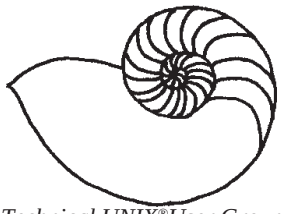
Volume 4, Number 2 December 1991 $2.50



TUUG Lines

Technical UNIX®User Group

Newsletter of the Technical UNIX® User Group

First Unix Symposium Deemed a Success

By Allan Moulding

Winnipeg's first UNIX symposium, which was held on November 4th and 5th, was a success, with about 150 attendees each day and about 120 of those in attendance both days. This was a large enough crowd that both the Winnipeg chapter of the Canadian Information Processing Society (CIPS) and TUUG made a profit. (Hurray!)

There were three tracks of five sessions each, covering management, technical, and novice areas, with management sessions having the biggest attendance of the three.

There was also a vendor exhibit area with a number of the major UNIX system vendors. Comments for improvement for future shows were that there should have been more software vendors and that there should have been more exhibit only attendees, so that it would be busier during the time when sessions were on. I liked it because the companies had their workstations there and those models weren't usually shown at business/computer shows in Winnipeg.

Rocky Nystrom, of Information Foundation in Denver, Colorado, gave the keynote address — a talk on the history of UNIX and the impact of open systems on the industry. Although I didn't see it, I heard his other topic, a management track session called Migrating to Open Systems, was even better than the keynote address. A video of that session might be shown at the January meeting.

Of the sessions I attended, I liked the session entitled Single System Image, by Pat Smith of Systemhouse, the best. He gave an example of decentralization of government services in Los Angeles, and the technologies involved in doing so.

Another session I saw was Understanding the Wonder of "C," by Wadson Tseng of IBM. I would have liked to see a more technical discussion of the language rather than an overview of the language, but it was kept at a lower technical level due to the possibly wide range of skill levels in the audience.

And last, but certainly not least, I would like to thank Susan Zuk and Al Hykaway, who put a lot of effort into getting this symposium together and making it the success that it was.

THIS MONTH'S MEETING

Meeting Location:

This year, we are invited to a Casino Night, being held by CIPS, at the Rorie Street Marble Club on December 10th, from 6:00 PM to 9:00 PM. Cost is $15, and includes a hot buffet dinner, and a chance to win fun prizes! Tickets will be available by contacting Susan Zuk at 788-7312. Friends, spouses, and dates are welcome to attend.

Meeting Agenda:

Eat, drink, and be merry; after all, it's almost Christmas time!

INSIDE THIS ISSUE

The Need for Standard Interfaces

By Gilbert Detillieux

The term “Open Systems” is all the rage now. Look at any computer magazine now, not only those dealing with UNIX, and you'll see the term. Even computer commercials during the recent Grey Cup game talked about open systems architecture. It was certainly fitting, and timely, that our recent symposium had that as its theme.

Looking beyond the hype, for a meaningful description of open systems, another term consistently reappears, and goes hand in hand with it – standard interfaces. This is, after all, what open systems are all about. It is standard interfaces, whether to users, to operating systems, to networks, or to other applications, that permit computer systems and applications from different companies to coexist, and allow greater user productivity.

In the symposium presentations, described later in this month’s industry article, the need for standard interfaces was stated many times, from the keynote address to the closing, and in many of the seminars in between.

Standard interfaces are what allow interoperability in hardware, such as modems. The variety that comes about as technology evolves can be confusing. This month’s hands-on article tries to straighten out the confusion in modem standards.

Finally, standard interfaces need not be a hindrance to technological innovation. In fact, the opposite is often true. In this month’s technology article, we see some of the innovation that is possible in an operating system, while maintaining compatibility with current standards.

The 1991-1992 Executive

President:	Susan Zuk	(W) 788-7312
Past President:	Eric Carsted	1-883-2570
Vice-President:	Richard Kwiatkowski	
Treasurer:	Rick Horocholyn	
Secretary:	Roland Schneider	1-482-5173
Membership Sec.:	Allan Moulding	269-8054
Mailing List:	Gilles Detillieux	489-7016
Meeting Coordinator:	Eric Carsted	1-883-2570
Newsletter editor:	Gilbert Detillieux	489-7016
Information:	Susan Zuk	(W) 788-7312
	(or)Gilbert Detillieux	489-7016

Copyright Policy and Disclaimer

Our Address

Technical UNIX User Group
P.O. Box 130
Saint-Boniface, Manitoba
R2H 3B4

Internet E-mail:
tuug@cs.umanitoba.ca

Group Information

The Technical UNIX User Group meets at 7:30 PM the second Tuesday of every month, except July and August. The newsletter is mailed to all paid up members one week prior to the meeting. Membership dues are $20 annually and are due at the October meeting. Membership dues are accepted by mail and dues for new members will be pro-rated accordingly.

