEPTION:
TUESDAY EVENING, JANUARY 18, 1994

VENDOR DEMONSTRATIONS:
TUESDAY AND WEDNESDAY, JANUARY 18-19, 1994

VENDOR DEMONSTRATIONS

At San Francisco, vendors will offer short talks and demonstrations of a product or service in a theater-style setting. Technical representatives then will answer specific questions from conference attendees. In addition to scheduled demonstrations, there will be a display area especially for publications and software/communications products and services. Vendors, to participate please contact: Cynthia Deno +1 (408) 335-9445, FAX: +1 (408) 335-2163, E-mail: cynthia@usenix.org
TUTORIAL PROGRAM

On Thursday and Friday of the Winter 1994 Conference, you can explore topics essential to successful use and development of UNIX and UNIX-like operating systems, X windows, networking and interoperability, advanced programming languages, and related areas of interest. The USENIX Association’s well-respected program offers you introductory and advanced, intensive yet practical tutorials. Courses are presented by skilled teachers who are hands-on experts in their topic areas.

At San Francisco, USENIX will offer two full days of tutorials covering topics such as:

- Introductory and advanced systems administration
- Distributed computing: DCE, DME, DFS, RPC
- Kernel internals: OSF/1, SVR4, 4.4BSD
- Developing and debugging X-based applications
- GUI technologies and builders

To provide the best possible tutorial slate, USENIX is constantly soliciting proposals for new tutorials. If you are interested in presenting a tutorial, contact the Tutorial Coordinator: Daniel V. Klein +1 (412) 421-2332, E-mail: dvk@usenix.org

INVITED TALKS

The conference technical sessions include one track of 30-minute long presentations of papers selected by the Program Committee. A second track of Invited Talks, compliments the first. These talks by invited experts provide introductory and advanced information about a variety of interesting topics, such as using standard UNIX tools, tackling system administration difficulties, or employing specialized applications. This second track also includes selections from the best presentations offered at 1993 USENIX single-topic Symposia and panel discussions.

The Invited Talks Coordinators welcome suggestions for topics as well as request proposals for particular Talks. In your proposal, state the main focus, include a brief outline, and be sure to emphasize why your topic is of general interest to our community. Please submit to: Brent Welch, Xerox PARC and Bob Gray, US WEST Advanced Technologies via e-mail to ITusenix@usenix.org

BIRDS-OF-A-FEATHER SESSIONS

The always popular Birds-of-a-Feather sessions (BOFs) bring together devotees of many varied disciplines for demonstrations, discussion, announcements, and mingling. BOFs are offered Monday through Thursday evenings. Schedule a BOF on site or in advance by contacting the USENIX Conference Office, +1 (714) 588-8649, E-mail: conference@usenix.org

WORK-IN-PROGRESS REPORTS

WIP Reports, scheduled during the conference technical sessions, are designed to introduce interesting new or ongoing work. At this conference, we are particularly interested in promoting student research. Presentations should be short, pithy and fun! The USENIX audience provides valuable discussion and feedback. Schedule your Work In Progress (WIP) Report by contacting the WIP Coordinator, Peg Schafer, BBN Inc., on-site or in advance via e-mail to wip@usenix.org

CONFERENCE PROGRAM COMMITTEE

PROGRAM CHAIR:
Jeffrey Mogul, Digital Equipment Corporation
Rafael Alonso, Matsushita Information Technology Laboratory
Brian N. Bershad, CMU
Nathaniel S. Borenstein, Bellcore
Frederick S. Glover, Digital Equipment Corporation, UNIX™ Software Group
Judith E. Grass, Corporation for National Research Initiatives
Michael B. Jones, Microsoft Research, Microsoft Corporation
Phil Karn, Qualcomm, Inc.
Samuel J. Leffler, Silicon Graphics, Inc.
D. R. McAuley, University of Cambridge
David Presotto, AT&T Bell Laboratories
Margo Seltzer, Harvard University
Cathy L. Watkins, Intel Corporation, O/S Technology Engineering

CONFERENCE PROGRAM AND REGISTRATION

Materials containing all details of the technical sessions and tutorial program, conference registration, hotel and airline discount and reservation information will be mailed at the end of September 1993.

PLAN NOW TO ATTEND
CALL FOR PARTICIPATION

The USENIX Association is pleased to host its sixth annual C++ Conference in Cambridge, Massachusetts, April 11-14, 1994. Monday and Tuesday will offer tutorials; technical sessions will take place Wednesday and Thursday. A post-conference advanced workshop will be held Friday, April 15.

TUTORIALS
- Monday and Tuesday, April 11-12, 1994
Two full-day tutorials per day will present intermediate and advanced topics of interest to C++ developers. Potential presenters should contact the program chair for information on how to propose a tutorial. The last date for submission of a tutorial proposal is October 15, 1993. Tutorial proposals will be reviewed and accepted on an on-going basis as they are received by the Committee.

All others are invited to contact the chair to suggest tutorials on topics they wish to see covered.

