EurOpen

News Letter

Number

Spring 1992

Volume 12

EurOpen: The European Forum for Open Systems

Volume 12, Number 1, Spring 1992

ISSN 1011-4211

Editorial Team

Contents

Editorial	5
Against User Interface Copyright	6
Executive Report	15
OpenForum Conference and Exhibition	29
Strategy and Business Conference	29
Call for Participation	30
EurOpen Publications	31
Calendar of UNIX Events	34
USENIX News	36
Workshop on Micro-Kernels & Other Kernel Architecture	36
Pre-Announcement and Call for Papers	36
OpenForum Call for Papers	37
Preliminary Announcement and Call for Papers	38
USENIX Call for Papers	38
UKUUG Column	39
Report on the October 1991 IEEE POSIX Meeting	40
EWG News	43
i2u Annual Convention 1992	44
USLE Column	45
Book Review	48
Book Reviews	49
Abstracts	51
EurOpen and EUnet Addresses	54

Production Team

Gina Baikenycz
Typesetter

The Instruction Set
City House
190 City Road
London EC1V 2QH
United Kingdom

Telephone +44 71 253 5121

E-mail gina@hoskyns.co.uk

Joe Di Mascio
Printer

Rank Xerox Business Services Ltd
89-91 Hampstead Road
London
NW1 3EL
United Kingdom

Telephone +44 71 388 5607
Facsimile +44 71 388 8841

Jeanne de Bont
Designer

Sikkel 17
6026 DH
Maarheeze
The Netherlands

Telephone +31 77 59 3353
Facsimile +31 77 59 4050

E-mail jdbo@oce.nl

Publisher

EurOpen
Owles Hall
Buntingford
Hertfordshire SG9 9PL
United Kingdom

Telephone +44 763 73039
Facsimile +44 763 73255

E-mail europen@EU.net

Editorial

Alain D. D. Williams
Parliament Hill Computers Ltd
Caversham, United Kingdom

E-mail addw@phcomp.co.uk

Software Patents and Interface Copyright

Richard Stallman has long been well known (amongst other things) for his views on the general free availability of source to programs.

Some of you may have seen him on his whistle stop tour of Europe recently, he was campaigning again, but this time on two issues which are much more important than that of source copyright.

Did you know that the almost universal use of **exclusive or** to put pixels on a bit map screen is subject to a patent? If you wish to write software that uses this technique you probably ought to pay royalties.

The use of software patents as a competitive tool is relatively new, but is rapidly becoming much more prevalent. Richard argues that if this trend continues it will soon be difficult to write new programs at all - unless you are an employee of a large enough organisation which can cross licence itself out of this problem. It is not surprising that the main proponents for patents and copyright interface copyright are the large established companies.

European law is different from US law, but there is pressure on the European legislators to adopt US practice - something that they have so far only just resisted.Whatever your conclusion this is something that you should know about, so turn the page.

The Newsletter is Changing

To provide you with ever better value, the contents of the newsletter is changing. Although the results of survey of the last issue arrived too late to have much effect on this issue we have started. Please do keep on sending me your opinions on what you see - let me know what YOU want in it.

How successful are the National Groups?

Once again here is the national group membership. Last year, the table was sorted by members per million of population. There is little change in the ranking by that metric, so this year I have sorted by percentage growth over last year's membership.

The results are interesting, and reflect the countries where Open Systems are gaining substantial new ground.

Our two new members show infinite growth - my challenge to them this year is to maintain their position at the top of the table!

Country	Population (Millions)	Members	Members per Millions Population	Last Year's Membership	Percentage Change
Algeria	25	32	1.3	0	∞
Tunisia	7.5	6	0.8	0	∞
Czechoslovakia	15.7	36	2.3	10	260
Spain	50	70	1.4	26	169.2
Hungary	10.6	38	3.5	16	137.5
Yugoslavia	22	35	1.6	16	118.8
Switzerland	6	121	20.2	71	70.4
Ireland	3.5	57	16.2	35	62.9
Belgium	9.9	156	15.8	108	44.4
Norway	4	283	70.8	225	25.8
Iceland	0.25	40	160	32	25.8
Germany	79.1	415	5.2	348	19.3
Italy	60	232	3.9	19.6	18.4
United Kingdom	57.1	693	12.1	588	17.9
Netherlands	14	432	30.9	370	16.8
Sweden	8.5	530	62.4	456	16.2
Austria	7.6	144	18.9	129	11.6
Finland	4.9	253	51.6	235	7.7
Portugal	9.8	113	11.5	105	7.6
Denmark	5.1	410	80.4	386	6.2
Soviet Union	285.9	32	0.1	32	0
France	56	509	9	1110	-54

Against User Interface Copyright

(October 20, 1991)

The League for Programming Freedom

In June 1990, Lotus won a copyright infringement suit against Paperback Software, a small company that implemented a spreadsheet that obeys the same keystroke commands used in Lotus 1-2-3. Paperback was not accused of copying code from 1-2-3 – only of supporting compatible user commands. Such imitation was common practice until unexpected court decisions in recent years extended the scope of copyright law.

Within a week, Lotus went on to sue Borland over Quattro, a spreadsheet whose usual command language has only a few similarities to 1-2-3. Lotus claims that these similarities in keystroke sequences and/or the ability to customize the interface to emulate 1-2-3 are enough to infringe.

More ominously, Apple Computer has sued Microsoft and Hewlett Packard for implementing a window system whose displays partially resemble those of the Macintosh system. Subsequently Xerox sued Apple for implementing the Macintosh system, which derives some general concepts from the earlier Xerox Star system. These suits try to broaden the Lotus decision and establish copyright on a large class of user interfaces. The Xerox lawsuit was dismissed because of a technicality; but if it had succeeded, it would probably have created an even broader monopoly than the Apple lawsuit may.

And Ashton-Tate has sued Fox Software for implementing a database program that accepts the same programming language used in dBase. This particular lawsuit was dropped by Borland, which bought Ashton-Tate in 1991, but the possibility of copyrighted programming languages remains. Adobe claims that the Postscript language is copyrighted, though it has not sued those who reject this claim. Wolfram Research claims that the language of Mathematica is copyrighted and has threatened to sue the University of California. If a programming language becomes copyrighted, the impact on users who have spent years writing programs in the language would be devastating.

While this paper addresses primarily the issue of copyright on specific user interfaces, most of the arguments apply with added force to any broader monopoly.

What Is a User Interface?

A user interface is what you have to learn to operate a machine; in other words, it is the language you use to communicate with the machine. The user interface of a typewriter is the layout of the keys. The user interface of a car includes a steering wheel for turning, pedals to speed up and slow down, a lever to signal turns, etc.

When the machine is a computer program, the interface includes that of the computer – its keyboard, screen and mouse – plus those aspects specific to the program. These typically include the commands, menus, programming languages, and the way data is presented on the screen.

A copyright on a user interface means a government-imposed monopoly on its use. In the example of the typewriter, this would mean that each manufacturer would be forced to arrange the keys in a different layout.

The Purpose of Copyright

In the United States, the Constitution says that the purpose of copyright is to "promote the progress of science and the useful arts." Conspicuously absent is any hint of intention to enrich copyright holders to the detriment of the users of copyrighted works.

The Supreme Court made the reason for this absence explicit, stating in *Fox Film vs. Doyal* that "The sole interest of the United States and the primary object in conferring the [copyright] monopoly lie in the general benefits derived by the public from the labours of authors."

In other words, since copyright is a government-imposed monopoly, which interferes with the freedom of the public in a significant way, it is justified only if the benefit to the public exceeds the cost to the public.

The spirit of individual freedom must, if anything, incline us against monopoly. Following either the Supreme Court or the principle of freedom, the fundamental question is: what value does user interface copyright offer the public – and what price would we have to pay for it?

Reason #1: More Incentive Is Not Needed

The developers of the Star, the Macintosh system, 1-2-3 and dBase claim that without interface copyright there would be insufficient incentive to develop such products. This is disproved by their own actions.

Until 1986, user interface copyright was unheard of. The computer industry developed under a system where imitating a user interface was both standard practice and lawful. Under this system, today's plaintiffs made their decisions to develop their products. When faced with the choice in actuality, they decided that they did, indeed, have "enough incentive".

Even though competitors were free to imitate these interfaces, this did not prevent most of the original products from being successful and producing a large return on the investment. In fact,

they were so successful that they became *de facto* standards. (The Xerox Star was a failure due to poor marketing even though nothing similar existed.)

Even if interface copyright would increase the existing incentive, additional improvements in user interfaces would not necessarily result. Once you suck a bottle dry, more suction won't get more out of it. The existing incentive is so great that it may well suffice to motivate everyone who has an idea worth developing. Extra incentive, at the public's expense, will only increase the price of these developments.

Reason #2: "Look and Feel" Will Not Protect Small Companies

The proponents of user interface copyright claim that it would protect small companies from being wiped out by large competitors. Yet look around: today's interface copyright plaintiffs are large, established companies. User interface copyright is crushing when the interface is an effective standard. However, a small company is vulnerable when its product is little used, and its interface is little known. In this situation, user interface copyright won't help the small company much.

Imagine a small company with 10,000 customers: a large company may believe there is a potential market of a million users, not reached by the small company, for a similar product. The large company will try to use its marketing might to reach them before the small company can.

User interface copyright won't change this outcome. Forcing the large company to develop an incompatible interface will have little effect on the majority of potential customers – those who have not learned the other interface. They will buy from the large company anyway.

What's more, interface copyright will work against the small company if the large company's product becomes an effective standard. Then new customers will have an additional reason to prefer the large company. To survive, the small company will need to offer compatibility with this standard – but, due to user interface copyright, it will not be allowed to do so.

Instead of relying upon monopolistic measures, small companies are most successful when they rely on their own inherent advantages: agility, low overhead, and willingness to take risks.

Reason #3: Diversity in Interfaces Is Not Desirable

The copyright system was designed to encourage diversity; its details work toward this end. Diversity is the primary goal when it comes to novels, songs, and the other traditional domains of copyright. Readers want to read novels they have not yet read.

But diversity is not the goal of interface design. Users of any kind of machinery want consistency in interfaces because this promotes ease of use. Thus, by standardizing symbols on automobile dashboards, we have made it possible for any licensed driver to operate any car without additional instruction. Incompatibility in interfaces is a price to be paid when worthwhile, not a benefit.

Significantly better interfaces may be hard to think of, but it is easy to invent interfaces which are merely different. Interface copyright will surely succeed in encouraging this sort of "interface development". The result will be gratuitous incompatibility.

Reason #4: Meaningful Competition Is Reduced

Under the regime of interface copyright, there will be no compatible competition for established products. For a user to switch to a different brand will require retraining.

But users don't like to retrain, not even for a significant improvement. For example, the Dvorak keyboard layout, invented several decades ago, enables a typist to type faster and more accurately than is possible with the standard "QWERTY" layout. Nonetheless, few people use it. Even new typists don't learn Dvorak, because they want to learn the layout used on most typewriters.

Alternative products that require such an effort by the consumer are not effective competition. The monopoly on the established interface will yield in practice a monopoly on the functionality accessed by it. This will cause higher prices and less technological advancement – a windfall for lucky businesses, but bad for the public at large.

Reason #5: Incompatibility Does Not Go Away

If there had been a 50-year interface copyright for the steering wheel, it would have expired not long ago. During the span of the copyright, we would have got cars steered with joysticks, cars steered with levers, and cars steered with pedals. Each car user would have had to choose a brand of car to learn to drive, and it would not be easy to switch.

The expiration of the copyright would have freed manufacturers to switch to the best of the known interfaces. But if Ford cars were steered with wheels and General Motors were steered with pedals, neither company could change interface without abandoning their old customers. It would take decades to converge on a single interface.

Reason #6: Users Invest More Than Developers

The plaintiffs like to claim that user interfaces represent large investments on their part.

In fact, the effort spent designing the user interface of a computer program is usually small compared to the cost of developing the program itself. The people who make a large investment in the user interface are the users who train to use it. Users have spent much more time and money learning to use 1-2-3 than Lotus spent developing the entire program, let alone what Lotus spent develop the program's interface *per se*.

Thus, if investment justifies ownership, it is the users who should be the owners. The users should be allowed to decide – in the marketplace – who may use it. According to *Infoworld* (mid January 1989), computer users in general expect user interface copyright to be harmful.

Reason #7: Discrimination Against Software Sharing

User interface copyright discriminates against freely redistributable software, such as freeware, shareware and public domain software.

Although it *may* be possible to license an interface for a proprietary program, if the owner is willing, these licenses require payment, usually per copy. There is no way to collect this payment for a freely redistributable program. The result will be a growing body of interfaces that are barred to non-proprietary software.

Authors of these programs donate to the public the right to share them, and sometimes also to study and change their workings. This is a public service, and one less common than innovation. It does not make sense to encourage innovation of one sort with means that bar donation of another sort.

Reason #8: Copyright Will Be a Tool For Extortion

The scope of interface copyright is so vague and potentially wide that it will be difficult for any programmer to be sure of being safe from lawsuits. Most programs need an interface, and there is usually no way to design an interface except based on the ideas you have seen used elsewhere. Only a great genius would be likely to envision a usable interface without a deep resemblance to current practice. It follows that most programming projects will risk an interface infringement suit.

The spirit of "Millions for defense, but not a cent for tribute" is little honoured in business today. Customers and investors often avoid companies that are targets of suits; an eventual victory may come years too late to prevent great loss or even bankruptcy. Therefore, when offered a choice between paying royalties and being sued, most businesses pay, even if they would probably win a suit.

Since this tendency is well known, companies often take advantage of it by filing or threatening suits they are unlikely to win. As long as any interface copyright exists, this form of extortion will broaden its effective scope.