Thoughts About the Symposium and TUUG

By Susan Zuk, President

As you know, the UNIX Symposium has come and gone. The hard work and planning really paid off! For those of you who were there, we hope you attended some seminars that helped you to understand the UNIX environment more and informed you of the multitude of ways that UNIX can add value to your shop.

From the evaluations that we received, it sounds like many people would like to see this event offered again. We have been discussing the possibility of holding this function once every two years. This decision will be made in the upcoming months.

The second point that was mentioned was the need for more applications to be displayed. We might be able to handle this requirement at our monthly meetings. If you have a specific topic or application you would like to see presented, please give our meeting coordinator, Eric Carsted, a call. He will try to accommodate your request. You can find his number on the previous page.

I would like to thank CIPS (Canadian Information Processing Society) for teaming with us. This event would not have been the success it was without their support. I also would like to thank all the corporate sponsors, speakers, and volunteers for all their hard work and time. This is the type of event that really shows community spirit.

Now on to new business: On Tuesday, December 10th, we are invited to join CIPS at their annual Casino Night, to be held at the Marble Club. This is a fun event where you can gamble with play money. This play money gives you the opportunity to win some great prizes in a Chinese Auction. Tickets for this event are $15.00 and include a hot buffet dinner. Call me to get tickets. Everyone is invited, so you can bring friends and/or spouses. Come on out for a fun evening and to celebrate in the holiday spirit.

A final notable mention is about UNIFORUM. This is the International Association of Open Systems Users, formerly known as the International Association of UNIX System Users. You may have heard us discuss this group either at our local meetings or heard Gerry Jolicoeur, President of the Ottawa UNIFORUM chapter, speak of this group at the UNIX Symposium. We are investigating our options as to whether to become a chapter and how it is accomplished. If you have anything to contribute or would like to see us join, please give me a call. I do not know if members have any opinions on this subject. We will probably be writing an article about this group and bringing this to a vote in the next couple of months.

I would like to take this opportunity to wish you and yours a very happy and healthy holiday season and we look forward to seeing you at the Casino Night. Take care.

William Safire's Rules for Writers

Found in a newsgroup article.

Submitted by Gilles Detillieux

Remember to never split an infinitive. The passive voice should never be used. Do not put statements in the negative form. Verbs has to agree with their subjects. Proofread carefully to see if you words out. If you reread your work, you can find on rereading a great deal of repetition can be avoided by rereading and editing. A writer must not shift your point of view. And don't start a sentence with a conjunction. (Remember, too, a preposition is a terrible word to end a sentence with.) Don't overuse exclamation marks!! Place pronouns as close as possible, especially in long sentences, as of 10 or more words, to their antecedents. Writing carefully, dangling participles must be avoided. If any word is improper at the end of a sentence, a linking verb is. Take the bull by the hand and avoid mixing metaphors. Avoid trendy locutions that sound flaky. Everyone should be careful to use a singular pronoun with singular nouns in their writing. Always pick on the correct idiom. The adverb always follows the verb. Last but not least, avoid cliches like the plague; seek viable alternatives.

UNIX Symposium 1991 Highlights

The two-day symposium was a very busy time for all of us at TUUG.
With three parallel tracks of seminars, it was impossible to attend everything.
Here are the highlights of what the editor was able to catch.

By Gilbert Detillieux

The UNIX Symposium 1991, organized by CIPS and TUUG, was the first gathering of UNIX users and vendors of this scale in Winnipeg. The two day agenda included 15 seminar sessions in three parallel tracks, a keynote speaker, a technical panel discussion, and a vendor exhibit area.

Each day started off with registration, then breakfast. Things started rolling at 9:00 AM, with the keynote speaker's address, on the first day.

The Role of UNIX in the Open Systems Environment

Rocky Nystrom, of Information Foundation in Denver, Colorado, gave his keynote address on this topic, which was also the theme of the symposium. This was a one hour presentation, attended by all first-day registrants.