TECHNICAL SESSIONS
- Wednesday and Thursday, April 13-14, 1994
The two days of technical sessions will cover the spectrum of recent research, development, and experience in C++. The technical sessions will begin with a keynote address and feature presentations of refereed papers. Controversial topics may be addressed in an alternative format, such as a panel session.

Time on Wednesday and Thursday evenings is reserved for Birds-of-a-Feather Sessions. (You may schedule a BOF on-site or in advance by email to conference@usenix.org) All papers accepted by the Program Committee will be published in the Conference Proceedings, which will be distributed to conference attendees. After the conference, Proceedings will be available for purchase from the USENIX Association.

Papers are solicited by the Program Committee on all aspects of C++, including:
- Design methods, patterns, and architectures
- Development environments and tools
- Class libraries, frameworks, and generators
- Compilation and run-time support
- Distributed and parallel systems
- Teaching and learning C++
- Databases and persistence
- Integration with other languages, databases, tools, and methods
- Application development

VENDOR DISPLAY
- Wednesday, April 13, 1994 Noon-6:00 pm
Well informed vendor representatives will demonstrate products and services useful to C++ and the wide range of object-oriented technology at the informal table-top display accompanying the USENIX C++ Conference. If your company would like to participate, please contact Cynthia Deno at +1 (408) 335-9445, FAX +1 (408) 335-2163, Internet: cynthia@usenix.org
HOW TO SUBMIT TO THE TECHNICAL SESSIONS

Extended abstracts of no more than ten pages should be submitted to the program chair by November 30, 1993. Electronic submissions (PostScript, troff, or TeX) are preferred. Otherwise, submit eight paper copies. All submissions should indicate the electronic mail address and/or telephone number of a principal contact. Authors will be notified of acceptance by January 15, 1994 and will be provided with guidelines for preparing the final paper. Authors of an accepted paper will deliver a presentation during the conference technical sessions and provide a final paper for publication in the Conference Proceedings. Final versions of papers are limited to twenty pages and are due by February 21, 1994.

HOW TO SUBMIT TO THE ADVANCED WORKSHOP

Friday, April 15, 1994

The one-day post-conference workshop will focus on the development of methods, tools, and services supporting the use of C++ in distributed computing. Attendance is limited, and based on acceptance of a position paper. Acceptance notices to all participants will be issued by March 1, 1994. Potential workshop attendees are invited to submit a position paper of at most 3 pages (ASCII) to the program chair no later than February 1, 1994. Position papers should briefly describe experiences, interests, works-in-progress, and/or ongoing research and development. (More substantive reports of completed work should instead be submitted to the technical sessions.)

A representative subset of authors of position papers will be invited to make informal presentations at the workshop.

WHERE TO SEND SUBMISSIONS AND ADDRESS INQUIRIES

All queries and submissions should be directed to the program chair:
Doug Lea
Dept. of Computer Science
SUNY Oswego
Oswego NY USA 13126
Phone: +1 (315) 341-2688
Internet: dl@g.oswego.edu

FOR PROGRAM AND REGISTRATION INFORMATION

Materials containing full details of the conference program, registration fees and forms, and hotel discount and reservation information will be mailed and posted to the net in February 1994. If you wish to receive registration materials, please contact:
USENIX Conference Office
22672 Lambert Street, Suite 613
Lake Forest, CA USA 92630
Phone: +1 (714) 588-8649; FAX: +1 (714) 588-9706
Internet: conference@usenix.org

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

CO-SPONSORED BY THE USENIX ASSOCIATION AND UNIFORUM CANADA

One of the major uses of UNIX is the support, development, and execution of applications which ultimately serve as tools for end-users. In addition, the current trend of downsizing major applications from monolithic data-center environments to less expensive, distributed workstations and client-server computing environments affords UNIX a serious position in the commercial marketplace. Because UNIX has become a viable commercial alternative, vendors are now porting and developing code for scientific and business applications which in the past have been the province of contributed code. Consequently, more and more computing and information systems professionals are encountering UNIX when developing and maintaining applications.

The purpose of the UNIX Applications Development Symposium is to expose the challenges of building and maintaining applications on UNIX platforms, to discuss solutions and experiences, and to explore existing practice and technique. Computing professionals who have long viewed UNIX as the program development platform of choice, as well as professionals new to the UNIX environment, will learn of helpful tools, novel approaches, and what not to do when developing for or porting an application to the UNIX environment.

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

The UNIX Applications Development Symposium will provide valuable information to designers, programmers, and managers who plan to port existing applications into the UNIX environment or move development and maintenance teams from various proprietary environments to UNIX.

TUTORIAL PROGRAM

The two, day-long tutorials are targeted to programmers and managers interested in developing applications in, and products for, the UNIX environment. Each is led by an experienced instructor who is an expert in his topic. The Monday tutorial by Richard Stevens covers the use of the UNIX environment to develop applications. The tutorial on Tuesday, presented by Rob Kolstad, covers design and implementation issues regarding effective use by an application of the UNIX environment.