Reason #9: Useful Innovation Is Inhibited

Due to the evolutionary nature of interface development, interface copyright will actually retard progress.

Fully fleshed-out interfaces don't often arise as *tours de force* from the minds of isolated masters. They result from repeated implementations, by different groups, each learning from the results of previous attempts. For example, the Macintosh interface was based on ideas tried previously by Xerox and SRI, and before that by the Stanford Artificial Intelligence Laboratory. The Xerox Star also drew on the interface ideas that came from SRI and SAIL. 1-2-3 adapted the interface ideas of Visicalc and other spreadsheets. dBase drew on a program developed at the Jet Propulsion Laboratory.

This evolutionary process resembles the creation of folk art rather than the way symphonies, novels or films are made. The advances that we ought to encourage are most often small, localized changes to what someone else has done. If each interface has an owner, it will be difficult to implement such ideas. Even assuming the owner will license the interface that is to be improved, the inconvenience and expense would discourage all but the most determined.

Users often appreciate small, incremental changes that make programs easier or faster to use. This means changes that are upwards compatible, or affect only part of a well-known interface. Thus, on computer keyboards, we now have function keys, arrow keys, a delete key and a control key, which typewriters did not have. But the layout of the letters is unchanged.

However, such partial changes as this are not permitted by copyright law. If any significant portion of the new interface is the same as a copyrighted interface, the new interface is illegal.

Reason #10: Interface Developers Don't Want Interface Copyright

At the 1989 ACM Conference on Computer-Human Interaction, Professor Samuelson of the Emory School of Law presented a "mock trial" with legal arguments for and against user interface copyright, and then asked the attendees – researchers and developers of user interfaces – to fill out a survey of their opinion on the subject.

The respondents overwhelmingly opposed all aspects of user interface copyright, by as much as 4 to 1 for some aspects. When they were asked whether user interface copyright would harm or help the field, on a scale from 1 (harm) to 5 (help), the average answer was 1.6.[1]

The advocates of user interface copyright say that it would provide better security and income for user interface designers. However, the survey shows that these supposed beneficiaries would prefer to be let alone.

Do You Really Want a User Interface Copyright?

For a business, "locking in" customers may be profitable for a time. But, as the vendors of proprietary operating systems have found out, this generates resentment and eventually drives customers to try to escape. In the long run, this leads to failure.

Therefore, by permitting user interface copyright, society encourages counterproductive thinking in its businesses. Not all businesses can resist this temptation; let us not tempt them.

Conclusion

Monopolies on user interfaces do not serve the users and do not "promote the progress of science and the useful arts." User interfaces ought to be the common property of all, as they undisputedly were until a few years ago.

---

1 *See the May 1990 issue of the Communications of the ACM, for the full results.*

(February 28, 1991)

The League for Programming Freedom

Software patents threaten to devastate America's computer industry. Patents granted in the past decade are now being used to attack companies such as the Lotus Development Corporation for selling programs that they have independently developed. Soon new companies will often be barred from the software arena – most major programs will require licenses for dozens of patents, and this will make them infeasible. This problem has only one solution: software patents must be eliminated.

The Patent System and Computer Programs

The framers of the United States Constitution established the patent system so that inventors would have an incentive to share their inventions with the general public. In exchange for divulging an invention, the patent grants the inventor a 17 year monopoly on its use. The patent holder can license others to use the invention, but may also refuse to do so. Independent reinvention of the same technique by others does not give them the right to use it.

Patents do not cover specific systems: instead, they cover particular techniques that can be used to build systems, or particular features that systems can offer. Once a technique or feature is patented, it may not be used in a system without the permission of the patent-holder – even if it is implemented in a different way. Since a computer program typically uses many techniques and provides many features, it can infringe many patents at once.

Until recently, patents were not used in the software field. Software developers copyrighted individual programs or made them trade secrets. Copyright was traditionally understood to cover the implementation details of a particular program; it did not cover the features of the program, or the general methods used. And trade secrecy, by definition, could not prohibit any development work by someone who did not know the secret.

On this basis, software development was extremely profitable, and received considerable investment, without any prohibition on independent software development. But this scheme of things is no more. A change in U.S. government policy in the early 1980's stimulated a flood of applications. Now many have been approved, and the rate is accelerating.

Many programmers are unaware of the change and do not appreciate the magnitude of its effects. Today the lawsuits are just beginning.

Absurd Patents

The Patent Office and the courts have had a difficult time with computer software. The Patent Office refused until recently to hire Computer Science graduates as examiners, and in any case does not offer competitive salaries for the field. Patent examiners are often ill– prepared to evaluate software patent applications to determine if they represent techniques that are widely known or obvious – both of which are grounds for rejection.

Their task is made more difficult because many commonly-used software techniques do not appear in the scientific literature of computer science. Some seemed too obvious to publish while others seemed insufficiently general; some were open secrets.

Computer scientists know many techniques that can be generalized to widely varying circumstances. But the Patent Office seems to believe that each separate use of a technique is a candidate for a new patent. For example, Apple was sued because the Hypercard program allegedly violates patent number 4,736,308, a patent that covers displaying portions of two or more strings together on the screen – effectively, scrolling with multiple subwindows. Scrolling and subwindows are well-known techniques, but combining them is now apparently illegal.

The granting of a patent by the Patent Office carries a presumption in law that the patent is valid. Patents for well-known techniques that were in use many years before the patent application have been upheld by federal courts. It can be hard to prove a technique was well known at the time in question.

For example, the technique of using exclusive-or to write a cursor onto a screen is both well known and obvious. (Its advantage is that another identical exclusive-or operation can be used to erase the cursor without damaging the other data on the screen.) This technique can be implemented in a few lines of a program, and a clever high school student might well reinvent it. But it is covered by patent number 4,197,590, which has been upheld twice in court even though the technique was used at least five years before the patent application. Cadtrak, the company that owns this patent, collects millions of dollars from large computer manufacturers.

English patents covering customary graphics techniques, including airbrushing, stencilling, and combination of two images under control of a third one, were recently upheld in court, despite the testimony of the pioneers of the field that they had developed these techniques years before. (The corresponding United States patents, including 4,633,416 and 4,602,286, have not yet been tested in court, but they probably will be soon.)

All the major developers of spreadsheet programs have been threatened on the basis of patent 4,398,249, covering "natural order recalc" – the recalculation of all the spreadsheet entries that are affected by the changes the user makes, rather than recalculation in a fixed order. Currently Lotus alone is being sued, but a victory for the plaintiff in this case would leave the other developers little hope. The League has found prior art that may defeat this patent, but this is not assured.

Nothing protects programmers from accidentally using a technique that is patented, and then being sued for it. Taking an existing program and making it run faster may also make it violate half a dozen patents that have been granted, or are about to be granted.

Even if the Patent Office learns to understand software better, the mistakes it is making now will follow us into the next century, unless Congress or the Supreme Court intervenes to declare these patents void.

However, this is not the whole of the problem. Computer programming is fundamentally different from the other fields that the patent system previously covered. Even if the patent system were to operate "as intended" for software, it would still obstruct the industry it is supposed to promote.

What Is "Obvious"?

The patent system will not grant or uphold patents that are judged to be obvious. However, the system interprets the word "obvious" in a way that might surprise computer programmers. The standard of obviousness developed in other fields is inappropriate for software.

Patent examiners and judges are accustomed to considering even small, incremental changes as deserving new patents. For example, the famous *Polaroid vs. Kodak* case hinged on differences in the number and order of layers of chemicals in a film – differences between the technique Kodak was using and those described by previous, expired patents. The court ruled that these differences were unobvious.

Computer scientists solve problems quickly because the medium of programming is tractable. They are trained to generalize solution principles from one problem to another. One such generalization is that a procedure can be repeated or subdivided. Programmers consider this obvious – but the Patent Office did not think that it was obvious when it granted the patent on scrolling multiple strings, described above.

Cases such as this cannot be considered errors. The patent system is functioning as it was designed to do – but with software, it produces outrageous results.

Patenting What Is Too Obvious to Publish

Sometimes it is possible to patent a technique that is not new precisely because it is obvious – so obvious that no one would have published a paper about it.

For example, computer companies distributing the free X Window System developed by MIT are now being threatened with lawsuits by AT&T over patent number 4,555,775, covering the use of "backing store" in a window system that lets multiple programs have windows. Backing store means that the contents of a window that is temporarily partly hidden are saved in off-screen memory, so they can be restored quickly if the obscuring window disappears.

Early window systems were developed on computers that could not run two programs at once. These computers had small memories, so saving window contents was obviously a waste of scarce memory space. Later, larger multiprocessing computers led to the use of backing store, and to permitting each program to have its own windows. The combination was inevitable.

The technique of backing store was used at MIT in the Lisp Machine System before AT&T applied for a patent. (By coincidence, the Lisp Machine also supported multiprocessing.) The Lisp Machine developers published nothing about backing store at the time, considering it too obvious. It was mentioned when a programmers' manual explained how to turn it on and off.

But this manual was published one week after the AT&T patent application – too late to count as prior art to defeat the patent. So the AT&T patent may stand, and MIT may be forbidden to continue using a method that MIT used before AT&T.

The result is that the dozens of companies and hundreds of thousands of users who accepted the software from MIT on the understanding that it was free are now faced with possible lawsuits. (They are also being threatened with Cadtrak's exclusive-or patent.) The X Window System project was intended to develop a window system that all developers could use freely. This public service goal seems to have been thwarted by patents.

Why Software Is Different

Software systems are much easier to design than hardware systems of the same number of components. For example, a program of 100,000 components might be 50,000 lines long and could be written by two good programmers in a year. The equipment needed for this costs less than $10,000; the only other cost would be the programmers' own living expenses while doing the job. The total investment would be less than a $100,000. If done commercially in a large company, it might cost twice that. By contrast, an automobile typically contains under 100,000 components; it requires a large team and costs tens of millions of dollars to design.

And software is also much cheaper to manufacture: copies can be made easily on an ordinary workstation costing under ten thousand dollars. To produce a complex hardware system often requires a factory costing tens of millions of dollars.

Why is this? A hardware system has to be designed using real components. They have varying costs; they have limits of operation; they may be sensitive to temperature, vibration or humidity; they may generate noise; they drain power; they may fail either momentarily or permanently. They must be physically assembled in their proper places, and they must be accessible for replacement in case they fail.

Moreover, each of the components in a hardware design is likely to affect the behaviour of many others. This greatly complicates the task of determining what a hardware design will do: mathematical modelling may prove wrong when the design is built.

By contrast, a computer program is built out of ideal mathematical objects whose behaviour is defined, not modelled approximately, by abstract rules. When an if-statement follows a while-statement,

there is no need to study whether the if-statement will draw power from the while-statement and thereby distort its output, nor whether it could overstress the while-statement and make it fail.

Despite being built from simple parts, computer programs are incredibly complex. The program with 100,000 parts is as complex as an automobile, though far easier to design.

While programs cost substantially less to write, market and sell than automobiles, the cost of dealing with the patent system will not be less. The same number of components will, on the average, involve the same number techniques that might be patented.

The Danger of a Lawsuit

Under the current patent system, a software developer who wishes to follow the law must determine which patents a program violates and negotiate with each patent holder a license to use that patent. Licensing may be prohibitively expensive, or even unavailable if the patent is held by a competitor. Even "reasonable" license fees for several patents can add up to make a project infeasible. Alternatively, the developer may wish to avoid using the patent altogether; but there may be no way around it.

The worst danger of the patent system is that a developer might find, after releasing a product, that it infringes one or many patents. The resulting lawsuit and legal fees could force even a medium-size company out of business.

Worst of all, there is no practical way for a software developer to avoid this danger – there is no effective way to find out what patents a system will infringe. There is a way to try to find out – a patent search – but searches are unreliable and in any case too expensive to use for software projects.

Patent Searches Are Prohibitively Expensive

A system with a hundred thousand components can use hundreds of techniques that might already be patented. Since each patent search costs thousands of dollars, searching for all the possible points of danger could easily cost over a million. This is far more than the cost of writing the program.

The costs don't stop there. Patent applications are written by lawyers for lawyers. A programmer reading a patent may not believe that his program violates the patent, but a federal court may rule otherwise. It is thus now necessary to involve patent attorneys at every phase of program development.

Yet this only reduces the risk of being sued later – it does not eliminate the risk. So it is necessary to have a reserve of cash for the eventuality of a lawsuit.

When a company spends millions to design a hardware system, and plans to invest tens of millions to manufacture it, an extra million or two to pay for dealing with the patent system might be bearable. However, for the inexpensive programming project, the same extra cost is prohibitive. Individuals and small companies especially cannot afford these costs. Software patents will put an end to software entrepreneurs.

Patent Searches Are Unreliable

Even if developers could afford patent searches, these are not a reliable method of avoiding the use of patented techniques. This is because patent searches do not reveal pending patent applications (which are kept confidential by the Patent Office). Since it takes several years on the average for a software patent to be granted, this is a serious problem: a developer could begin designing a large program after a patent has been applied for, and release the program before the patent is approved. Only later will the developer learn that distribution of the program is prohibited.

For example, the implementors of the widely-used public domain data compression program **`compress`** followed an algorithm obtained from the journal *IEEE Computer*. (This algorithm is also used in several popular programs for microcomputers, including **`PKZIP`**.) They and the user community were surprised to learn later that patent number 4,558,302 had been issued to one of the authors of the article. Now Unisys is demanding royalties for using this algorithm. Although the program **`compress`** is still in the public domain, using it means risking a lawsuit.