He began by defining the term "open systems" and the requirements of an open system: compatibility, portability, scalability, and interoperability. He then went on to explain why both end-users and vendors are moving to UNIX. One of the main benefits for both is technology independence, the fact that UNIX exists on such a wide variety of machines, and can easily be ported to new hardware technology.

He then went on to show how UNIX System V meets the requirements of open systems, and what the benefits are for developing applications. After describing some of the standards for UNIX, such as the System V Interface Definition, X/Open's Common Application Environment, and POSIX, and also talking about some of the tools available to application developers, he concluded by saying that UNIX plays a key role in the open systems environment, since it allows portability of the development tools, and the resulting applications.

The Exhibit Area

The exhibit area, consisting of 13 exhibitor booths, was particularly busy at the coffee breaks and at lunch, as well as during the designated vendor exhibit time slot. Booths were arranged in a U shape, making it easy to see every booth. Several booths had an Ethernet network connection, allowing communication, and access to services provided by other systems on the net. This network, as well as a server system, were provided by the University of Manitoba Computer Services department. It was hoped that a "live" Internet hook-up could be provided, but that could not be arranged in time for the symposium.

Unisys had a variety of systems on display, from PC's, to workstations, to tower-type servers. A new workstation, based on the SPARC processor, was introduced. This system runs standard SunOS, and also has a 386 coprocessor for DOS.

Amdahl, maker of IBM mainframe compatible systems, was also at the show. This booth was the only one to not have any equipment on display. (It would have been quite difficult to set up a mainframe system in this area.)

Network Computing Devices (NCD) had a few of their X Window terminals on display, including a brand new 19" colour unit with a RISC processor. Demos were being run from one of their UNIX systems (a Sun SPARCstation), and from the U of M's server system.

IBM was there, of course, and in its booth, displayed its RS/6000 workstations. CSB Systems, an IBM VAR, had an adjacent booth.

Microstructure, a dealer for Altos and various other hardware, as well as a systems integrator for business accounting systems, had some of their equipment on display. Unfortunately, some of newer equipment they hoped to show was not delivered in time.

Hewlett Packard showed their Model 730 (snake) workstation, and one of their X Window terminals. An SNMP monitor displayed a graphical diagram of the network, and network usage statistics.

Data General showed their Aviion workstation, based on the Motorola 88000 RISC processor.

Digital Equipment showed their DECstation 5000/200 workstation, based on the MIPS R3000, as well as a DEC PC system.

Sun Microsystems, who were late in setting up, due to equipment shipping problems, showed a couple SPARCstation workstations.

M&L Data Services, an NCR reseller, showed some NCR equipment, as well as software products they carry.

Electro Systems Group showed various peripherals, especially a variety of printers that they sell.

Overcoming the Barriers in an Organization's Decision to Establish an Open Systems Environment

Gerald Jolicoeur, of Uniforum Canada, opened the management track seminar series with this talk. Using his experience as a manager and EDP executive within the federal government, he outlined many of the major impediments to adopting new technology, particularly within large organizations. He then went on to describe some of the ways that management can deal with the myths and presumptions that create the barriers.

Migrating to Open Systems

Rocky Nystrom, the keynote speaker, covered this topic in the second management track seminar. There were 81 people in attendance for this talk, by far the highest of any of the seminars. To make the migration manageable, he argued, one should focus on new applications, and enhancements to existing applications, as candidates for implementation on open systems, rather than attempting a complete switch for all existing applications.

He then described the Application Connectivity Engineering (ACE) approach, in which new applications and application enhancements are implemented on open systems, with appropriate connections to existing applications. An important part of the process, he argued, is the need for application prototyping. He compared this to the job of a police sketch artist (programmer), working on a sketch (prototype), based on information provided by the victim (user).

The benefits of this approach were illustrated by three case studies, using work his company had done for American Airlines, Sterling Inc., and the U.S. Army, as examples.

Technical Panel

A panel of six "experts," with experience in a variety of subjects related to UNIX, was assembled. Eric Carsted, past president of TUUG, was the moderator. Questions from the audience, on a wide variety of technical topics, were handled by whichever panellists were able to provide an answer. This was a more structured version of our group's round table discussions.

X Terminals

Kevin Wall, of NCD, gave this presentation as the third technical track seminar. He started off by showing the evolution of computing, from time-sharing, to fully distributed computing, then finally to network computing. He then described the benefits of this approach.

He then went on to a detailed look at the X Window system. He listed some of the versions that have been developed, and which companies are endorsing these standards. A look at the X Window system architecture followed, with a description of the various elements in the system: client software, applications, window managers, fonts, server software, and display devices. Finally, a comparison between X terminals and workstations was made, giving some of the relative benefits.