INVITED TALKS AND PANEL SESSIONS

As part of the technical sessions, invited talks provide introductory and advanced information about a variety of interesting topics, such as using standard UNIX tools and employing specialized applications. We welcome suggestions for topics, as well as request proposals for particular talks. You are encouraged to direct a proposal to the program chair. State the main focus, include an outline, and emphasize why the topic of the talk is of general interest.

Panel sessions on technical issues are welcome. Persons interested in participating in panel discussions should also contact the program chair.

WORKS-IN-PROGRESS REPORTS

These reports provide researchers, developers, and implementors with ten minutes to speak on current work and receive valuable feedback. Present your interim results, novel approaches, or newly-completed work. Schedule your report in advance or on-site. Inquiries about WIP's should be directed to the program chair.

SUGGESTED TOPICS

- Graphical User Interfaces - The X Window System. User Interface Design and Standards. Open Look, Motif, and NeWS. Style guides and toolkits. Importance of consistency and ease of use.
Dates For Referred Paper Submissions

- Extended Abstracts Due: January 10, 1994
- Notifications to Authors: January 26, 1994
- Final Papers Due: March 11, 1994

For Program and Registration Information

Materials containing all details of the technical and tutorial programs, conference registration, and hotel and airline discounts and reservation information will be mailed in mid-February 1994. 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

Submission Details

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

A submission must be in the form of an extended abstract (1500-2500 words, 3-5 pages in length). The extended abstract should represent your paper in short form. It should demonstrate that you have a real project, that you are familiar with the work in your area, and that you can clearly explain yourself.

Papers will be judged on technical merit, relevance to the theme, and suitability for presentation. Software and hardware developers who wish to share their experiences, innovative solutions, and techniques are encouraged to submit papers.

Please submit one copy of an extended abstract (e-mail preferred) via:
- E-mail: <app-dev-sub@math.psu.edu>
- FAX: +1 (814) 865-3735 to Jim Duncan re: USENIX App Dev 94
- Postal mail: Jim Duncan
  USENIX App Dev 94
  Math Depart., Penn State University
  218 McAllister Building
  University Park PA USA 16802

Please refer to "USENIX App Dev 94" on all FAXes and postal mail. Please direct inquiries regarding the symposium to <jim@math.psu.edu>. 
Fourth International Workshop On Network And Operating System Support For Digital Audio And Video
November 3 – 5, 1993
Lancaster House Hotel
Lancaster, UK

In cooperation with:
ACM SIGCOMM, ACM SIGOPS, ACM SIGOIS, IEEE,
USENIX Association

Relevant Areas

Important topics for the workshop include (but are not limited to):

- Broadband/ATM networks
- Multimedia network interfaces
- Comms protocols for multimedia
- Microkernel and OS support
- Application of real-time techniques
- Media synchronisation
- Quality of service support
- Multimedia storage and I/O architectures
- Distributed multimedia systems
- Integrative standards, e.g. TINA and ODP
- Performance studies

Workshop Theme

Network and operating system support for digital audio and video is currently an area of intense activity, and a number of core techniques are beginning to emerge. The emphasis of this workshop will be on integration of key technologies to produce complete solutions.

The role of the operating system is seen as particularly important in this respect. Papers are also welcomed on practical experiences of developing multimedia systems.

The workshop is intended to bring together practitioners from a variety of areas, including communications and networks, operating systems, real-time systems and distributed computing. It is intended that the workshop will produce an agreed position statement on the state of the art in the field, highlighting major areas requiring future research.

Instructions For Submitting Papers

Authors are requested to submit to <av-workshop@comp.lancs.ac.uk>, a 500–2000 word position paper or an extended abstract of a full paper (in raw, unformatted text) by electronic mail.

Only if electronic submission is impossible, papers may be sent to:

Prof. W.D. Shepherd
Computing Department
Engineering Building
Lancaster University
Lancaster LA1 4YR, UK
Fax: +44 524 381707
Phone +44 524 59 3827
Email: <doug@comp.lancs.ac.uk>

The proceedings of the workshop will be published by Springer-Verlag and the best papers will be forwarded to selected journals for publication.

Important Dates

Abstracts due: July 19, 1993
Acceptance notification: August 30, 1993
Final paper due: October 4, 1993

Program Committee

Doug Shepherd, Lancaster University, UK (Chair)
Gordon Blair, Lancaster University, UK
Katrin Braun, Siemens, Germany
Guy Cherry, Tektronix, USA
S. Christodoulakis, MUSIC, Greece
Robert Enser, AT&T Bell Labs, USA
Domenico Ferrari, U. of California, Berkeley
J.J. Garcia-Luna, SRI International, USA
Riccardo Gusella, HP Labs, USA
Ralf Herrenschwab, IBM ENC, Germany
Andrew Hopper, Olivetti Research Limited, UK
Jim Kurose, U. of Massachusetts, USA
Albert Kuendig, ETH, Zurich, Switzerland
Tom Little, Boston University, USA
Derek McAuley, Cambridge University, UK
Richard Nicol, BT Labs, UK
Duane Northcutt, Sun Labs, USA
Steve Pink, SICS, Sweden
Radu Popescu-Zeletin, GMD FOKUS, Germany
P. Venkat Rangan, U. of California., San Diego
Jonathan Rosenberg, Bellcore, USA
Jean-Bernard Stefani, CNET, Paris, France
Daniel Swinehart, Xerox Parc, USA
Steve Wright, HP Labs, UK
Martina Zitterbart, U. of Karlsruhe, Germany
Call For Papers: Sun User Group
Eleventh Annual Conference
December 7–9, 1993
San Jose, California