The Patent Office does not have a workable scheme for classifying software patents. Patents are most frequently classified by end results, such as "converting iron to steel;" but many patents cover algorithms whose use in a program is entirely independent of the purpose of the program. For example, a program to analyse human speech might infringe the patent on a speed-up in the Fast Fourier Transform; so might a program to perform symbolic algebra (in multiplying large numbers); but the category to search for such a patent would be hard to predict.

You might think it would be easy to keep a list of the patented software techniques, or even simply remember them. However, managing such a list is nearly impossible. A list compiled in 1989 by lawyers specializing in the field omitted some of the patents mentioned in this paper.

Obscure Patents

When you imagine an invention, you probably think of something that could be described in a few words, such as "a flying machine with fixed, curved wings" or "an electrical communicator with a microphone and a speaker". But most patents cover complex detailed processes that have no simple descriptions – often they are speed-ups or variants of well-known processes that are themselves complex.

Most of these patents are neither obvious nor brilliant; they are obscure. A capable software designer will "invent" several such improvements in the course of a project. However, there are many avenues for improving a technique, so no single project is likely to find any given one.

For example, IBM has several patents (including patent number 4,656,583) on workmanlike, albeit complex, speed-ups for well-known computations performed by optimising compilers, such as register colouring and computing the available expressions.

Patents are also granted on combinations of techniques that are already widely used. One example is IBM patent 4,742,450, which covers "shared copy-on-write segments." This technique allows several programs to share the same piece of memory that represents information in a file; if any program writes a page in the file, that page is replaced by a copy in all of the programs, which

continue to share that page with each other but no longer share with the file.

Shared segments and copy-on-write have been used since the 1960's; this particular combination may be new as a specific feature, but is hardly an invention. Nevertheless, the Patent Office thought that it merited a patent, which must now be taken into account by the developer of any new operating system.

Obscure patents are like land mines: other developers are more likely to reinvent these techniques than to find out about the patents, and then they will be sued. The chance of running into any one of these patents is small, but they are so numerous that you cannot go far without hitting one. Every basic technique has many variations, and a small set of basic techniques can be combined in many ways. The patent office has now granted at least 2000 software patents – no less than 700 in 1989 alone, according to a list compiled by EDS. We can expect the pace to accelerate. In ten years, programmers will have no choice but to march on blindly and hope they are lucky.

Patent Licensing Has Problems, Too

Most large software companies are trying to solve the problem of patents by getting patents of their own. Then they hope to cross-license with the other large companies that own most of the patents, so they will be free to go on as before.

While this approach will allow companies like Microsoft, Apple and IBM to continue in business, it will shut new companies out of the field. A future start-up, with no patents of its own, will be forced to pay whatever price the giants choose to impose. That price might be high: established companies have an interest in excluding future competitors. The recent Lotus lawsuits against Borland and the Santa Cruz Operation (although involving an extended idea of copyright rather than patents) show how this can work.

Even the giants cannot protect themselves with cross-licensing from companies whose only business is to obtain exclusive rights to patents and then threaten to sue. For example, consider the New York-based Refac Technology Development Corporation, representing the owner of the "natural order recalc" patent. Contrary to its name, Refac does not develop anything except lawsuits – it has no business reason to join a cross-licensing compact. Cadtrak, the owner of the exclusive-or patent, is also a litigation company.

Refac is demanding five percent of sales of all major spread-sheet programs. If a future program infringes on twenty such patents – and this is not unlikely, given the complexity of computer programs and the broad applicability of many patents – the combined royalties could exceed 100% of the sales price. (In practice, just a few patents can make a program unprofitable.)

The Fundamental Question

According to the Constitution of the United States, the purpose of patents is to "promote the progress of science and the useful arts." Thus, the basic question at issue is whether software patents, supposedly a method of encouraging software progress, will truly do so, or will retard progress instead.

So far we have explained the ways in which patents will make ordinary software development difficult. But what of the intended benefits of patents: more invention, and more public disclosure of inventions? To what extent will these actually occur in the field of software?

There will be little benefit to society from software patents because invention in software was already flourishing before software patents, and inventions were normally published in journals for everyone to use. Invention flourished so strongly, in fact, that the same inventions were often found again and again.

In Software, Independent Reinvention Is Commonplace

A patent is an absolute monopoly; everyone is forbidden to use the patented process, even those who reinvent it independently. This policy implicitly assumes that inventions are rare and precious, since only in those circumstances is it beneficial.

The field of software is one of constant reinvention; as some people say, programmers throw away more "inventions" each week than other people develop in a year. And the comparative ease of designing large software systems makes it easy for many people to do work in the field. A programmer solves many problems in developing each program. These solutions are likely to be reinvented frequently as other programmers tackle similar problems.

The prevalence of independent reinvention negates the usual purpose of patents. Patents are intended to encourage inventions and, above all, the disclosure of inventions. If a technique will be reinvented frequently, there is no need to encourage more people to invent it; since some of the developers will choose to publish it (if publication is merited), there is no point in encouraging a particular inventor to publish it – not at the cost of inhibiting use of the technique.

Overemphasis of Inventions

Many analysts of American and Japanese industry have attributed Japanese success at producing quality products to the fact that they emphasize incremental improvements, convenient features and quality rather than noteworthy inventions.

It is especially true in software that success depends primarily on getting the details right. And that is most of the work in developing any useful software system. Inventions are a comparatively unimportant part of the job.

The idea of software patents is thus an example of the mistaken American preoccupation with inventions rather than products. And patents will encourage this mistaken focus, even as they impede the development work that actually produces better software.

Impeding Innovation

By reducing the number of programmers engaged in software development, software patents will actually impede innovation. Much software innovation comes from programmers solving problems while developing software, not from projects whose specific purpose is to make inventions and obtain patents. In other words, these innovations are by-products of software development.

When patents make development more difficult, and cut down on development projects, they will also cut down on the by-products of development – new techniques.

Could Patents Ever Be Beneficial?

Although software patents in general are harmful to society as a whole, we do not claim that every single software patent is necessarily harmful. Careful study might show that under certain specific and narrow conditions (necessarily excluding the vast majority of cases) it is beneficial to grant software patents.

Nonetheless, the right thing to do now is to eliminate all software patents as soon as possible, before more damage is done. The careful study can come afterward.

Clearly software patents are not urgently needed by anyone except patent lawyers. The pre-patent software industry had no problem that was solved by patents; there was no shortage of invention, and no shortage of investment.

Complete elimination of software patents may not be the ideal solution, but it is close, and is a great improvement. Its very simplicity helps avoid a long delay while people argue about details.

If it is ever shown that software patents are beneficial in certain exceptional cases, the law can be changed again at that time – if it is important enough. There is no reason to continue the present catastrophic situation until that day.

Software Patents Are Legally Questionable

It may come as a surprise that the extension of patent law to software is still legally questionable. It rests on an extreme interpretation of a particular 1981 Supreme Court decision, *Diamond vs. Deihr.*[1]

Traditionally, the only kinds of processes that could be patented were those for transforming matter (such as, for transforming iron into steel). Many other activities which we would consider processes were entirely excluded from patents, including business methods, data analysis, and "mental steps." This was called the "subject matter" doctrine. *Diamond vs. Deihr* has been interpreted by the Patent Office as a reversal of this doctrine, but the court did not explicitly reject it. The case concerned a process for curing rubber – a transformation of matter. The issue at hand was whether the use of a computer program in the process was enough to render it unpatentable, and the court ruled that it was not. The Patent Office took this narrow decision as a green light for unlimited patenting of software techniques, and even for the use of software to perform specific well-known and customary activities.

Most patent lawyers have embraced the change, saying that the new boundaries of patents should be defined over decades by a series of expensive court cases. Such a course of action will certainly be good for patent lawyers, but it is unlikely to be good for software developers and users.

1 *See "Legally Speaking" in Communications of the ACM}, August 1990.*

One Way to Eliminate Software Patents

We recommend the passage of a law to exclude software from the domain of patents. That is to say that, no matter what patents might exist, they would not cover implementations in software; only implementations in the form of hard-to-design hardware would be covered. An advantage of this method is that it would not be necessary to classify patent applications into hardware and software when examining them.

Many have asked how to define software for this purpose – where the line should be drawn. For the purpose of this legislation, software should be defined by the characteristics that make software patents especially harmful:

- Software is built from ideal infallible mathematical components, whose outputs are not affected by the components they feed into.

- Ideal mathematical components are defined by abstract rules, so that failure of a component is by definition impossible. The behaviour of any system built of these components is likewise defined by the consequences of applying the rules step by step to the components.

Software can be Easily and Cheaply Copied

Following this criterion, a program to compute prime numbers is a piece of software. A mechanical device designed specifically to perform the same computation is not software, since mechanical components have friction, can interfere with each other's motion, can fail, and must be assembled physically to form a working machine.

Any piece of software needs a hardware platform in order to run. The software operates the features of the hardware in some combination, under a plan. Our proposal is that combining the features in this way can never create infringement. If the hardware alone does not infringe a patent, then using it in a particular fashion under control of a program should not infringe either. In effect, a program is an extension of the programmer's mind, acting as a proxy for the programmer to control the hardware.

Usually the hardware is a general purpose computer, which implies no particular application. Such hardware cannot infringe any patents except those covering the construction of computers. Our proposal means that, when a user runs such a program on a general purpose computer, no patents other than those should apply.

The traditional distinction between hardware and software involves a complex of characteristics that used to go hand in hand. Some newer technologies, such as gate arrays and silicon compilers, blur the distinction because they combine characteristics associated with hardware with others associated with software. However, most of these technologies can be classified unambiguously for patent purposes, either as software or as hardware, using the criteria above. A few gray areas may remain, but these are comparatively small, and need not be an obstacle to solving the problems patents pose for ordinary software development. They will eventually be treated as hardware, as software, or as something in between.

What You Can Do

One way to help eliminate software patents is to join the League for Programming Freedom. See the end of this article.

Fighting Patents One by One

Until we succeed in eliminating all patenting of software, we must try to overturn individual software patents. This is very expensive and can solve only a small part of the problem, but that is better than nothing.

Overturning patents in court requires prior art, which may not be easy to find. The League for Programming Freedom will try to serve as a clearing house for this information, to assist the defendants in software patent suits. This depends on your help. If you know about prior art for any software patent, please send the information to the League at the address given above.

If you work on software, you can personally help prevent software patents by refusing to cooperate in applying for them. The details of this may depend on the situation.

Conclusion

Exempting software from the scope of patents will protect software developers from the insupportable cost of patent searches, the wasteful struggle to find a way clear of known patents, and the unavoidable danger of lawsuits.

If nothing is changed, what is now an efficient creative activity will become prohibitively expensive. To picture the effects, imagine if each square of pavement on the sidewalk had an owner, and pedestrians required a license to step on it. Imagine the negotiations necessary to walk an entire block under this system. That is what writing a program will be like if software patents continue. The sparks of creativity and individualism that have driven the computer revolution will be snuffed out.

What You Can Do

Don't do business as usual with the plaintiffs, Xerox, Lotus, and Apple. Buy from their competitors instead; sell their stock; develop new software for other computer systems rather than theirs, and port existing applications away from their systems.

- Don't work for the "look and feel" plaintiffs or accept contracts from them.
- Give copies of this paper to your friends, colleagues and customers.
- Join the League for Programming Freedom. The League is a grass-roots organization of programmers and users opposing software patents and interface copyrights. (The League is not opposed to copyright on individual programs.) Annual dues for individual members are $42 for employed professionals, $10.50 for students, and $21 for others. We appreciate activists, but members who cannot contribute their time are also welcome.

To contact the League, telephone +1 617 243 4091, send Internet mail to the address league@prep.ai.mit.edu, or write to:

League for Programming Freedom
1 Kendall Square #143
PO Box 9171
Cambridge, MA 02139
USA

European addresses are:

Alan Bundy
80 South Bridge, Edinburgh, EH1 1HN, UK
Telephone +44 31 650 2716
E-mail bundy@ed.ac.uk

Rog Nagler
Olsen & Associates
Seefeldstrasse 223, CH-8008, Zurich, Switzerland
Telephone +411 386 4848
Facsimile +411 552282
E-mail nagler@olsen.ch

Also:

E-mail alo@hut.fi
E-mail gerben@rug.nl
E-mail felix@escape.vsse.in-berline.de
E-mail meh@gollum.uio.no

Join the EurOpen Working Group on Copyright Legislation, for contact points see the end of this newsletter.

You may also contact the legislators in your own country. The situation is not quite as bad in Europe as it is in the USA. Recommendations have been put before the European Council of Ministers who, after much deliberation, have not come to a conclusion.

There is still plenty of opportunity for *you* to do something. Write to your member of parliament, point out that this will damage European interests more than US ones.

The European Community has adopted a directive whose most natural interpretation imposes copyright on all kinds of interfaces, even on programming languages. Since the other countries of Europe are considering joining the EC, they also are in danger of being covered by the directive. In July 1990 the European Parliament passed a series of amendments supporting more open borrowing of elements from computer software.

Other, benign interpretations of the directive are also possible, but they are unlikely to be chosen by judges unless the governments of the individual EC countries explicitly mandate them. Convincing the governments requires political pressure from the programmers and users of Europe.

Lobbyists working on this issue say that most legislators are unfamiliar with computers and do not understand how harmful interface copyright could be. Thus, what programmers need to do is to educate their legislators.

One idea is to start teaching your representative the basics of using 1-2-3. Once the representative sees how much work is involved in learning to use a command language, explain that you have only taught one tenth of the subject. This should drive the point home.

Executive Report

Helen M W Gibbons
European Forum for Open Systems (EurOpen)
Buntingford
Hertfordshire
United Kingdom

E-mail europen@EU.net

Helen Gibbons is the business manager of EurOpen and is contactable at the EurOpen secretariat.