Single System Image

Pat Smith, of Systemhouse Inc, gave the last management track seminar. In this presentation, he describes an ideal user's perspective of information systems – the single system image. In this image, the users see all information they have a need and a right to access in a unified fashion, without regard to the location of this information, and the various technologies involved in obtaining it. As a case study, he described the distribution of government services within Los Angeles to local public libraries, how the public uses these services, and how they are implemented.

Modem Mumbo-Jumbo

Modem compatibility is an "alphabet soup" of de facto and industry standards.
Here is a simple guide to some of the options available today.

By Bill Garfield

Reprinted with permission from the May 1991 issue of the Muddy Water Computer Society newsletter.

At the risk of boring everyone with repetition, the question pops up so often that the following deserves repeating from time to time:

Modem Types

V.22 (no *bis*) describes a 1200 bps modem.
V.22bis (*bis* meaning "second") describes a 2400 bps modem.
V.32 (no *bis*) describes a 9600 bps modem.
V.32bis describes a 14,400 bps modem.

Modem Options

MNP-3 and MNP-4 are old methods of *error correction*.
MNP-5 is an old method of *data compression*.
V.42 (no *bis*) is a new method of *error correction*.
V.42bis is a new method of *data compression*.

Obviously then, you could have a modem which is *both* V.22bis and V.42bis... V.22, V.22bis, V.32, and V.32bis *all* refer to modulation, or more clearly, *speed* standards. V.42 and V.42bis refer to error correction and compression standards.

You should only consider MNP, V.42 and V.42bis as being *options* to V.22, V.22bis, V.32, V.32bis, and HST modems.

Describing a modem as being a "V.42 modem" or a "V.42bis modem" is completely wrong and ambiguous. You don't have a V.42 or V.42bis modem. What you have is a V.22, V.22bis, V.32 or V.32bis modem which *includes* V.42 or V.42bis as an option. You may also have an HST modem which includes V.42 or V.42bis as an option.

So, what talks to what, and at what speed?

Garfield's Famous Modem Speed Compatibility Chart

	V.32	V.32bis	HST964	HST1442	Dual1442	Dual1443
V.22	1200	1200	1200	1200	1200	1200
V.22bis	2400	2400	2400	2400	2400	2400
V.32	9600	9600	2400	2400	9600	9600
V.32bis	9600	14400	2400	2400	9600	14400
HST 964	2400	2400	9600*	9600*	9600*	9600*
HST 1442	2400	2400	9600*	14400*	14400*	14400*
Dual 1442	9600	9600	9600*	14400*	14400*	14400*
Dual 1443	9600	14400	9600*	14400*	14400*	14400

* uses proprietary HST modulation to achieve this speed

To display the model number of your USRobotics modem, enter the command "**ATI**" followed by the "Enter" key. Note: That's the letter **I**, not the numeral **1**.

As you can see, there are two models of the HST and two models of the Dual. The HST 964 is the oldest and slowest of the HST family, with a top speed of 9600 bps, and the least high speed "connectivity" of the HST family. Supplies of these should be reasonably well depleted from store stocks, but are readily available on the used market for around $350 or less. Look for them to continue to drop in value. The HST 964 cannot be upgraded, but does include MNP 4 and MNP 5.

Next came the HST modem 1442, the first HST ever capable of 14,400 bps operation. It was backward compatible with the earlier HST 964, but still somewhat limited in its high speed "connectivity." Many stores still have ample supplies. New prices will run close to $600. Expect to pay $400 to $500 for used ones. Just be sure it's the real thing, and not the older HST 964. The HST 1442 was a favourite among BBS operators (*sysops*) before the introductions of the "Dual Standard." The HST 1442 originally came with MNP. V.42 and V.42bis were available in later production units.

The HST/V.32 "Dual Standard" model 1442, introduced in 1989, was the first USRobotics product capable of a high speed (9600 bps) connection with a non-USR product. Until the recent introduction of the model 1443, the model 1442 Dual Standard was "the" modem to have if you could afford it. The 1442 Dual was seldom available in stores, but many mail order houses sold them for prices ranging from $800 to $1000. A few of them are just now becoming available on the used market, but they're still commanding top dollar at $600 and higher. Originally offered with MNP, V.42 and V.42bis were available in later units. The HST 1442 is capable of being upgraded to a "Dual" though conversion cost is substantial.