Theme:
The Metamorphosis of SunOS: from classroom to boardroom
Suns in the corporate environment
Migrating from mainframes to Suns
Conversion to Solaris 2: a global issue

The Sun User Group is pleased to host its Eleventh Annual Conference and Exhibit, which will address the use of Suns in education, research, and business. Technical papers and presentations concerning this topic, as well as other topics of interest to the Sun/SPARC community, are invited. Manufacturers of computer equipment and software based on SPARC/Solaris technology are encouraged to participate in this conference with presentations, and participation in the conference exhibition.

SUG conferences are attended by members from all over the world. The 1992 conference and exhibit was attended by over 4000 people from 24 countries and 43 states.

Submission Guidelines:
Submissions should be in the form of extended abstracts (750 to 1000 words) and be sufficiently complete to allow the committee to understand and evaluate the submission. Abstracts should include:

1. Author name(s), postal and email address(es), and telephone number(s)
2. Presenter name(s), postal and email address(es), and telephone number(s)
3. Title of the paper
4. Time needed for presentation/questions (30, 45, 60, 90 min. spots)
5. Audio-visual requirements
6. Student paper entry (full-time students only)

Authors whose submissions are accepted will receive instructions for the preparation of final papers which will be published in the conference proceedings. The Presenter will receive one free registration for the conference. Registration for any tutorial must be purchased.

Important! All presentations will require a paper submission for inclusion in the conference proceedings.

Submit one hardcopy and one electronic copy to the program chair:

Peter Galvin
Systems Manager
Brown University
Computer Science Department
P.O. Box 1910
Providence, RI 10912

Email: <pg@cs.brown.edu>
Phone: 401/863-7623
Fax: 401/863-7719

Student Papers:
There will be an award for the best student paper. Be sure to indicate with your submission if you are a full time student. A cash prize and free registration will be awarded by the Conference Program Committee.

Deadlines:
Abstracts due: August 7, 1993
Notifications to authors: September 4, 1993
Final papers due: October 12, 1993

The Program Committee will select presentations from among those submitted. The committee consists of experts from many areas of the Sun/Solaris arena, including:

Pete Cottrell, U. of Maryland
Casper Dik, U. of Amsterdam
Dinah McNutt, Tivoli Systems
Steven Miller, U. of Illinois
Ruth Milner, Natl. Radio Astronomy Observatory
Gene Rackow, Argonne National Lab
Hal Stern, SMCC
Leonardo Topa, MIT
It will be aided by:

S. Lee Henry, SUG board liaison
*Johns Hopkins University*

Alex Newman, SUG liaison
*Sun User Group*

Michael Pearlman, SUG board liaison
*Rice University*

Possible themes and topics are listed below. These are only for reference, however, and all submitted papers will be considered for presentation at the conference.

**Thematic Track**

Topics directly related to the theme of the program: Migrating from mainframes to Sun systems, UNIX in the corporate environment, Suns and mission-critical work, Solaris 2 migration issues, ideas, and roadmap.

**Beginning UNiX/Beginning System Administration**

Practical information for new users and system administrators. Presentations which will allow a new Sun user to be able to install workstations and keep them running. They could include file system layout, basic security, NIS, DNS, the boot process, script programming, user dot files, administration files, day-to-day operations, keeping a system log, using the resources available on the Internet, and so on.

**Minitutorials and Q&A**

These sessions should be designed to directly address Sun users’ needs. They could include step-by-step guides to administration, networking, programming in various tools, and understanding aspects of system operation such as performance and utilities. Q&A sessions are important and interesting to attendees because of their interactive, problem solving and question-resolving nature. Previous talks in this vein have included “securing your environment” and “system administration tips and tricks.”

**System Administration, System Security**

Talks in this area should address the interests of those who have been SunOS users for a year or more. Some of the more in-depth topics: mixed environments, backups, PPP/SLIP, automounter, perl, tools for problem troubleshooting, and remote off-site administration.

**Programming and Development Environments**

These presentations concern Sun’s programming languages and those offered by third-parties. With Sun’s C compiler no longer bundled with the operating system there are opportunities for third party compilers. Also of interest are tools and techniques for program development.

**Technical Product Information**

This topic provides a chance for vendors to beat their own (technical) drums and describe the compelling technical advantages of their products. Panels of competitive products will be assembled when it seems appropriate to do so. No sales-oriented or nontechnical talks will be accepted.