The Executive Committee

The Executive Committee, which at present consists of Michel Gien (Chairman), Frances Brazier (Vice Chair), Nigel Martin (Treasurer), Ernst Janich, Norman Hull and Kim Biel-Nielsen, with Teus Hagen as ex-chairman, held its last meeting for 1991 on 15 November and was due to hold its first full meeting for 1992 on 9 February 1992.

At the Governing Board meeting which will be held on 4-5 April 1992 the members of the Executive Board will stand down in accordance with the By-Laws and voting will take place to fill the 5 agreed seats. Nominations for the new Executive Committee *must* be made through a National Group and stocks of nomination forms were sent out in early December 1991 to each National Group Secretariat. The retiring committee only are welcome to put their names forward independently. the final date for receipt of nominations was 20 February 1992.

Appointment of Executive Director

As announced in the last newsletter Pierre Scheuer has jointed the EurOpen team as Executive Director. Since he only started his full time duties with EurOpen on 1 November 1991, Pierre has not had a great deal of time for action, but has nonetheless been extremely busy attending meetings with UniForum and helping to set up "OpenForum" in November. He has also taken over responsibility for the budgets and has spent some time at Owles Hall familiarising himself with the financial history and getting to know the Secretariat team.

Mr Scheuer sees EurOpen publications as an area of key concern. He intends to make this his first priority, and is discussing with Alain Williams possible improvements to the Newsletter. His second priority is to make a success of OpenForum. He will also be looking for ways to achieving a good understanding and close liaison with the National Groups.

Conferences for 1992

The next major event will be a Workshop held in co-operation with USENIX in Jersey, Channel Islands, on 6-9 April 1992. Tutorials will be held on Monday 6 April and in the morning of Tuesday 7 April. Tutorial topics include:

- GCC – the GNU C Compiler (Richard Stallman)
- Designing and Maintaining an Internetwork Gateway (Tina Darmohray)
- UNIX Programming Tools (Kenneth Ingham)
- Portability for C and UNIX (Steven Pemberton)
- Widgets – Introduction to Programming with the X Toolkit (Berry Kercheval)

The Conference Workshop will be held from 14.00 on Tuesday 7th and will finish in the afternoon of Thursday 9 April.

Conference Keynote Speakers:

- Mike Banahan (The Instruction Set – UK)
- Jean Hyenne (AFNOR – France)
- Andrew Hume (Bell Labs – USA)

Other distinguished speakers include:

- Donald Lewine (Data General Corp – USA)
- Axel Schreiner (University of Osnabrueck – Germany)
- Barry Shein (Software Tool & Die – USA)

Information on this event was sent to all members during 1991 and full details appear elsewhere in this newsletter.

The second major event of the year will be OpenForum '92. this event, jointly sponsored by EurOpen and UniForum will be held in Utrecht, Holland, on 25-27 November 1992. The event will comprise a major professionally organised Exhibition, Tutorials and World Class Industry Conference at which keynote talks by international authorities on open systems will be supported by interactive plenary sessions featuring international speakers from industry, the user community and the world of research and development. Panel sessions in parallel streams will focus on strategic and business issues as well as open systems technology.

The EurOpen Secretariat has already sent out an advertising brochure and Call for Papers to all members. Details of this event also appears in this Newsletter.

Conference Proceedings

The Proceedings of the Technical Conference held in Budapest, Hungary on 16-20 September 1991 has now been published by EurOpen and the European Forum for Open Systems and is available at a cost of 25 GBP or 35 ECU (members) and 65 GBP or 95 ECU (non-members) – all prices are inclusive of postage and packing (payment with order).

The 322 page book contains 27 papers covering Distributed Applications, Standards – Internationalisation, Intelligent Systems, Object Management, Security and User-Interfaces; design tools and includes papers by Mike Griffin and David Tilbrook.

The Autumn 1991 Conference in Budapest was the first EurOpen event to be held in Eastern Europe.

Subscriptions and Group Contributions

The subscriptions for direct members for 1992 has been raised to 65 ECU for individuals and 100 ECU for institutional members. Direct membership is open to those who would like to join EurOpen but in whose country no National Group exists at present.

Normally membership of EurOpen is effected only through joining a National Group and a full list of National Groups in membership appears at the end of the newsletter. The Group contribution for 1992 is 65 ECU per member. All National Groups were invoice on their current numbers at the beginning of January, and the new By-Laws, which are available from the EurOpen Secretariat, give details of payment schedules.

Newletter Questionnaires

Thank you to everyone who completed and returned the questionnaire included in the last newsletter.

The winners of the draw are:

A Macpherson	UK
A J Mason	UK
R Foulkes	UK
T Biedassek	Germany
S Nieuwenhuijs	The Netherlands
J Pierce	Ireland
R Smakman	The Netherlands
Daan Josephus Jitta	The Netherlands
Edward L De Wild	the Netherlands
Charles Sharp	UK
Steven Dietch	Sweden
Jonas Heyman	Sweden
Sean Byrne	Ireland
Urs Emmengger	Switzerland
Dick Mihalovsky	Finland

They are invited to be EurOpen's guest at dinner one evening during the Jersey Conference.

EurOpen Secretariat

Finally, the members of the EurOpen team at Owles Hall would like to thank everyone who sent Christmas cards and would like to wish all members of EurOpen a very happy and prosperous 1992.

The Secretariat is pleased to announce that it is now fully computerised and can be contacted on:

E-mail europen@EU.net
E-mail helen@europen.uucp
E-mail bill@europen.uucp

EurOpen & USENIX Spring 1992 Workshop/Conference

Jersey, Channel Islands 6th-9th
April 1992

The Spring 1992 event will be held in conjunction with the USENIX Association. This joint venture to be held on the island of Jersey will be the first event of this kind for EurOpen and USENIX.

One of the major goals of Open Systems is to support the distribution and mobility of data, users and processing. The purpose of the Spring 1992 EurOpen and USENIX Workshop is to identify the challenges of this goal and to discuss possible solutions. The workshop will address issues of concern to the system designer, developer, manager, administrator and end-user.

EurOpen (The European Forum for Open Systems) is an international federation of UNIX and Open Systems Users Groups. EurOpen is dedicated to

- promote and advance the knowledge, use and application of open computer systems using compatibility technique pioneered by the UNIX operating system.
- facilitate the exchange of information and views on the use and development of open systems.
- inform public opinion upon the subject of open systems.
- provide a focus for the standardisation of techniques used in open systems.
- encourage internationalisation of open systems, for the benefit of users of open systems.

EurOpen now has nearly 6,000 members in 26 groups throughout Europe and North Africa. EurOpen is a non-profit making organisation.

USENIX, the UNIX and Advanced Computing Systems professional and technical organisation, is a not-for-profit association dedicated to

- fostering innovation and communicating research and technological developments,
- sharing ideas and experiences, relevant to UNIX-related and advanced computing systems,
- providing a forum for the exercise of critical thought and airing of technical issues.

Founded in 1975, the Association sponsors two annual technical conferences and frequent symposia and workshops addressing special interest topics, such as C++, Mach, systems administration, and security. USENIX+1 publishes proceedings of its meetings, a bi-monthly newsletter ;login:, a refereed technical quarterly, Computing Systems, and is expanding its publishing role with a book series on advanced computing systems. The Association also actively participates in and reports on the activities of various ANSI, IEEE and ISO standards efforts.

Tutorials for this joint event will be held on Monday 6th and in the morning of Tuesday, 7th April. The Workshop/Conference will be held from 14.00 on Tuesday 7th and will finish in the afternoon of Thursday 9th April.

The venue for the Workshop/Conference is:

Hotel de France
St. Saviours Road
St. Helier
Jersey
Channel Islands

Telephone +44 534 38990
Facsimile +44 534 35354

Jersey is the most southerly of the British Isles and lies in the English Channel approximately 160 kilometres south of the English coast and 22 kilometres from the west of France.

Although a small island, 14x8 kilometres, it is very picturesque, renowned for its duty free prices and tourist industry as well as its offshore financial centre and its horticultural and agricultural produce.

Conference and Tutorial Enquiries
Helen Gibbons
EurOpen Secretariat
Owles Hall
Buntingford
Hertfordshire SG9 9PL
United Kingdom

Telephone +44 763 73039
Facsimile +44 763 73255

E-mail: europen@EU.net

Programme of Events Tutorials

Monday
6 April 1992, Hotel de France

Registration from 08.00 in the hotel foyer
Tutorials start at 09.30

Tutorial M1 – GCC – The GNU C Compiler

Tutor Richard Stallman

Intended Audience

This tutorial is an introduction to the internals of a mostly portable, highly optimizing C compiler. Attendees should have a working knowledge of C, be familiar with at least one machine architecture and assembly language, and have at least a general idea of how optimizing compilers work (such as from reading the dragon book by Aho, Sethi and Ullman).

Subject Matter

This tutorial will explain the major interfaces and algorithms used in GCC and what its major components do. The interfaces include the tree structures nre language front-ends must generate, the RTL data structure used by the optimizers, and the machine description files that must be written in order to port GCC to a new architecture.

This tutorial will also give you an in-depth explanation of how a particular optimizing compiler works: what components it has and what they do. Armed with this information, you should be able to read the source code for a particular component of GCC and understand it in detail. As GCC source is widely available and freely redistributable, it will be easy for you to continue to learn from and use the source after you leave the tutorial.

Biography

Richard M Stallman graduated magna cum laude with an MA in physics from Harvard University in 1974. He was employed from 1971 to 1984 as system developer at the MIT Artificial Laboratory, where his best known work was the creation of the original Emacs editor. He founded the GNU Project in 1984, and has since created GNU Emacs, the GNU debugger, GCC, and a technical community to continue their development. In 1990 he was the recipient of a MacArthur Fellowship and the ACM Grace Hopper award.

Tutorial M2 – Designing and Maintaining an Inter-network Gateway

Tutor Tina Darmohray

Most sites have a need to communicate electronically beyond their own borders. Typically users depend heavily on this capability. Unfortunately the pieces and tools involved in building a gateway are a not-so-easily acquired system administration forte. In this tutorial we will explore the issues in the design of an inter-network connection and present a firewall network model. We will cover sendmail, the domain name system, the hardware needed, and give sample router, DNS, and sendmail.cf files.

Inter-network Gateways

Nomenclature
Design Considerations and Tradeoffs

Implementing the Gateway

The Router
The Gateway
Gateway Security
Domain Name Server
Electronic Mail
Proxy FTP/Telnet

Who Should Attend

This course is valuable for systems administrators, design and development engineers, programmers, technical and operational managers, and all interested professionals involved in designing, configuring, implementing, or maintaining computer networks, and inter-network gateways or communication. Previous experience with TCP/IP networks will be helpful.

Monday
6 April 1992, Hotel de France

Registration from 08.00 in the hotel foyer
Tutorials start at 09.30

Tutorial M3 – Object-oriented Programming with ANSI C

Tutor Axel T Schreiner, University of Osnabrück, Germany

Prerequisites

You have to know C before you can use it in an object-oriented style. The techniques described in the tutorial make extensive use of ANSI C facilities, but they can be adapted to Classic C. A working knowledge of ANSI C helps, but if necessary, I will point out the differences where necessary.

Goals and Content

Have you heard about object-oriented programming? Are you confused by classes, objects, instances, methods, inheritance, dynamic binding, polymorphisms and all that? Does OOP really do everything but write the manual by itself? You would like to try it, but just haven't got the time to wade through the C++ manual?

Well, this is the tutorial for you, even if it may destroy your illusions. We will look at all the object-oriented terminology and make it work just using plain ANSI C. In this way you will not only be able to talk about the concepts, but you see how they are implemented, you can use them, and you can estimate, how expensive and how useful OOP is.

In the afternoon we will look at a few larger examples: a calculator for plain terminals, curses, and X; an accounting program, and possibly a mask-based data entry system. For the examples we will use a simple awk-based preprocessor, which saves some typing and gives the technique the appearance of a "real" object-oriented dialect of ANSI C.

If a participant tells me how to get it to him or her, I will make the sources to the examples and to the preprocessor available on a DOS floppy or by e-mail.

Biography

I have used UNIX and C for more than a decade, and I have written, translated, and edited a number of books in this area in English and German. I have programmed with C++ for about a year, and I have looked at Objective C (on a NeXT), SmallTalk, and Eiffel. I still like ANSI C better, and I have developed and used the technique shown in this tutorial to implement a couple languages and for a few courses and seminars. It works!

Only EurOpen National Group members and members of USENIX are permitted to attend tutorials.

M1, M2 and M3 will start at 09.30 and finish at approximately 17.30.

T1 and T2 will start at 09.30 and finish at 13.00. Lunch will be provided for all tutorial delegates.

Tuesday
7 April 1992, Hotel de France

Registration from 08.00 in the hotel foyer
Tutorials start at 09.30 and finish at 13.00

Tutorial T1 (half day) – UNIX Programming Tools

Tutor Kenneth Ingham

Intended Audience

This tutorial is aimed at programmers who know basic Unix topics and are familiar with the C programming language (All examples will be in C). It is designed to teach the concepts and uses of some of Unix's tools for application development.

The Unix operating system comes with a variety of tools which help the programmer produce cleaner, more accurate code with less work. The knowledge of when and why to use these tools is not covered in the manuals so users must learn this for themselves, often by trial and error. Additionally, while the manuals are good references, they are often not the best for learning the fundamentals of the Unix programming tools.

This tutorial will cover the following Unix tools:

- System building tools: make and imake
- Debuggers: dbx (and an overview of adb)
- Version control tools: (rcs and sccs)

This tutorial will teach not only how to use these tools, but also when and why they are useful (and when they might not be useful). The tutorial employs detailed examples to illustrate the tools being covered.