Last but not least, the top o' the line (so far) is USRobotics' latest entry into the warp speed modem market, the HST/V.32bis "Dual Standard" model 1443. Almost identical to the 1442, the 1443 "Dual" adds V.32bis modulation, making it capable of operating at maximum speed (14,400 bps) with V.32bis industry standard modems. The 1443 Dual includes V.42 and V.42bis options. New prices are the same as for the previous model 1442. The 1443 is not yet available on the used market. Prior models cannot be upgraded to this model. The "HST only" version is not available in the 1443 configuration.

***Bill Garfield** is a computer user here in Winnipeg, and shows up on various local bulletin boards. This article originally appeared as a message on the* MWCS BBS, *in response to some of the confusion about modem types.*

An Operating System Matures

The future of UNIX is now taking shape.
The day may come, though, when it no longer resembles today's versions.

By Rick Cook

Reprinted with permission from the January 1991 issue of CommUNIXations, published by UniForum.

If UNIX was a person, it would be old enough to drink, drive and vote. It was created in the late 1960's on a small minicomputer that had a large (for the time), fast (for the time) hard disk and teletype machines for terminals. Obviously, computing has come a long way since then and so has UNIX. Computing will continue to change in the 1990's and UNIX will have to keep up.

Basically, there is nothing "wrong" with UNIX. Its popularity (and longevity) is a testament to the soundness of its basic design. But an operating system is fundamentally an intermediary between computer hardware and its applications, and therefore it must change when they change.

While the original versions of UNIX served multiple character-based terminals from a central processor, today's most common UNIX systems are networks of PCs or workstations connected on local-area networks. A file server provides support. UNIX works fairly well in such a system as long as the active files and processors are in close proximity. It doesn't perform nearly as well in a massively distributed architecture with multiple processors. In such an environment, a single job may involve many resources, and parts of the job may be scattered across the system.

Distributed processing has to pass information and programs transparently between processors. It also needs a method of handling diffuse file systems. The popular idea for supporting widely distributed multiprocessing is *message passing*, in which threads (or tasks, depending on the operating system) communicate by sending messages back and forth to specific "ports" on other threads or tasks. For example, Chorus, a UNIX-like system that grew out of a French project to develop an efficient operating system for distributed multiprocessing, uses this method.

Message passing has the advantage of being able to work simply over a range of network sizes. It is mostly independent of how closely coupled the processors are and, because it is asynchronous, doesn't make assumptions about timing. The trade-off is that it adds overhead, though most operating system designers have found that acceptable in return for the decentralization and generality of design it offers.

The file system is a thornier problem and has produced a wider range of solutions. The most extreme is Plan 9 from Bell Labs, which creates a virtual file system for every user. More commonly, remote systems are treated as subdirectories on the local file system.

Another feature that is useful for distributed processing is *advanced virtual memory management*. For instance, Mach, the UNIX-like system created at Carnegie Mellon University that forms the basis of the OSF/1 and Next operating systems, goes well beyond the paging scheme in the Berkeley-derived versions of UNIX. Among other things, it allows the programmer to designate chunks of memory as read-only.

According to the *UNIX System V Roadmap* from UNIX International, future versions of AT&T's System V.4 will include support for multiprocessing, although at first only for symmetrical shared-memory processors. A later version will support loosely coupled distributed computing over a network. Some commercial implementations, such as Hewlett-Packard's HP-UX, now support limited synchronous and asynchronous multiprocessing.

UNIX multitasking is built around *processes*. As systems become more distributed and software and user interfaces more sophisticated, it would be helpful to break thing into smaller chunks. This is the idea behind *threads*, which are basic units of computing, smaller than processes. The ideal thread is the smallest piece of code in a program that can execute independently.

Threads run in a context called a *task* (in Mach) or an *actor* (in Chorus). These are collections of resources needed for the thread to execute, including but not limited to memory space. Several threads can run within the same task. Multithreading provides an efficient way to handle processes that are separate but need to work closely together.

Change is nothing new for UNIX. There have been major changes in UNIX, most notably the support for virtual memory. Researchers have been working on the current crop of problems for a decade. Solutions are at hand, most of which will be incorporated into future versions of UNIX. Most of these ideas have been tested in other operating systems.

The four UNIX-like systems profiled below have served as "test beds" for some key concepts. They range from experimental systems developed for research to full-blown, commercially available operating systems. Taken as a group, they provide a window on the future of UNIX.