**Suns in the Office/Home**

How do you integrate Sun computers into the office? Topics can include PC-type products on the Sun, how well they emulate, communicate, or convert information and how well they are received, the interface to IBM PC and Apple networks, mail and printer access, home hardware maintenance, links to the office (PPP/SLIP), product licensing, and Solaris on the Intel platform.

**Research, Real-time computing, Image Processing, Scientific**

Although there are other conferences that deal solely with the technical issues of research, this topic deals with how a Sun system facilitates research, and tools which can help the scientist. Topics include Suns and real-time computing, cross compiling (i.e., vxworks, os9, vrtx etc.), and data acquisition, among others.
Well...not really. But your Sun User Group membership you can save hundreds of dollars a year -- and that's a whole lot of lunches.

Here's what you do get for your $40 yearly membership in SUG:

**$100 combined discount on registration at SUG's two annual conferences -** The place to go to see the state of the art in SUN/Sparc technology!

**Discounts worth hundreds of dollars with major technical publishers like Addison-Wesley, O'reilly & Associates, and Prentice Hall!**

**Discounts on software and hardware from such leading vendors as Dux, Qualix, Software Moguls, Dataram worth hundreds more!**

**The opportunity to purchase SUG's award-winning software libraries. Hundreds of megabytes of useful software and archives, in one convenient compact disk. Available only to SUG members.**

**A subscription to Readme - SUG's bimonthly newsletter, brimming with news, reviews, and interviews that will keep you up to date with the rapidly changing Sun/Sparc environment!**

**Access to SUG's Members-only Electronic mailing list: the hot new forum for exchanging technical information and getting the inside scoop on What's going on in the workstation world!**

If you use even one of these benefits, you will come out ahead - use a few, and you will consider your SUG membership to be the most valuable peripheral your SUN/Sparcstation is connected to!
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!
## 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 Winter '93</td>
<td>33</td>
<td>40</td>
<td>$___</td>
<td>$25</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>San Antonio Summer '92</td>
<td>23</td>
<td>30</td>
<td>$___</td>
<td>14</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>San Francisco Winter '92</td>
<td>30</td>
<td>39</td>
<td>$___</td>
<td>22</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Nashville Summer '91</td>
<td>32</td>
<td>38</td>
<td>$___</td>
<td>22</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Dallas Winter '91</td>
<td>28</td>
<td>34</td>
<td>$___</td>
<td>18</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Anaheim Summer '90</td>
<td>22</td>
<td>22</td>
<td>$___</td>
<td>15</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Washington, DC Winter '90</td>
<td>25</td>
<td>25</td>
<td>$___</td>
<td>15</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Baltimore Summer '89</td>
<td>20</td>
<td>20</td>
<td>$___</td>
<td>15</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>San Diego Winter '89</td>
<td>30</td>
<td>30</td>
<td>$___</td>
<td>20</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>San Francisco Summer '88</td>
<td>29</td>
<td>29</td>
<td>$___</td>
<td>20</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Dallas Winter '88</td>
<td>26</td>
<td>26</td>
<td>$___</td>
<td>15</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Phoenix Summer '87</td>
<td>35</td>
<td>35</td>
<td>$___</td>
<td>20</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Washington, DC Winter '87</td>
<td>10</td>
<td>10</td>
<td>$___</td>
<td>15</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Atlanta Summer '86</td>
<td>37</td>
<td>37</td>
<td>$___</td>
<td>20</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Denver Winter '86</td>
<td>25</td>
<td>25</td>
<td>$___</td>
<td>15</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Portland Summer '85</td>
<td>45</td>
<td>45</td>
<td>$___</td>
<td>25</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Dallas Winter '85</td>
<td>15</td>
<td>15</td>
<td>$___</td>
<td>10</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Salt Lake City Summer '84</td>
<td>29</td>
<td>29</td>
<td>$___</td>
<td>20</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Washington, DC Winter '84</td>
<td>25</td>
<td>25</td>
<td>$___</td>
<td>15</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Toronto Summer '83</td>
<td>32</td>
<td>32</td>
<td>$___</td>
<td>20</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>San Diego Winter '83</td>
<td>28</td>
<td>28</td>
<td>$___</td>
<td>15</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td><strong>LARGE INSTALLATION SYSTEMS ADMINISTRATION</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LISA VI Oct. '92</td>
<td>23</td>
<td>30</td>
<td>$___</td>
<td>12</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>LISA V Sept. '91</td>
<td>20</td>
<td>23</td>
<td>$___</td>
<td>11</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>LISA IV Oct. '90</td>
<td>15</td>
<td>18</td>
<td>$___</td>
<td>8</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>LISA III Sept. '89</td>
<td>13</td>
<td>13</td>
<td>$___</td>
<td>9</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>LISA II Nov. '88</td>
<td>8</td>
<td>8</td>
<td>$___</td>
<td>5</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>LISA I April '87</td>
<td>4</td>
<td>4</td>
<td>$___</td>
<td>5</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td><strong>C++</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C++ Conference Aug. '92</td>
<td>30</td>
<td>39</td>
<td>$___</td>
<td>20</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>C++ Conference April '91</td>
<td>22</td>
<td>26</td>
<td>$___</td>
<td>11</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>C++ Conference April '90</td>
<td>28</td>
<td>28</td>
<td>$___</td>
<td>18</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>C++ Conference Oct. '88</td>
<td>30</td>
<td>30</td>
<td>$___</td>
<td>20</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>C++ Workshop Nov. '87</td>
<td>30</td>
<td>30</td>
<td>$___</td>
<td>20</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td><strong>SECURITY</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UNIX Security III Sept. '92</td>
<td>30</td>
<td>39</td>
<td>$___</td>
<td>11</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>UNIX Security II Aug. '90</td>
<td>13</td>
<td>16</td>
<td>$___</td>
<td>8</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>UNIX Security Aug. '88</td>
<td>7</td>
<td>7</td>
<td>$___</td>
<td>5</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td><strong>MACH</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mach Symposium III April '93</td>
<td>30</td>
<td>39</td>
<td>$___</td>
<td>18</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Mach Symposium Nov. '91</td>
<td>24</td>
<td>28</td>
<td>$___</td>
<td>14</td>
<td>$___</td>
</tr>
<tr>
<td></td>
<td>Mach Workshop Oct. '90</td>
<td>17</td>
<td>20</td>
<td>$___</td>
<td>9</td>
<td>$___</td>
</tr>
</tbody>
</table>