Biography

Kenneth Ingham has worked with UNIX for eleven years, spending the last six years as a systems programmer at the University of New Mexico. He has taught classes for the University, for UniForum, for Usenix, and for companies in New Mexico. He is the author of the book UNIX Tool Building (Academic Press). Currently Kenneth is a freelance consultant, teaching classes on UNIX and doing other interesting projects.

Tutorial T2 (half day) – Portability for C and Unix

Tutor Steven Pemberton

Open Systems

Suddenly everyone is talking open systems. Heterogeneous systems, divergent architectures, different suppliers, but a uniform operating environment for the user.

This has the benefit for the software supplier of widening the market, but all of a sudden programs are expected to run identically on a wide range of machines under ostensibly the same, but in different degrees varying, operating systems.

Software producers cannot afford to maintain different versions of their programs for every combination of architecture and operating environment, and this raises the challenge to the software writer to write code that is maximally portable over a wide range of systems and architectures.

C

The programming language C has risen to ascendency for this kind of work, and is probably the language that is available on the widest range of systems. However, despite this wide availability, it offers far from automatic portability, and is riddled with pitfalls and traps for the unwary. While it is perfectly possible to write portable programs in C, you have to be aware of the problem areas if you want to avoid the agonies of continually modifying source code each time a new machine and operating system combination comes along.

Unix

With the advent of open system awareness, the operating system Unix has come to the fore as a unique solution, as a system that is already available over a wide range of architectures and that is easy to port to new ones. After a long period of divergence, the various versions of Unix are now entering a period of convergence, and with the release of System V.4 and the publication of the POSIX standard, the differences are steadily diminishing.

However, there are still many areas where the conscientious software producer must take special care if programs are to remain portable over a maximal range of systems, especially if attention must also be paid to compatibility with older systems, and even systems like OS/2 and MS-DOS.

Tuesday
7 April 1992, Hotel de France

Registration from 08.00 in the hotel foyer
Tutorials start at 09.30 and finish at 13.00

Tutorial T2 (Continued)

The Tutorial

This half day tutorial is designed to give an overview of the minefield of C and Unix portability, and offer general solutions for maximising software portable over a wide range of architectures and platforms.

The topics covered include:

Philosophy: how to approach the problems of portability.

C: The language, the preprocessor, the ANSI standard and its relation to K&R, libraries available, variations you can expect over different compilers, variations over different platforms.

Unix: libraries available, variations you can expect, the role of the POSIX standard.

The shell and its environment: portable shell scripts, environment variables, make.

Automating portability: How to let the computer do the hard work; a case study.

The Instructor

Steven Pemberton has many years of experience porting system software written in C and other languages, to a diverse range of machines and operating systems, and giving courses on portability. He is author of enquire.c, a program that automatically determines many properties of your machine and C compiler, that is used worldwide, for instance by the GNU project's gcc C compiler.

Tutorial T3 (half day) – Widgets – Introduction to Programming with the X Toolkit
Tutor Berry Kercheval

SUMMARY

The basics of building X Window System programs using the X Toolkit will be presented.

Intended Audience

A knowledge of C programming and experience using the X Window System are required. Attendees will learn to build applications using the standard Xt Widget sets. The MIT Athena Widgets will be emphasised, but Motif and Open Look examples will also be presented. Knowledge gained will be applicable to any Xt-based widget library.

Topics Covered

Basics: Definitions. X programming model. "Modeless" programming. What are callbacks? Compiling and linking. "Hello world" Example.

Widgets: What is a widget? The Class Hierarchy and the widget tree. Programming conventions. Widget resources. How a widget works in a program.

Resources: Where do resources come from? How can a program change them? How can resources change the look or behaviour of a program.

Callbacks: How do callbacks work. Passing arguments to callbacks. Attaching actions to widgets.

Fancy Widget Tricks: Composite widgets. Geometry management. Shell Widgets. Pop-up widgets. Menus.

Event Handling: Main Loop – what it really does. Getting alternate input. Timers. Grabs and Input Focus. Dispatching events. Work procedures – doing stuff "in background".

Translation Management: Translating events to actions. Translation tables. Action procedures – defining, registering. Accelerators.

One or more example programs, illustrating all the major ideas, will be presented.

Programme of Events

Provisional Technical Programme

Tuesday 7 April, Hotel de France
Technical Sessions start at 14.00
Registration from 08.00 in the hotel foyer

Chair: Frances Brazier
Session: Opening Session (Vrije University) (NL)

Keynote: Portability
Mike Banahan (Instruction Set) (UK)

Chair: Patrick Burbaud
Session: From Portability To Porting

Marc Poinot (SEXTANT Avionique) (FR)
Software Preporting

Francis Ducroux (AFUU) (FR)
Focusing on Portability

Questions and Answers

BOF: Portability

Wednesday 8 April, Hotel de France
Technical Sessions start at 09.30
Registration from 08.00 in the hotel foyer

Chair: Patrick Burbaud
Session: Portability and Standards

Keynote: Jean Hyenne (AFNOR) (FR)

Chair: Frances Brazier
Session: Standard Environments

Donald Lewine (Data General Corp.) (USA)
Certifying Binary Applications

Derek M. Jones (Knowledge Software Ltd) (UK)
Applications Posix.1 Conformance Testing

Berry Kercheval (MTS at Protocol Engines, Inc)(USA)
X is The Worst Window System (Except For All The Others)

Chair: Barry Shein
Session: Software Patents

Richard M. Stallman
(Free Software Foundation) (USA)
The Problem With Software Patents

Chair: David Tilbrook
Session: Practice

Brian O'Donovan (Digital) (IR)
Practical Problems With Porting Software

Reinhard Koller (AT)
PHOENIX: A Health Information System Based on Unix Client Server

Steve Chamberlain (Cygnos) (USA)
What Makes GNU Software So Portable?

Glenn Kowack (EUnet) (NL)
EUnet Status and Direction

BOFS: EUnet, Copyright Legislation, Standards

Thursday 9 April, Hotel de France
Technical Sessions start at 09.30

Chair: Barry Shein
Topic: Portability in a Research Environment

Keynote: Andrew Hume (Bell Labs) (USA)

Chair: David Tilbrook
Session: Paradigms

Barry Shein (Software Tool & Die) (USA)
Inter-Fashion Portability

Gert Florijn (SERC) (NL)
Camera: Cooperation in Open Environments

Axel Schreiner (Univ of Osnabrueck) (DE)
Are we smart enough for C++?

Questions and Answers

Chair: Frances Brazier
Portability Panel

Closing Remarks

Reserve Papers

Chii-Ren Tsai (VDG Inc) (USA)
Distributed System and Security Management with Centralized Control

Kenneth J. Chan (University of Liverpool) (UK)
Porting under UNIX – Problem Areas and a Proposed Strategy

General Information

Venue and Conference Hotel

Hotel de France
St. Saviours Road
St. Helier
Jersey
Channel Islands

Telephone + 44 534 38990
Facsimile + 44 534 35354

The Hotel de France is situated within a 10 minute walk from the centre of St. Helier.

This Leisure and Conference Complex offers a classically elegant hotel together with extensive purpose built Conference facilities.

Hotel Accommodation

Rooms are being held at the Hotel de France at specially reduced rates for delegates: £45.00 per person per night sharing a twin/double room and £65.00 per person per night for single occupancy (bed and breakfast). We would like to encourage as many people as possible to stay there in order to facilitate "out of session" discussions and informal meetings. Bookings should be made direct with the hotel using the Hotel Booking Form on page 12.

From the Airport the Hotel is easily reached by Taxi. Approximate cost £7.00. Travelling time 15-20 minutes.

By Bus – No. 15 bus from Airport to Weighbridge St. Helier (£1.00) then bus number 3b, 4, 20 and 21 from Weighbridge stopping at the bottom of the Hotel de France entrance (0.45p). Approximate travelling time 45 minutes.

Jersey – Political status – Jersey is one of the British Isles BUT not part of the United Kingdom. The Island is self-governing, determines its own internal politics and fixes the level of taxation.

VAT (Value Added Tax) is not applicable in Jersey.

Currency – Jersey is within the Sterling area but the Island also issues its own Bank notes and coins.

Flights – Jersey is serviced by all UK airports.

Direct flights from Europe:
Paris, Dinard, Cherbourg, Amsterdam, Zürich and Düsseldorf.

Conference Dinner

The Conference Dinner is included in the price of the Conference fee and will be held at 19.00 on Wednesday 8th April at

La Mare Vineyards
St. Mary
Jersey

Telephone +44 534 81178
Facsimile +44 534 85210

This will be a competitive wine-tasting and buffet supper set around a beautiful old Jersey farmhouse with established vineyards and cider apple orchards.

Coaches will leave the Hotel de France at 18.30 and will take delegates to La Mare Vineyards.

Extra tickets are available for partners at a cost of 55 ECUs.

Tape Distribution

EurOpen will provide a tape distribution service at the Jersey Conference. The tape will cost 100 ECUs and will be available in two formats:

1600 bpi, 1/2 inch, reel tape and 1/4 inch, QIC-24, cartridge.

The theme of the tape will be 'Portable Tooling'.

Only EurOpen National Group or direct members are permitted to order a tape.

Students

A student reduction of 50% will be available for the Conference fee (excluding a ticket for the Social Event). A reduction of 50% will also be available for Tutorials – if space is available.

Please mark the Booking Form clearly 'Student' and provide proof of full time student status.

A 25% reduction of fees is also available to those working in a University or College.

In order to obtain this 25% reduction it is necessary to send proof of status together with the booking form.

Student Grants

Grants are being offered to assist students to attend the Conference. An application must be made well in advance of the Conference. A decision will be made before the event whether an application qualifies for a grant. Payment will not be made until after the Conference but the applicant will be able to proceed in the knowledge that the grant will be forthcoming. Applicants will be advised of their grant no later than two weeks after the closing date.

Applications should be made on the Student Grant Application form on page 11 together with a booking for the Conference on the other form. If your booking is dependent on obtaining the grant, please write on the top of your booking form: **"Grant Dependent".**

Priority will be given to:

1 Students giving a talk at the Conference.
2 Students doing work for EurOpen or a National Group.
3 Students in full time education.
4 Other deserving cases like research students.

You can apply for partial coverage of expenses for travel in Europe, accommodation and Conference fees, but not for meals. Student status, or other deserving status, must also be documented by a copy of a valid student registration card or the like. After the event, original bills must be included with the claims form.

Please note the closing date for student grants is 1st March 1992.

Electronic Mail at the Conference

Delegates can be reached during the Conference by EUnet mail. Messages will be printed, sealed and posted on the message board. To reach people at the Conference use the following address:

firstname_lastname_organisation@europen_conf.EU.net
or
firstname-lastname-organisation@europen-conf.uucp

Delegates should limit the amount of messages forwarded to this address.
Distribution lists should **not** be forwarded.
Delegates will also be able to send mail.
Worldwide TELNET access via InterEUnet is also planned.

Language

The official language of the Conference will be English. No translation will be provided.

Liability

EurOpen will not accept any responsibility for damage to property or injury to persons during the entire event. Participants are recommended to arrange for their own personal travel and health insurances.

Cancellations

It is regretted that no refund of fees will be possible in cases of cancellation, unless the cancellation is made more than one month before the start of the Conference. No cancellation will be accepted unless it is sent to the EurOpen Secretariat in writing.

How to Pay and Book

Use a photocopy of the booking form for each person. Please note that bookings can only be accepted when accompanied by payment or evidence of payment.

Telephone bookings can be accepted when paying by Credit Card. Fax bookings can be accepted when paying by Credit Card or direct to the bank (proof of payment to be faxed also).

The EurOpen Secretariat will acknowledge your bookings by sending a receipted invoice together with full details for registration.

How to Pay & Book

Costs	ECU
Full day tutorial if booked before 21st Feb	380
1/2 day Tutorial if booked before 21st Feb	210
Full day Tutorial if booked after 20th Feb	480
1/2 day Tutorial if booked after 20th Feb	310
Conference Fee	
Members (EurOpen & USENIX)	
if booked before 21st Feb	415
if booked after 20th Feb	515
Non-members	615
Conference Dinner – extra ticket	55
Conference Tape	100

Costs	£
Full day tutorial if booked before 21st Feb	270
1/2 day Tutorial if booked before 21st Feb	150
Full day Tutorial if booked after 20th Feb	340
1/2 day Tutorial if booked after 20th Feb	220
Conference Fee	
Members (EurOpen & USENIX)	
if booked before 21st Feb	295
if booked after 20th Feb	365
Non-members	435
Conference Dinner – extra ticket	38
Conference Tape	70

Payments

EurOpen is a European Federation of National Groups and as such prefers to use the European Currency – ECU's – for payments. To pay in ECUs please note the two methods of payment.

1 By Direct Payment to EurOpen's Bank Account

The Bank of Scotland
International Division
Operations Department
PO Box 86, 120 St. Vincent Street
Glasgow G2 5DZ, Scotland

Account No. 41791 ECU 01
Bank Sorting Code: 80-20-13

Please tell your bank that you will pay all charges so that EurOpen will receive the full amount.

2 By UK Cheque or Bankers' Draft, made payable to EurOpen and drawn on a UK bank. Eurocheques are acceptable, but each cheque must be 170 ECU's or less.

EurOpen is setting up facilities in Europe for delegates paying by Credit Card in ECU's, but unfortunately this is a very new facility and not fully functional at the time of printing this booklet.

If you have real difficulties in paying in ECU's you may pay in £ Sterling using the following prices **only.**

N.B. Please note that these are fixed prices and may not be converted in accordance with any exchange rate at any time. If you choose the £ option it must be at the prices stated.