Mach

Begun at Carnegie Mellon in work done for the Defense Advanced Research Projects Agency with cooperation from Digital Equipment Corporation, Mach is available on machines ranging from Sun workstations to DEC VAXes (although most of the implementations for these systems come from third parties such as Encore and Sequent), as well as being used by Next and the Open Software Foundation. Mach is UNIX-like and UNIX-derived, but it is not strictly UNIX. The current version, 2.5, was built using code from BSD4.2 and System V, but the next version, 3.0, will have no UNIX code in the kernel and a later release, 3.X, will have no UNIX code in the servers or libraries.

Mach's three main advantages over standard UNIX are its three basic concepts: multithreading, interprocess communications and advanced virtual memory management. "Consider an application that wants to do both sound and graphics," says Avi Tevanian, Jr., manager of system software at Next, Inc., in Redwood City, CA, and one of the developers of Mach. "You've got two flows of control within a program that are tightly coupled yet really separate.

The easy way to program that is to have multiple threads." Mach's designers worked hard to make interprocess communications run fast, which is important for graphical user interfaces like NextStep. "Our windowing system is based on a client/server model. That means applications need to communicate with the windowing system," Tevanian explains. "We have an interactive environment, which means that the turnaround time for message passing is very important."

Like most of these new operating systems, Mach has a lean kernel. It consists basically of support for threads, virtual memory and interprocess communications. Implementing those three fundamentals is essentially all the Mach kernel does.

Chorus

Like Mach, Chorus can resemble various operating systems, thanks to a stripped-down kernel — or *nucleus*, as its developers call it. The major interest today is in Chorus/Mix, which adds UNIX compatibility. The Chorus nucleus is built around a real-time executive, which contains the bare minimum of support. The executive has the mechanism for multitasking with threads, does thread scheduling and interrupt handling, and has primitives for synchronization and local communication among threads.

In addition, the basic version includes a machine-dependent supervisor that maps interrupt-handling code to interrupts. This makes for a tiny nucleus, only about 25K in its smallest versions. Chorus' thread-management mechanism is preemptive; that is, priority levels are assigned when threads are created.

These features make Chorus useful for embedded, real-time systems. A more complete implementation of the nucleus, including virtual memory management and interprocess communications, would take up about 100K. But in any case the real-time executive, supervisor, memory manager and process communications module are all that goes into the nucleus.

"Basically, our microkernel handled those three functions," says Will Neuhauser, president of Chorus Systems, Inc., in Beaverton, OR, the U.S. subsidiary of Chorus Systèmes, S.A., of Paris. "The fundamental thing it was designed for is distributed computing. Communications is message-based so we can build a true distributed operating system, not just distributed applications."

In Chorus, threads are grouped together in actors, or address spaces. Threads in the same actor can communicate directly through shared memory or by message passing, while threads belonging to different actors must use message passing. Messages pass to and from ports or groups of ports on the actors. It doesn't matter to the system where the actors are located.

Virtual Memory can also be distributed over the network. The Chorus virtual memory manager is demand-paged so pages are brought in only when needed. It offers separate protected address spaces and can share memory between different tasks.

Last fall, Chorus and Unisys Corporation announced a joint development agreement to develop a System V.4-compliant version of Chorus/Mix. (The present version complies with V.3.2.) Unisys plans to use Chorus/Mix as the basis for its version of enterprise-wide computing.

V

The V operating system is a research tool begun at Stanford University in 1981. "To the user it's a UNIX-like system but it's not a version of UNIX," says David Cheriton, associate professor of computer science at Stanford and one of the principal developers of V. "It was written from scratch."

V's significance is that it is built around communications. "The kernel consists of lightweight processes, communications between those processes, a virtual memory system and nothing else," says Cheriton. Even the job-level scheduler and longer-term scheduler are outside the kernel.

V is dedicated to the message-passing concept. Communications is based on it and the operating system is optimized for speedy message handling. This includes providing common message operations as single functions, rather than building them up out of primitives the way most operating systems do. "We recognized that the message primitives in a conventional operating system are often used simply to implement a remote procedure call," Cheriton says. "Rather than having simple primitives like *send* and *receive*, we have a combined primitive, which sends a request and gets a response back. That allowed many optimizations in reducing the number of kernel calls for making a remote procedure call." At Stanford, V is run over a network of Suns, MicroVAXes and DECstations connected by Ethernet. It also has run between processor cards on a VME bus.