*Continued - see reverse*
<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>DISTRIBUTED &amp; MULTIPROCESSOR SYSTEMS (SEDMS)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SEDMS III</td>
<td>Mar. '92</td>
<td>30</td>
<td>36</td>
<td>20</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>SEDMS II</td>
<td>Mar. '91</td>
<td>30</td>
<td>36</td>
<td>20</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>SEDMS</td>
<td>Oct. '89</td>
<td>30</td>
<td>30</td>
<td>20</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>GRAPHICS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Graphics Workshop V</td>
<td>Nov. '89</td>
<td>18</td>
<td>18</td>
<td>10</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>Graphics IV</td>
<td>Oct. '87</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>Graphics III</td>
<td>Nov. '86</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>Graphics II</td>
<td>Dec. '85</td>
<td>7</td>
<td>7</td>
<td>5</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>OTHER WORKSHOPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>File Systems</td>
<td>May '92</td>
<td>15</td>
<td>20</td>
<td>9</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>UNIX Transaction Processing</td>
<td>May '89</td>
<td>12</td>
<td>12</td>
<td>8</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>Software Management</td>
<td>Apr. '89</td>
<td>20</td>
<td>20</td>
<td>15</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>UNIX &amp; Supercomputers</td>
<td>Sept. '88</td>
<td>20</td>
<td>20</td>
<td>10</td>
<td>$</td>
</tr>
</tbody>
</table>

Discounts are available for bulk orders. Please inquire.

PAYMENT OPTIONS*

- Check enclosed - payable to USENIX Association
- Charge my: ______ VISA ( ) ______ MC ( )
- Purchase order enclosed

* 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.

Please mail or fax this order form with payment to:

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710
FAX 510/548-5738

- If you are not a member and wish to receive our membership information packet, please check this box.

**If you are paying member price, please include member's name and/or membership number.**
MEMBERSHIP INFORMATION

Any individual or institution may become a member of the USENIX Association by filling out an application form and paying the appropriate annual fee.

There are five classes of membership:

Student: $20
Open to any full-time student at an accredited educational institution. A copy of the current student I.D. card must be provided.

Individual: $65
Open to any individual or institution. Individual Members may vote.

Corporate: $325
Corporate Membership is open to any individual or institution.

Educational: $160
Educational Membership is open to accredited educational institutions.

Supporting: $1000
Open to any individual or institution that wants to support the Association to a greater degree than through the Corporate Membership fee.

Corporate, Educational and Supporting members have one designated representative who receives all services available to Individual Members, plus copies of the proceedings from all conferences and symposia that are held during the term of membership.

SAGE, the System Administrators’ Guild

The System Administrators’ Guild (SAGE) is a Special Technical Group within the USENIX Association, devoted to the furtherance of the profession of systems administration. SAGE brings together system administrators for professional development, for the sharing of problems and solutions, and to provide a common voice to users, management, and vendors on topics of system administration.

A number of working groups within SAGE are focusing on special topics such as conferences, local organizations, professional and technical standards, policies, system and network security, publications, and education. USENIX and SAGE will work jointly to publish technical information and sponsor conferences, tutorials, and local groups in the systems administration field.

To become a SAGE member you must be a member of USENIX as well.

MEMBERSHIP APPLICATION

New ☐ Renewal ☐

Name ________________________________

Address ________________________________

City __________________ State ______ Zip ______ Country ______

Phone __________________ email address: __________________

☐ $20 Student (full-time) ☐ $160 Educational Institution ☐ $25 - SAGE (Additional)
☐ $65 Individual ☐ $325 Corporate
☐ $1000 Supporting

PAYMENT OPTIONS

☐ My total amount for membership is $.
☐ Check enclosed payable to USENIX Association.
☐ Please charge my: ☐ Visa ☐ MasterCard (Educational and Corporate members only).