To pay in £ Sterling please note the two methods of payment:

1 By direct Payment to EurOpen's bank, which is:

The Bank of Scotland
61 Grassmarket
Edinburgh EH1 2JF
Scotland

Account No. 00613997
Bank Sorting Code: 80-31-50

Please tell your bank that you will pay all charges so that EurOpen will receive the full amount.

2 By UK Cheque or Bankers' Draft, made payable to EurOpen and drawn on a UK bank. Eurocheques are acceptable, but each cheque must be £100 or less.

If you wish to pay by Credit Card (Visa, Access, Mastercard OR Eurocard) please use the £ Sterling prices only. Card details appear on the booking form.

Please indicate clearly on the booking form whether you are using ECU or £ Sterling for payment. Your invoice will then be raised in this currency.

To book a place at the Tutorials and/or Conference, complete one booking form for each person and return it with full remittance or with evidence of payment to:

EurOpen
Owles Hall
Buntingford
Herts SG9 9PL
United Kingdom

Telephone +44 763 73039
Facsimile +44 763 73255
Email europen@EU.net

Booking Form for Workshop/Conference and Tutorials

Please complete this form and send it, with cheque or evidence of payment, to *EurOpen Secretariat, Owles Hall, Buntingford, Herts SG9 9PL, UK* (Block capitals please). Please note that forms sent without cheque or evidence of payment will be returned to you unregistered.

Surname _____ Usual first name _____

Company / Organisation _____

Address _____

Country _____ Post / Zip Code _____

Telephone / Fax / Telex / Email _____

EurOpen member? (National Group or direct) Yes ☐ No ☐ Student? Yes ☐ No ☐

Please read the sections on "Costs" and remember that pre-booking saves money.

Please indicate the currency to be used for payment: ECU ☐ £ ☐
and complete this form using the relevant prices stated on page 9.

Conference
Please reserve me a 3 day place for the Technical Sessions. _____

Tutorials *(members only)*
Please reserve me a place for Tutorial No: _____ on Monday 6th April _____

Please reserve me a place for Tutorial No: _____ on Tuesday 7th April _____

Do you require Vegetarian meals? Yes ☐ No ☐

Extra ticket for Conference Dinner ☐
(One is included in the Conference price)

EurOpen
Please enrol me as an institutional member of EurOpen via the appropriate national group.

Yes ☐ No ☐

Tape
Please reserve me a copy of the Conference tape.

☐ 1600 bpi, 1/2" reel _____

☐ 1/4" QIC-24, cartridge _____

Total amount _____

Please read the section **"How to Pay and Book"**

Payment Method

☐ UK Cheque, Banker's Draft or Eurocheque. The Cheque must be enclosed.

☐ Direct Payment. The bank advice note showing details and date of payment must be enclosed.

☐ by Visa

☐ **by Access / Eurocard / Mastercard**

All bank charges must be borne by you and not EurOpen – please tell the bank this. EurOpen must receive the actual amount due.

Name as it appears on the card (block capitals) _____

Address of cardholder _____

Card Account Number _____ Date of Expiry _____

Signed _____ Date _____

Student Grant Application Form

Return to *EurOpen Secretariat, Owles Hall, Buntingford, Hertfordshire SG9 9PL UK*
(Block Capitals Please)

Closing Date: 1st March 1992

Name

Address

Post / Zip Code

Telephone

EUnet address

Position

University / Organisation

Status Speaker ☐ Helper ☐ Student ☐ Other ☐ Tick as appropriate

Full time student status or other deserving status must be documented by a copy of a valid student registration card or the like.

Expenses requested / reclaimed *(Give best estimate where necessary)*

Travel

Accommodation

Conference fee

Tutorial fee

Total amount

Date

Signature

OFFICE USE ONLY

Received EurOpen Granted Amount

Hotel de France

St. Saviours Road,
St. Helier,
Jersey,
Channel Islands
Telephone +44 534 38990
Facsimile +44 534 35354

Hotel Booking Form for EurOpen Spring '92 Conference

(6-9 April)

Please complete and return this form, together with appropriate deposit to the Conference Office at the above address.

PERSONAL DETAILS

Name and Initials

Address for Correspondence

Telephone Facsimile

ACCOMMODATION REQUIREMENTS

Please indicate type of occupancy

(single occupancy, double or twin occupancy)

Date of arrival Date of Departure

Estimated time of arrival at Hotel

The bed and breakfast rate during the conference period will be £45.00 per person per night sharing a twin/double and £65.00 per person per night for single occupancy.

All accounts are to be settled on departure from the Hotel.

DEPOSIT REQUIREMENTS

A deposit of £25.00 per person is required upon receipt of booking form. Please forward a cheque payable to the Hotel de France or alternatively we can charge your credit card account.

* Please delete as applicable

* American Express / Access / Visa

Number

Name of Cardholder

Expiry Date

Signature Date

OpenForum Conference and Exhibition

23-27 November 1992

Royal Dutch Fairgrounds, Utrecht, The Netherlands

EurOpen and UniForum are to co-sponsor the first Pan-European Open Systems exhibition and Conference

OpenForum '92 is the first pan-European event specifically for Open Systems and combines:

- A strategy and business conference
- A high level technical conference
- A set of tutorials seminars
- An Open Systems products trade show

All the ingredients are gathered to make it the most outstanding European Open System event to date.

The sponsors are the two largest independent user groups in the world. Both groups are very experienced in organising world famous Open Systems conferences. Up to now they have worked apart in their respective continents, now they are working together - the result will be staggering.

OpenForum'92 will be managed by a consortium of Reed Exhibition Companies and Royal Dutch Fairs. These are two of Europe's foremost exhibition organisers which, together, are responsible for over 100 events in Europe each year, many of them computer related. They are supported by A Plus Group, the specialist public relations and marketing communications group.

The Royal Dutch Fairs complex in Utrecht will host the conference and the exhibition.

Utrecht was chosen for its central location, being easily accessible from all the major European computer markets and benefiting of the Dutch extended trade capabilities.

"OpenForum will complement the several national Open Systems shows, providing a pan-European dimension for the first time. The timing of the event - just preceeding the beginning of the single market - is deliberate. Open Systems lower the traditional barriers to computing in the same way that single market legislation lowers trade barriers."

Michel Gien, EurOpen chairman.

Dates

23-24 November	Tutorials
25-27 November	Strategy and Business Conference
25-27 November	Technical Conference
25-27 November	Products Exhibition

Strategy and Business Conference

Conference Format

In addition to invited keynote addresses by top level executives and plenary sessions conducted by technology leaders, the OpenForum strategy and business conference is composed of three main elements:

- Panel sessions which provide multiple points of view on topics of interest to users, developers and marketers.
- Submitted paper presentations which provide a more in-depth report on a specific topic or relate experiences within the industry.
- Tutorial seminars which provide training in specific topics of interest to end-users, developers and administrators.

Panel Sessions

Eight panel session tracks are planned for the OpenForum strategy and business conference:

Track Topics

- New to Open Systems
- Organisational advantages of Open Systems
- Interoperability
- Technology futures
- Open Systems solutions
- Economics of Open Systems
- Procurement in industry and government
- Migration strategies

Some tracks may run concurrently. Sessions will be 1.5 hours in total length. Your proposal is to include the topic, content and intended audience for a session, specifying how it relates to a particular track. Your proposal is to also include the names of three qualified panellists and a biographical statement from the author. Proposals must contain at least 200 words and be typewritten.

You, in effect, are the architect of your session.

Call for Participation

Tutorials

Proposals for tutorial seminars, to be held on Monday and Tuesday, are invited. While the emphasis will be on in-depth coverage of essential areas of UNIX and Open Systems technology, proposals for introductory classes are also welcomed. Each class should be supported by comprehensive notes.

Tutors should be individuals with an in-depth practical knowledge of the topics being offered, who are able to present the material in a clear and coherent manner. Experience has shown that a full day class can be taught by either one or two people.

Submissions, by 15 March, for tutorials must include a detailed outline of the proposed class, a description of the target audience and a general description of the seminar materials. A short biography of the tutor(s) should also be included. Submissions may be made either by electronic mail or in typewritten format to the address below.

Two months before Open Forum '92 tutors will be required to submit a final tutorial programme, which shall not vary substantially from the original proposal, and one set of class notes. The class notes will be reproduced by OpenForum in binder form for each student. Instructors failing to meet this deadline will be required to reproduce their own class notes at their own expense.

Tutors will receive a honorarium upon successful completion of the class.

Business Conference

Proposals are invited for session in the business conference. Suggested topics are:

- Application portability
- Interoperability
- Open Systems and network administration
- Migration from proprietary environments
- Human considerations
- Standards and products
- Security and risk management
- Transaction processing
- Access to "old" information and data
- Distributed Open Systems environments

Extended abstracts must be submitted by 15 March 1992 by post or e-mail to the Chair of the strategy and business conference programme. They must be 1,000-1,200 words in length and be typewritten. The proposed subject should relate specifically to the conference tracks or topics. Content should reference current and new material. In addition to short biographical statement specifying the author(s) experience as it relates to the proposed subject matter, the intended audience should also be specified.

Accepted submitted papers should be from 8 to 15 pages in length and fit into a 20 minute presentation. Authors must meet the 15 August 1992 deadline for submission of the final paper. Authors who do not meet this deadline will not be allowed to present their paper. *Papers which merely profile a company's product are poorly received and therefore have little change of being accepted for presentation.*

Technical Conference

The OpenForum '92 technical conference focuses on distributed computing and features keynote addresses, technical presentations, panel discussions and a work in progress session. The opening keynote address will be given by Professor Andrew S. Tanenbaum from Vrije Universiteit, Amsterdam, The Netherlands.

Original contributions are sought that address fundamental issues of distributed computing. Topics of interest include, but are not limited to:

- Distributed system management
- Reliability and availability in distributed systems
- Security in open distributed environments
- Microkernel experiences
- Operating system support for multimedia
- Languages for distributed computing
- Communication issues of distributed computing
- Distributed file systems
- Distributed applications

Submissions

Please submit an extended abstract not exceeding 3,000 words by post or e-mail to the Chair of the technical conference programme by 15 March 1992. Each submission must provide sufficient detail to allow the program committee to assess the merits of the contribution. Submitted abstracts must indicate clearly their relevance and contribution to the overall theme of the conference.

Work-In-Progress Sessions

Authors are invited to submit Work-In-Progress (WIP) proposals limited to 3,000 words or 5 pages. Please submit WIP proposals to the Program Committee Chair by post or e-mail. These sessions provide speakers with 10 minutes to speak on current work and receive valuable feedback.

General Information

All enquiries and proposals should be addressed to one of:

EurOpen	UniForum
Owles Hall	2901 Tasman Drive, Suite 201
Buntingford	Santa-Clara
Herts SG9 9PL	CA 95054
UK	USA
Telephone +44 763 73039	Telephone +1 408 986 8840
Facsimile +44 763 73255	Facsimile +1 408 986 1645
E-mail europen@EU.net	E-mail ed@uniforum.org

Important Dates

Deadline for abstract submission	15 March 1992
Acceptance notification	10 May 1992
Deadline for final papers	15 August 1992
Deadline for Work in Progress submission	1 November 1992

EurOpen Publications

Publications available though EurOpen

			£	ECU
EurOpen Proceedings	Dublin	Autumn 1983	2.00	3
	Nijmegen	Spring 1984	5.00	7
	Cambridge	Autumn 1984	5.00	7
	Paris	Spring 1985	5.00	7
	Copenhagen	Autumn 1985	10.00	15
	Finland/Sweden	Spring 1987	20.00	30
	Dublin	Autumn 1987	20.00	30
	Munich	Spring 1990	20.00	30
	Nice	Autumn 1990	25.00	37
	Tromsø	Spring 1991	25.00	37
	Budapest	Autumn 1991	25.00	37
Directories	European E-Mail directory	2nd edition	20.00	30
4.3 BSD Manuals (EurOpen Members Only)	User's Manual Set Programmer's Set System Manager's Manual		60.00	88
;Login:	The USENIX newsletter	6 issues (one year)	20.00	30
Computing Systems	The USENIX Journal	4 issues (one year)	30.00	44
USENIX Proceedings	C++ Conference	1991	25.00	37
	USENIX Conference	Summer 1991	30.00	44
	Anaheim Conference	June 1990	25.00	37
	C++ Conference	Apr 1990	30.00	44
	Washington DC Conference	Jan 1990	27.50	40
USENIX Proceedings	Graphics Workshop V	Nov 1989	19.00	28
	Dist & Multiproc Workshop	Oct 1989	33.50	49
	Large Inst Sys Admin III Workshop	Sept 1989	15.00	22
	Baltimore Conference	June 1989	24.00	35
	UNIX Trans Proc Workshop	May 1989	13.50	20
	Software Management Workshop	Apr 1989	24.00	35
	San Diego Conference	Feb 1989	33.50	49
	C++ Conference	Oct 1988	33.50	49
	C++ Workshop	Nov 1987	33.50	49
	Graphics Workshop IV	Oct 1987	17.00	25
	Washington DC Conf	Jan 1987	20.00	30
	Graphics Workshop III	Dec 1986	17.00	25
EurOpen Sweatshirts	M/L/XL	White, Grey, Black	12.00	16
AFUU Publications The following publications are written in the French language	Dossier Benchmarks AFUU 1989 Results	Mars 1990	24.00	35
	(Summary of benchmarks sent on request	Mars 1991	34.00	50
	Réussir avec UNIX (for administrators)	1988	4.00	5
	Vivre avec UNIX 2 - Administration du Systéme	1988	4.00	5
	Vivre avec UNIX 1 - Utilisation du Systéme	Edition 1988	34.00	50
	CONVENTION UNIX 90 - Proceedings		14.00	20
	End-users and Solutions Proceeding	Edition Mars 1990	14.00	20