Plan 9

The AT&T Bell Laboratories development team in Murray Hill, NJ, named their new operating system after what is generally regarded as the worst science-fiction movie ever made, *Plan 9 From Outer Space*. Like the movie, Plan 9 takes some getting used to but unlike it, Plan 9 from Bell Labs is no turkey. It is like UNIX in many ways and in fact several UNIX pioneers have worked on it, including Dennis Ritchie and Ken Thompson (original developers of UNIX) and Peter Weinberger (the "w" in the *awk* utility). "This was an opportunity to build a system that was well integrated in the distributed sense and also a simplification in the sense of using fewer fundamental ideas and pushing them harder," says Weinberger, now director of the software and systems research laboratory at Bell Labs.

Plan 9 classifies everything as servers. There are three different types of servers: terminals, which are disk-less workstations; CPU servers, which do the actual computing; and file servers, which hold all permanent files. In its present incarnation, Plan 9 assumes that all serious computing will be done on CPU servers. The terminals have enough processing power to handle display functions and run simple programs like text editors, but they have no permanent local storage — indeed, they cannot have any under Plan 9's file structure.

The CPU servers are basically interchangeable. It is easy to add more to the network to increase computing power. One advantage of this approach is that you don't have to replace everyone's workstation when you want to increase processing power: You just add another CPU server. File servers act as front ends for the file system and are the only places on the system where files reside permanently. The operating system makes no distinction between

local and remote file servers.

The file system is probably the most striking thing about Plan 9. At the user level there is no such thing as a file structure. Each user has his or her own file name space, which is the only file system the user sees. In UNIX, of course, the entire file system is organized as a tree branching out from a single root directory. In Plan 9 the file system is also organized as a tree but each user sees a different tree. Not only does each user have what amounts to a separate virtual file system but the entire environment is modeled in terms of that file system. This is true of UNIX, too, but Plan 9 takes the idea further. Everything the user has access to, from a window on the screen to an executing process, is a file or a directory.

The other striking feature of Plan 9 is that there is no difference between local and remote files. Like UNIX, Plan 9 has powerful facilities for communication between files. Combined with the ideas of private name space and non-locality of files, this opens a number of possibilities. "You can package all kinds of objects and make them look like pieces of the file system," Weinberger says. "One example is */proc*, which looks like a directory in the file system although it is made up of the kernel, and inside it is a file for each running process. In Plan 9 there is a directory for each process."

That directory has its own name space, which contains attachments to all the services the process is using, and through those services reaches the files in the services. For example, several windows on one screen can appear as files. "Programs can see what I have in my windows and act on that," says Weinberger. "Aside from the fact that they were started by the window manager, they can be completely independent of it."

Each window can be treated as a separate file and operated on like any other file. In fact a window manager can run inside a window, which is handy for debugging. Plan 9 is not a commercial product and there are no plans to make it one. However, Weinberger expects its concepts to influence the future direction of UNIX.

By Any Other Name

Many of these operating systems features are interesting and perhaps useful, but if they're included in an operating system is the result still UNIX? If we use the broad definition of UNIX as an operating system that will run UNIX programs, it is really a software interface and any operating system that conforms to that interface is UNIX — regardless of whether it is legally entitled to call itself that. Even in the narrow sense of UNIX as the operating system licensed from AT&T, the latest release, System V.4, includes many features that were not in the original versions.

The ancient Greek philosophers posed a question about the Athenian state barge, which was so old that, at one time or another, every plank, peg and scrap of rope on it had been replaced. Therefore, was this the same ship as the one launched over a hundred years ago? If not, when did it become a different ship?

Ultimately, UNIXes of the future will be like the Athenian barge. They will have the form of UNIX, which is to say the same software interface, but will be almost totally different in organization, detail and underlying philosophy. And like the barge, they will still take people where they want to go.

***Rick Cook** is a free-lance writer based in Phoenix, whose work has appeared in a variety of publications. He is a Contributing Editor of* CommUNIXations.

Enter the Microkernel

Among operating system mavens, *microkernel* is a hot concept. All of the operating systems mentioned in this article are built around the microkernel concept and UI's *System V Roadmap* points to a microkernel for a future version of UNIX.

A microkernel is a stripped-down kernel that contains only essential services and is reorganized into modules to make it easier to maintain. A microkernel also makes it easier to create minimal systems, such as for small personal computers or embedded applications. Most operating systems that take this approach have kernels of about 100K, although Chorus has a minimum configuration as small as 25K. The 100K size is perhaps one-fourth as large as the kernel in version 4.3 of Berkeley UNIX (BSD, an admittedly bulky UNIX).