Account # __________ Exp. Date __________

Outside the U.S.A.? Please make your payment in U.S. currency by one of the following:

• Charge (Visa, MasterCard, or foreign equivalent)
• International postal money order
• Check - issued by a local branch of a U.S. Bank

Signature __________________________

$24 of your annual membership dues is for a one-year subscription to the newsletter, ;login:, and $30 of your dues is for a one-year subscription to the journal, Computing Systems.

USENIX Mailing List

☐ I do not want my address made available to other members.
☐ I do not want my address made available for commercial mailings.
Local User Groups

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

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 <fruug-info@fruug.org>.

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

FL - Melbourne

Meets the 3rd Monday of every month.

Space Coast UNIX User's Group
Steve Lindsey (407) 242-4766
<lindsey@onet.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
<mcrtys!tony>
Ed Gallizzi, Ph.D. (813) 864-8272
<e.gallizzi@compmail.com>
Jay Ts (813) 979-9169
<uunet!pdn!tscs!metran!jan>
Dave Lewis (407)242-4372
<dhl@ccd.harris.com>

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@csitp.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!galstl!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@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
<mecc/pjh>

**NM – Albuquerque:**
ASIGUNIX meets every 3rd Wednesday of each month. Phil Hortz 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
<unigroup@murphy.com>
Bob Young
(212) 490-8470

**OK - Tulsa:**
Meets 2nd Wednesday of each month.

Tulsa UNIX Users Group, "$USR"
Stan Mason (918) 560-5329
<tuulix@usr.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@cactus.org>
Tom Painter (512) 835-5457
<president@cactus.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.com>

**TX – Houston:**
Meets 3rd Tuesday of each month.

Houston UNIX Users Group
(Hounix) answering machine (713) 684-6590
Bob Marcum, President (713) 270-8124
Chuck Bentley, Vice-president
(713) 789-8928
<chuckb@hounix.uucp>

**WA – Seattle:**
Meets monthly.

Seattle UNIX Group Membership Info.
Bill Campbell (206) 947-5591
6641 East Mercer
Mercer Island, WA 98040-0820
<bill@celestial.com>

**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
<deva@revcan.ont.ca>
**BAY LISA**

The Bay-LISA group meets monthly in Santa Clara, CA to discuss topics of interest for administration of sites with more than 100 users and/or computers.

Send email to <baylisa-info@sysadmin.com> or contact:

Bjorn Satdeva
408/ 241-3111
<bjorn@sysadmin.com>

---

**Prime Time Freeware**

**Volume 2, Number 1**

PTF 2-1 contains 1200 MB of compressed archives, unpacking to more than 3000 MB of source code and documentation, current as of January 15, 1993. Here are some of the larger items, prefaced by their compressed sizes:

<table>
<thead>
<tr>
<th>KB (C) Package Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>299508 UK TeX (LaTeX, etc.) Archive</td>
</tr>
<tr>
<td>82198 SRC Modula-3</td>
</tr>
<tr>
<td>81650 NetLib Archive (math and sim.)</td>
</tr>
<tr>
<td>78792 ICOT (5th Gen. AI Code)</td>
</tr>
<tr>
<td>36548 InterViews (UI dev. system)</td>
</tr>
<tr>
<td>36540 Scheme Language</td>
</tr>
<tr>
<td>34040 Ptolemy (HW CAD and sim. system)</td>
</tr>
<tr>
<td>26964 StatLib Archive (statistics.)</td>
</tr>
<tr>
<td>26206 Cygnus Progressive Release</td>
</tr>
<tr>
<td>19773 POSTGRES RDBMS</td>
</tr>
<tr>
<td>18046 CMU Common Lisp</td>
</tr>
<tr>
<td>15752 AnalyticCalc</td>
</tr>
<tr>
<td>15457 Icon Language</td>
</tr>
<tr>
<td>12748 Barkley Tcl/Tk Archive</td>
</tr>
<tr>
<td>12478 Self Language</td>
</tr>
<tr>
<td>12070 G++ (etc.) for DOS</td>
</tr>
<tr>
<td>11710 Lucid Emacs</td>
</tr>
<tr>
<td>11672 GoPATH (UI Dev. System)</td>
</tr>
<tr>
<td>11137 GCC (C++ compiler, etc.)</td>
</tr>
<tr>
<td>11134 Std. ML of New Jersey</td>
</tr>
</tbody>
</table>

---

**Volume 2, Number 2**

PTF 2-2 will be available in July, 1993. It will contain updated copies of small and/or updated packages from PTF 2-1, an updated selection of materials from PTF 1-2, and a variety of totally new material. Contact PTF <ptf@cfcl.com> for details, after July 1.