			£	ECU
	Technical Conferences Proceeding	Edition Mars 1990	14.00	20
	CONVENTION UNIX 90 Tutorials	Sendmail, Annexe 1 et 2	11.00	15
		Postcript	11.00	15
		Langage C++	11.00	15
		Systéme V administration	11.00	15
UKUUG Publications	UNIX – The Legend Evolves	1990 Summer Proceedings	30.00	44
	Edinburgh	Winter 1991 Proceedings	15.00	22
UniForum Publications	Your Guide to POSIX		7.00	10
	POSIX Explored: System Interface		7.00	10
	POSIX Update: Shell and Utilities		7.00	10
	Network Substrata		7.00	10
	Network Applications		7.00	10
	1992 UniForum Products Directory		70.00	105
	UniForum monthly (formerly CommUNIXations)	Cost per issue	5.00	7
	UniForum monthly - Internat edition - quarterly	Cost per issue	5.00	7
NLUUG Publications	UNIX en Standaardisatie	November 1988	17.00	25
	UNIX & Mens-Machine Interaktie	May 1989	17.00	25
	UNIX & Connectivity	November 1989	17.00	25
	Systeem Beheer	May 1991	17.00	25
i2u Publications	Proceedings of the i2u annual Convention	Mainly in Italian	160.00	225
	Catalog of UNIX Products and Suppliers in Italy	In Italian - 1100 pages	80.00	115

EurOpen Tape Distribution

	£	ECU		£	ECU
EurOpenD1 R6	42.00	60	EurOpenD13	42.00/60.00	60/86
EurOpenD2	42.00	60	EurOpenD14	42.00/60.00	60/86
EurOpenD3	42.00/60.00	60/86	EurOpenD15	80.00/120.00	120/172
EurOpenD4	50.00/60.00	72/60	EurOpenD16	42.00/60.00	60/86
EurOpenD5	42.00	60	EurOpenD17	42.00/60.00	60/86
EurOpenD6	78.00/96.00	112/114	EurOpenD18	42.00/60.00	60/86
EurOpenD7	24.00/60.00	35/86	EurOpenD19	42.00/60.00	60/86
EurOpenD8	42.00/60.00	60/86	EurOpenD20	24.00/60.00	35/86
EurOpenD9	52.00/60.00	75/86	EurOpenD21	42.00/60.00	60/86
EurOpenD10	33.00/60.00	48/86	EurOpenD22	42.00/60.00	60/86
EurOpenD11	42.00/60.00	60/86	EurOpenD23	42.00/60.00	60/86
EurOpenD12	42.00/60.00	60/86			

The first price listed is for $^1/_2$" 9-track reel tapes, the second one is for $^1/_4$" QIC-24 Sun format cartridge.

Please note that for distributions D1, D2 and D4 a copy of your source licence agreement with AT&T for at least UNIX version 7 should be enclosed.

All 1/2", 9-track, reel tapes come in tar format, 1600 bpi. 800 bpi is possible on request. Cartridge tapes come in tar format, written with dd, with a blocking of 126b. This is a so-called QIC-24 format, written on a Sun. QIC-11 is available on request.

Payments

EurOpen is a European Federation of National Groups and as such prefers to use the European Currency - ECUs - for payments. To pay in ECUs please note the two methods of payment.

All prices include post and packing. Non members should add 50% to these prices.

1 By Direct Payment to EurOpen's ECU Bank Account

The Bank of Scotland, International Division , Operations Dept, PO Box 86 120 , St. Vincent St., Glasgow G2 5DZ , Scotland

Account Number: 41791 ECU 01
Bank Sort Code: 80-20-13

Please tell your bank that you will pay all charges so that EurOpen will receive the full amount.

2 By UK Cheque or Bankers' Draft, made payable to EurOpen and drawn on a UK bank. Eurocheques are acceptable, but each cheque must be for 170 ECUs or less.

EurOpen is setting up facilities in Europe for delegates paying by Credit Card in ECUs, but unfortunately this is a very new facility and not fully functional at the time of printing this newsletter. If you have real difficulties in paying in ECUs you may pay in £ sterling using the following methods of payment:

3 By Direct Payment to EurOpen's bank. Please tell your bank that you will pay all charges so that EurOpen will receive the full ammount due. The Bank of Scotland, Account Number: 00613997, 61 Grassmarket, Bank Sort Code: 80-31-50, Edinburgh, Scotland EH1 2JF

4 By VISA/ACCESS/EUROCARD/MASTERCARD

Order Form for Publications and Software

This page may be photocopied for use. Please use capitals.

Name _____

Address _____

_____

_____

Company name _____

Email address _____

Telephone _____

Facsimile _____

I would like to order the following publications:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Tape format, either 1/2" 9-track, or 1/4" cartridge: _____

Copy of EurOpen (or national UUG) membership (or payment) form enclosed? Yes / No

Copy of AT&T source licence enclosed? (For D1, D2, D4.) Yes / No

"I declare to indemnify the European Forum for Open Systems for any liability concerning the rights to this software, and I accept that EurOpen takes no responsibilities concerning the contents and proper function of the software."

Date _____

Signature _____

*I enclose my payment in the sumof £/ECU_____
*I have completed my credit card details below

* Delete as applicable

I would like to order the following tapes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Payments

Please see the previous page for payment methods and tick the desired method below.

1 ☐

2 ☐

3 ☐

4 ☐ By VISA
☐ By ACCESS/EUROCARD/MASTERCARD

Name as it appears on the card (block capitals)

_____

Address of card holder _____

_____

_____

Card Account No _____

Date of Expiry _____

Signed _____

Date _____

Calendar of UNIX Events

This is a combined calendar of planned conferences, workshops, or standards meetings related to the UNIX operating system. The information here is collected by those listed below after an idea by John S. Quarterman of Texas Internet Consulting. The information comes from the various conference organizers, ;login:, Communications of the ACM, CommUNIXations, and many others. We encourage others to reuse this information, but we ask for proper acknowledgment, for example by including this statement.

If you have a UNIX related event that you wish to publicise then contact either John S Quarterman at *jsq@tic.com*, Alain Williams at *addw@phcomp.co.uk*, Susanne W Smith at *sws@calvin.wa.com*, or Carolyn Carr at *carolyn@usenix.org* giving brief details in the style that follows.

Abbreviations:

APP	Application Portability Profile
C	Conference or Center
CC	Computer Communication
G, MD	Gaithersburg, Maryland
GM	Generam Meeting
LISA	Large Installation System Administration
MHS	Message Handling Systems & Application Layer CommunicationProtocols
OSE	Open Systems Environment
S	Symposium
SEDMS	Symposium on Experiences with Distributed and MultiprocessorSystems
T	Tradeshow
U	UNIX
UG	User Group
W	Workshop

1992

March	INDC 92', *IFIP TC6, Finland*
March 4-7	Computers in Libraries, *Meckler, Westport, Connecticut , USA*
March 11-18	CeBIT 92, *Hannover, Germany*
March 18	Sun UKUG, Migrating from PCs to Suns, *Venue Unknown, UK*
March 24-27	AFUU, Convention UNIX 92, *CNIT, Paris-La Defense, France*
April 6-9	EurOpen, *Jersey, UK*
April 6-10	IEEE 1003, *Atlanta, Georgia, USA (location tentative)*
April 27-28	USENIX Micro-kernel, *Seattle, Washington, USA*
May	USENIX, Filesystems Workshop, *Location Unknown*
May 2-5	AIUUG, CT, *Algiers, Algeria*
May 4-8	DECUS S, *Atlanta, Georgia, USA*
May 18-22	ISO/IEC JTC1 SC22 WG15, *New Zealand (tentative)*
May 20	Sun UKUG, Paper In, Paper Out, *Venue Unknown, UK*
May 20-22	i2u C, *Milan, Italy*
May 21-22	USENIX, C, File Systems, *Ann Arbor, Michigan, USA*
Summer	UKUUG C, *QueensUniversity, Belfast, Northern Ireland*
June 2-4	UNIX EXPO West, *Anaheim CC, Anaheim, California, USA*
June 8-12	USENIX, *Marriott, San Antonio, Texas, USA*
June 21-24	Sun Users Group, *Washington DC, USA*
July	PCI, *Boston, USA*
July 13-17	IEEE 1003, *Chicago, Illinois, USA (location tentative)*
August 10-13	USENIX C++, *Portland, USA*
September 8-11	AUUG C T, *World Congress Centre, Melbourne, Australia*
September 14-17	USENIX, C, Security, *Baltimore, Maryland, USA*
September 22 - 24	SUKUG C, *Birmingham, United Kingdom*
Autumn	USENIX Security III, *Location unknown*
Autumn	ISO/IEC JTC1 SC22 WG15, *Denmark*
October 6	WG15, *Denmark*
October 19-23	USENIX, C, System Administration, *Long Beach, California, USA*
October 19-23	IEEE 1003, *Montreux (location tentative)*
October 26-30	Interop, *Moscone C, S.F., California, USA*
November 25-27	EurOpen/UniForum, *Utrecht, Netherlands*
December	UKUUG/UKnet, *Manchester, UK*

1993

January 11-15	TCOS WG, *New Orleans, Louisiana, USA (location tentative)*
January 25-29	USENIX, *Town & Country, San Diego, California, USA*
March 15-18	UniForum, *Moscone Center, San Francisco, California, USA*
March 24-31	CeBIT 93, *Hannover, Germany*
April 5-19	TCOS WG, *Boston, Massachusetts, USA (location tentative)*
April 26-30	EurOpen, *Southern Spain (tentative)*
June 21-25	USENIX, *Cincinnati, Ohio, USA*
July 12-16	TCOS WG, *Hawaii (location tentative)*
Autumn	Europen/UniForum, *Utrecht, The Netherlands*

October 18-22	TCOS WG, *Atlanta, USA (location tentative)*
October 25-29	Interop, *Moscone C, S.F., California, USA*

1994

January 17-21	USENIX, *Hilton, San Francisco, California, USA*
February 14-17	UniForum, *Dallas CC, Dallas, Texas, USA*
March 16-23	CeBIT 94, *Hannover, Germany*
March 23-25	UniForum, *San Francisco, California, USA*
April 18-22	EurOpen, *Switzerland (tentative)*
June 6-10	USENIX, *Boston, Massachusetts, USA*
September 12-16	Interop, *Moscone C, S.F., California, USA*
Autumn	Europen/UniForum, *Utrecht, The Netherlands*

1995

January 16-20	USENIX, *Marriott, New Orleans, Louisiana, USA*
February 21-23	UniForum, *Dallas CC, Dallas, Texas, USA*
May 1-5	EurOpen, *Scotland (tentative)*
June 19-22	USENIX, *Hilton, San Franciso, California, USA*

1996

March 11-14	UniForum, *Moscone Center, San Francisco, California, USA*

1997

March 10-13	UniForum, *Moscone Center, San Francisco, California, USA*

EurOpen National group addresses can be found at the back of this newsletter.

Here is a list of acronyms that you might find useful:

ACE	Advanced Computing Environments
ACM	Association for Computing Machinery
AFUU	The Association Française des Utilisateurs d'UNIX
AUUG	The Australian UNIX systems Users Group
DECUS	The Digital Equipment Computer Users Society
ECUG	The European C++ User Group
EurOpen	The European Forum for Open Systems, (was EUUG)
EXUG	European X User Group
FNUG	Federation of NCR User Groups
GUUG	The German UNIX Systems User Group
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
Interex	The International Association of Hewlett-Packard Computer Users
JUS	Japan UNIX Society
MCNTI	Moscow International Center of Science and Technical Information
NCR UUG	NCR UNIX User Group, Inc.
NIST	The National Institute of Standards and Technology
NLUUG	The Netherlands UNIX Users Group
NSF	National Science Foundation
SAB	Standards Activities Board
SERC	NSF/Purdue/Forida Software Engineering Research Center
SUUG	Soviet UNIX Users' Group
Sinix	The Singapore UNIX Association
UKUUG	The United Kingdom Unix systems Users' Group
USENIX	The Professional and Technical UNIX Association
UniForum	The International Association of UNIX Systems Users

Workshop on Micro-Kernels & Other Kernel Architecture

Seattle, Washington, USA

27-28 April 1992

This workshop is aimed at comparing and contrasting existing micro-kernels and their macro-kernel counterparts. Industry pundits are claiming that microkernel technology is the next step in kernel design. A detailed technical investigation of the microkernel technology will be made to discover whether the claims have merit or are just this year's buzz word. Our intent is to identify micro-kernels strengths and weakness in comparison to each other and to the macro-kernels that they hope to replace. The comparisons will include functionality, modularity, ease of extension, maintainability, and performance.

The first day will be devoted to talks of a tutorial nature on currently important microkernels and other kernels. The following talks have already been committed from:

- Ameoba - Robbert Van Renesse (Cornell University)
- Plan 9 - Dave Presotto (AT&T Bell Laboratories)
- Mach - Rich Draves (Carnegie Mellon University)
- NT - Dave Cutler (Microsoft)
- Chorus - Marc Rozier (Chorus Systemes)

The second day is a peer-reviewed technical track presenting original work on all aspects of microkernels or kernel architecture. Presentations range from formal research reports to practical problem-solving sessions, and will be published in the proceedings.

Pre-registration materials containing the program and hotel reservation information was mailed to the USENIX mailing list in late Frebruary 1992. If you did not receive this announcement directly and wish to be on the list for receipt of these materials please contact:

USENIX Conference Office
22672 Lambert St.
Suite 613, El Toro
CA 92630 , USA
E-mailconference@usenix.org
Telephone +1 714 588 8649

Program Committee:

Lori S. Grob	Chorus systems (Program Chair)
Edward D. Lazowska	University of Washington
Robbert van Renesse	Cornell University/Vrije Universiteit
Avadis Tevanian, Jr.	NeXT Computer, Inc.