UNIX began with a microkernel of sorts. Its creators made some fundamental decisions about what was really necessary in an operating system. Those fundamentals went into a small kernel that was coded to be efficient rather than easily maintained. Everything else, from the user interface to device drivers, was left out of the kernel.

Things in the kernel generally run faster than things outside it. As a result, there's a tendency to put more things inside. The natural result, Peter Weinberger points out, is a kernel with middle-aged spread. "The first couple of times through you can make the kernel simple and keep it simple," he says. "After 10 or 15 years, that's really hard to do."

The choice of what is important also changes over time. For example, communications wasn't initially considered basic enough to go into the kernel but the file system was. However, in a system characterized by massively parallel architectures or distributed processing (which are increasingly in demand today), communications is critical and the file system is better off outside the kernel because not all computers in the network necessarily have the same one.

The original UNIX kernel was not modular. As it got larger, it became increasingly difficult to maintain or for one programmer to understand. Modern microkernels are highly structured and modular, unlike the kernel in most versions of UNIX.

Hand in hand with modularity goes reorganization of kernel services. As mentioned, the trend is to have only the absolute minimum of services in the kernel. This produces the side effect that a new operating system may look like UNIX — or nearly any other operating system — depending on what is layered on top of its kernel. This is the secret of making operating systems that are as different from UNIX as Chorus and Mach look and act exactly like UNIX to applications and users.

TUUG Business Meeting Minutes

XCAN Grain Ltd.
TD Tower, 1200 - 201 Portage Ave.
Tuesday, November 12, 1991

Round Table: Discussion about UNIFORUM

Business Meeting: (Susan Zuk presided.)

A) President's Report:

Traditional December meeting wine and cheese will be replaced by Casino Night in conjunction with CIPS. Tickets are available from Susan Zuk.

B) Membership Secretary's Report:

Current paid-up membership is 38.

C) Newsletter Editor's Report:

The newsletter needs articles on a variety of topics. Goal is to have a technical, novice and management related article in each newsletter, as well as letters to the editor. A writeup on the symposium would be a good management article.

D) Treasurer's Report:

The year-end report was in the November newsletter. The TUUG name has been renewed for the next three years.

E) Meeting Coordinator's Report:

Next meeting is Casino Night:

Proposed January meeting presented topic: Videotape of Rocky Nystrom's symposium talk "Migrating to Open Systems"

Topics for future meetings: Presentations by SCO, Frame, Uniplex, book show and tells.

May have secondary meetings on specific topics.

F) New Business:

Moved by Gilbert Detillieux, seconded by Eric Carsted, that TUUG funds should be used to buy Al Hykaway and Susan Zuk gifts, in recognition of their work in organizing the UNIX Symposium. Carried.

Presented Topic:

Unify Database and Accell 4GL

TUUG Lines Newsletter Deadlines for 1991-1992 Fiscal Year

What follows is a list of deadline dates for each upcoming issue of the newsletter for the next fiscal year (including this issue).

The submission deadline is the last date on which material will be accepted by the editor for inclusion in the next issue of the newsletter. Material submitted after that date will be considered for subsequent issues.

The mailing deadline is the last date on which the newsletter should be mailed out to members, in order for it to be received sufficiently in advance of the next meeting. This is also likely to be the last date on which the TUUG executive meeting will be held.

Newsletter Issue	Submission Deadline	Mailing Deadline	Meeting Date
Dec '91	Nov 22	Nov 30	Dec 10
Jan '92	Dec 27	Jan 4	Jan 14
Feb '92	Jan 24	Feb 1	Feb 11
Mar '92	Feb 21	Feb 29	Mar 10
Apr '92	Mar 27	Apr 4	Apr 14
May '92	Apr 24	May 2	May 12
June '92	May 22	May 30	June 9
Sept '92	Aug 21	Aug 29	Sept 8
Oct '92	Sept 25	Oct 3	Oct 13

Next Month

Meeting:
Brrr... The cold January winds will likely be howling. What better way to spend the meeting night than watching a good videotape? We have Rocky Nystrom's popular management track seminar, Migrating to Open Systems, on tape and will likely be showing that after our usual round table and business meeting. Location TBA.

Newsletter:
A tutorial on magnetic tape formats and usage, by Roland Schneider, is in the works for next month's "Hands-on" heading. We're still looking for articles under the "Industry" and "Technology" headings.