**Ordering and Prices**

PTF issues cost $60 (a special discounted price of $50 for SUG and USENIX members), plus 7% tax if in California and shipping/handling charges, as follows:

<table>
<thead>
<tr>
<th>S/H Item 1</th>
<th>Items 2-N</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA $5</td>
<td>$1 each</td>
</tr>
<tr>
<td>Foreign $10</td>
<td>$4 each</td>
</tr>
</tbody>
</table>

PTF takes Mastercard and Visa, postal money orders in US funds, and checks in US funds that are payable through a us bank. Either issue (two ISO-9660 CD-ROMs and a 50+ page booklet) can be ordered from:

Prime Time Freeware
370 Altair Way, #150
Sunnyvale, CA 94086
Tel: +1 408/433-9662
Fax: +1 408/432-6149
Email: <ptf@cfcl.com>
Calendar of Events

1993

Jun  5-11  DECUS, Atlanta, GA
     8-10  CERT Security Seminars
     21-25  USENIX, Cincinnati, OH
Jul  7- 9  SAGE-AU, Melbourne, Australia
     12-16  IEEE 1003, Denver, CO
     27-29  Sun User Group-East, Boston, MA
Aug  1    ACM Siggraph, Anaheim, CA
     2 - 3  Mobile & Location-Independent Computing, Cambridge, MA
     10-13  FIRST Computer Security Incident Handling, St. Louis, MO
     17-20  INET '93, San Francisco, CA
     23-27  INTEROP, San Francisco, CA
Sept 20-22  Microkernels II, San Diego, CA
     23-24  SEDMS IV, San Diego, CA
     22-24  EuroOpen/GUUG, Wiesbaden, FRG
     26 -  ACM OOPSLA, Washington, DC
Oct  1     "    "
     27-30  AUUG, Sydney, Australia
Oct  4-6  UNIX Security Symposium IV, Santa Clara, CA
     14-15  WWOS -IV, Napa, CA
     18-22  IEEE 1003, Bethesda, MD
     25-29  INTEROP '93 Europe, Paris, France
Nov  1- 5  LISA VII, Monterey, CA
     3- 5  NOSSDAV Workshop, Lancaster, UK
     29 -  FedUNIX, Washington, DC
Dec  3      "    "
     4-10  DECUS, San Francisco, CA
     7- 9  Sun User Group, San Jose, CA
     8-10  JUS UNIX Fair, Japan

1994

Jan  10-14  IEEE 1003, Irvine, CA
     17-21  USENIX, San Francisco, CA
Mar 23-25  UniForum, San Francisco, CA
Apr 11-14  C++, Cambridge, MA
     25-28  UNIX Applications Development
             Toronto, Canada
Apr TBA  SANS III, Washington, D.C. area
May  2- 6  Networld+INTEROP 94, Las Vegas, NV

May / June 1993

1995

Jan  16-20  USENIX, New Orleans, LA
Feb 21-23  UniForum, Dallas, TX
May 13-19  DECUS, New Orleans, LA
Jun 19-23  USENIX, San Francisco, CA
Nov  2- 8  DECUS, San Francisco, CA

1996

Jan  22-26  USENIX, San Diego, CA
Mar 12-14  UniForum, San Francisco, CA
May 18-24  DECUS, Orlando, FL
Nov 16-22  DECUS, Anaheim, CA

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

* = events sponsored by the USENIX Association.

ACM: Association for Computing Machinery
AUUG: Australian UNIX Users Group
CERT: Computer Emergency Response Team
DECUS: Digital Equipment Computer Users Society
EuroOpen: European Forum for Open Systems
FIRST: Forum of Incidence Response & Security Team
IEEE: Institute of Electrical and Electronics Engineers
IETF: Internet Engineering Task Force
INET: Internet Society
Interex: Intl. Association of Hewlett-Packard Comp. Users
JUS: Japan UNIX Society
LISA: USENIX Systems Administration Conference
NOSSDAV: Network and Operating System Support for Digital Audio and Video
OOPSLA: Object - oriented Programming Systems, Languages, and Applications
SAGE: System Administrators' Guild
SANS: System Administration, Networking & Security
SEDMSS: Symposium on Experiences with Distributed and Multiprocessor Systems
UKUUG: United Kingdom UNIX Systems Users Group
UniForum: International Association of UNIX and Open Systems Professionals
WWOS: Workshop on Workstation Operating Systems
**UPCOMING SYMPOSIA AND CONFERENCES**

### 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

### 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

### OCTOBER 4–7, 1993
**4TH UNIX SECURITY SYMPOSIUM**
- Program Chair: Bill Cheswick, AT&T Bell Laboratories
- Santa Clara Marriott Hotel, Santa Clara, California
- Co-sponsored with SAGE, the Systems Administrators' Guild

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

### JANUARY 17–21, 1994
**WINTER 1994 TECHNICAL CONFERENCE**
- Program Chair, Jeffrey Mogul, Digital Equipment Corporation
- San Francisco Hilton, San Francisco, California

---

**TO RECEIVE FULL INFORMATION**

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