Pre-Announcement and Call for Papers

USENIX C++ Technical Conference

Portland, Oregon, 10 - 13 August 1992

USENIX is pleased to host its fifth C++ conference in Portland, Oregon, 10-13 August 1992. Monday and Tuesday will offer tutorials; Wednesday and Thursday are technical sessions. This announcement provides early information about the date of events as well as persons to contact for further information.

Schedule of Events

- Tutorials, 10-11 August

Introductory and intermediate tutorials will be provided on the C++ language, libraries, and environments. Please contact the program chair if you wish to propose to give a tutorial or to suggest a topic you would like to see covered in a tutorial.

- Technical Sessions, 12-14 August

Topics for the technical sessions will cover the spectrum of recent research, development, and experience developing C++ software. Papers are solicited on all aspects of C++, including:

- Compilation/Interpretation
- Class Libraries and Frameworks
- Databases and persistence
- Distributed programming
- Programming environments (including design and analysis)
- Standardization and internationalization
- Experience (including maintenance and reuse)

Submissions

Extended abstracts of at most 2500 words (10 pages double-spaced) should be submitted electronically (PostScript, troff, or TeX) or eight (8) copies on paper to the program chairman.

Relevant Dates

Abstracts Due:	20 March 1992
Notification of Acceptance:	15 May 1992
Final Papers Due:	19 June 1992

Queries about the technical program and all submissions should be directed to the program chairman:

Jonathan E. Shopiro
UNIX System Laboratories, Inc. (a subsidiary of AT&T)
184 Liberty Corner Road, Room 4N-C05
Warren, NJ 07059-0908 USA

Telephone +1 908 580-4229
Facsimile +1 908 580-5631
Internet: shopiro@usl.com

Technical Program Committee

- Jonathan E. Shopiro, UNIX System Laboratories (Chair)
- Dag M. Bruck, Lund Institute of Technology, Sweden
- Theodore C. Goldstein, Sun Microsystems Laboratories
- Keith Gorlen, National Institutes of Health
- Brian M. Kennedy, Texas Instruments
- Dmitry Lenkov, Hewlett-Packard
- Mark Linton, Silicon Graphics
- Barbara E. Moo, AT&T Bell Laboratories
- Martin O'Riordan, Microsoft
- Jim Waldo, Hewlett-Packard

Call for Papers

USENIX Systems Administration Conference (LISA VI)

Long Beach, CA, USA 19-23 October 1992

The annual LISA conference provides a forum in which system administrators from a variety of sites can meet to share new ideas and experiences. A growing success for the past five years, LISA is the only conference which focuses specifically on the needs of system administrators. In previous years, LISA has targeted large installations. However, this year we are extending the scope of LISA to include system administrators from all UNIX sites.

The Sixth USENIX System Administration Conference (LISA) will be held in Long Beach, CA on 19-23 October 1992. A dual-track tutorial program will be offered during the first two days of the conference, followed by a three-day technical conference. The tutorial program will address issues in introductory and advanced system administration.

The program committee will be reviewing papers submitted on subjects including (but not limited to):

- Tools for Real-Time System Troubleshooting
- Remote/Off-site System Administration
- Tricks in User Education
- Graphical User Interfaces for System Administration
- Distributed System Administration
- Experiences Using Third-party Administration Software
- Network Growth and Performance Management
- How to Grow Your Own Junior System Administrators
- Network Management
- Wireless LANs
- System Security Monitoring
- Evaluating Performance of High-End Workstations and Servers
- Keys to Successful, Painless Upgrades
- Object Management Systems for System Administration
- Standardization of System Administration
- Heterogeneous System Administration
- System Archiving and Backups

We are especially interested in papers which provide freely available or fully described solutions to existing problems. We are also looking for papers which, in some way, advance the state of the art.

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 judging). 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. Final papers should be from 5 to 20 pages in length, including diagrams and figures. Papers should include a brief description of the site, an outline of the problem and issues, and a description of the solution. We require electronic form of the extended abstract; we require both hardcopy and electronic (nroff/troff or ASCII) form of the final paper.

Conference proceedings will be distributed to all the attendees and also will be available after the conference from the USENIX Association.

In addition to tutorials and regular technical sessions, a handful of other events will be included as part of the program. For example, the program may include special panels, work-in-progress reports, birds-of-a-feather (BOF) sessions, and invited talks. The program committee invites you to submit informal proposals, ideas, or suggestions you might have on any of these topics.

Important Dates

Extended Abstract Deadline:	29 June 1992
Acceptance Notification:	20 July 1992
Final Papers Received:	31 August 1992

Contact Information

Submit electronic copy of extended abstracts (preferably by electronic mail) to:

Trent Hein
XOR Computer Systems
2525 Arapahoe, Suite E4-264
Boulder, Colorado 80302, USA

Telephone +1 303 440-6093
E-mail trent@xor.com

Program Committee

Trent Hein (program chair)	XOR Computer Systems
Rik Farrow	UNIX World
Jeff Forys	University of Utah
John Hardt	Martin Marietta Astronautics
Rob Kolstad (board liaison)	Berkeley Software Design, Inc.
Herb Morreale	XOR Computer Systems
Pat Parseghian	AT&T Bell Laboratories
Jeff Polk	Sun Microsystems

Preliminary Announcement and Call for Papers

The Third USENIX UNIX Security Symposium

Baltimore, MD, USA, 14-17 September 1992

In cooperation with

The Computer Emergency Response Team (CERT)

The goal of this symposium is to bring together security practitioners, system administrators and system programmers, and anyone with an interest in computer security as it relates to networks and the UNIX operating system. The symposium will consist of tutorials, invited speakers, technical presentations, and panel sessions.

This will be a three-day, single-track symposium. The first day will be devoted to tutorial presentations. The following two days will include technical presentations and panel sessions. There will also be two evenings available for birds-of-a-feather sessions and work-in-progress sessions.

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)

Important Dates

Extended abstracts due	15 May 1992
Program Committee decisions made	15 June 1992
Camera-ready papers due	31 July 1992

Send seven copies of each submission to the program chair:

Edward DeHart
Computer Emergency Response Team
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213 3890, USA
Telephone +1 412 268 6179 E-mailecd@cert.sei.cmu.edu

Program Committee

- Ed DeHart (Program Chair) Computer Emergency Response Team
- Matt Bishop Dartmouth College
- Bill Cheswick Bell Laboratories
- Ana Maria De Alvaré Silicon Graphics, Inc.
- Jim Ellis Computer Emergency Response Team
- Barbara Fraser Computer Emergency Response Team
- Ken van Wyk Computer Emergency Response Team

Call for Papers

Workshop on File Systems

Ann Arbor, Michigan, USA 21-22 May 1992

Workshop Announcement

The USENIX Association and the University of Michigan's Center for Information Technology Integration are sponsoring a workshop on file systems, to be held on 21 and 22 May, 1992, on the campus of the University of Michigan, USA.

The goals of the workshop are to bring together researchers and practitioners on all aspects of file systems, including but not limited to:

- file system performance measurement and models;
- WORM and other optical systems;
- log-structured, RAID, and other high-performance systems;
- mass-storage and archival systems;
- support for replication, consistency, and mobility in distributed file systems;
- naming and location in very-large distributed file systems.

You are invited to submit original papers from any area related to file systems for presentation at the workshop and inclusion in the proceedings. There will be opportunities for "less polished" papers to be presented in the workshop sessions.

Five copies of a full paper or extended abstract should be submitted to:
Workshop on File Systems
Center for Information Technology Integration
The University of Michigan, 519 W. William Street
Ann Arbor, MI 48103-4943, USA

Papers should include an attached separate front sheet describing: the title of the paper, the name(s) of the author(s), affiliation, mailing address, telephone, telefax, and email address

Papers will be selected by the program committee based on originality, relevance, and impact.

Important Dates

Manuscripts due:	15 March 1992
Program decision:	1 April 1992
Camera-ready copy due:	15 April 1992

Program Committee:
- Peter Honeyman (CITI)
- Michael L. Kazar (Transarc)
- Larry McVoy (Sun Microsystems)
- Mendel Rosenblum (Stanford)
- Liuba Shrira (MIT)
- Local Arrangements:
- Carol Kamm (CITI)
- Judy DesHarnais (USENIX)

For further information, contact:
E-mail fsworkshop@citi.umich.edu.
Telephone +1 313 763 4403 Facsimile +1 313 763 4434

UKUUG Column

Mick Farmer
Department of Computer Science
Birkbeck College
London
United Kingdom

E-mail mick@cs.bbk.ac.uk

Mick is the Secretary of the UKUUG. His primary interests are Ornithology (restricted to the Western Paleartic at the moment because of cost) and Oenophilism (especially pre-1962 Bordeaux, pre-1980 Burgundy, and 1945 Port). His secondary interests include Software Consultancy (to pay for the above primary interests) and Distance Learning Methods (especially interactive video and hypertext). When not pursuing these and other interests he can be found at Birkbeck College (London) where he teaches in the Department of Computer Science.

He lives in Lewisham (South East London) with his wife Sue and a TV called Sonya. His neighbours have two children and a dog.

Start Bit

I've been asked to keep this short, to the point and, especially, fit it on one page. Apparently it's an attempt to reduce the cost of producing the Newsletter, while keeping it interesting. So here's an update on the wonderful conferences and workshops organised by the UKUUG.

Edinburgh Conference, December 1991

This was a runaway success, attended by over 150 people, with many others being turned away. Our thanks go to the conference organiser, Hugh Conner, and to Jim Reid, from the council, for organising such an outstanding event. The standard of papers was very high and, of course, the proceedings were available at registration (thanks to the Imperial College typesetting mafia).

Our winter meeting traditionally has a strong networking flavour and this obviously had an affect on numbers. During the last day the delegates voted overwhelmingly for more information on IP services, so a workshop is currently being organised (see below).

IP Workshop, March/April 1992

Details are currently being finalised. For further information contact our secretariat or send e-mail to ukuug-wshop-ip@uknet.ac.uk.

LISA Workshop, 1992

This workshop, scheduled for February 1992, has been postponed until later in the year, probably late summer. For information on this workshop contact our secretariat or send e-mail to ukuug-wshop-lisa@uknet.ac.uk. If you've already made enquires by e-mail, you don't need to repeat it as we've got your details stored!

Future Conferences

We're currently considering the format and frequency of future UKUUG conferences. Details will be announced as soon as they become available. For general information on our conferences contact our secretariat or send e-mail to ukuug-conf-info@uknet.ac.uk. If you have anything to say on this subject, please send e-mail to the above address.

Wedding of the Year

The chair of UKUUG, Sunil K Das, and his long-suffering partner, Hazel Tomlinson, are getting married on July 4th. We suspect that the date was chosen so that Sunil could take Hazel to America and claim that the celebrations were for them! Whatever, we wish them every happiness for the future.

Stop Bit

Thank you to all those UKUUG members who bothered to let me know that they read this column. What about the new format though? Do we still have readers?

Report on the October 1991 IEEE POSIX Meeting for EurOpen

Stephen R. Walli
EurOpen Institutional Representative

E-mail stephe@mks.com

The October meeting of the IEEE POSIX committees was a strange mixture of administrivia and crankiness. Perhaps it was merely my outlook on the latter. There was no central theme for the week, as there has been in the past with items such as the GUI Wars.

The foundation of POSIX was under attack this week in some interesting and possibly necessary ways. These discussions concern options and testability, and options (different) and profiling. The Project Management Committee (PMC) made some carefully worded statements about the continued sponsorship of P1201 (Graphic User Interfaces), the Sponsor Executive Committee (i.e. TCOS-SS Governing Board) meeting was over in record time Thursday night, and we all went home, exhausted as usual.

Other items of note include a debate over the forthcoming European IEEE POSIX meeting next Fall, and some new work items for consideration.

Let's start with the really interesting bits ....

Profiles and POSIX.1

There was considerable discussion in the POSIX.1 working group (Base Interfaces) surrounding the old issue of chopping the standard up into chunks of functionality. This discussion centres around the issue of whether or not a profile can point to pieces of the POSIX.1 standard.

The POSIX.4 (Real-time extensions) document has its functionality divided up into separate sections, each called out by an option name. For example, if an implementation supports binary semaphores then the symbol _POSIX_BINARY_SEMAPHORES is defined. This makes it very easy for a profile to point to a piece of required functionality in the real-time document.

POSIX.1 only supports a few such options for conforming implementations: NGROUPS_MAX, _POSIX_JOB_CONTROL, and _POSIX_CHOWN_RESTRICTED.

The POSIX.13 real-time profiling group wants the POSIX.1 standard further optionised to allow its functionality to be called out separately, such as an option for the file system, an option for process control, and so on. This would allow the real-time embedded profile to specify a POSIX.1 style file system, but not require POSIX.1 style process control. The embedded profile cares about threads, or possibly a single process. Nothing so "cumbersome" as multiple processes is required.

There are members of POSIX.1 who completely disagree with this view of the standard. POSIX.1 defines a portable programming interface which serves a broad range of general applications. It is viewed as a minimal set. Nothing may be removed. POSIX.1 can never be subsetted.

Part of this has to do with the sanctity of the term "POSIX". To the original standards development group, "POSIX" means POSIX.1. If you come from the POSIX.2 Shell and Utilities group, it means a POSIX.2 Shell and Utilities environment, which doesn't necessarily require a POSIX.1 system underneath it. Some people are magnanimous enough to recognise it to mean a POSIX.1 implementation with a POSIX.2 Shell, and so on.

There is a real concern that anything else isn't POSIX, nor can it use the name POSIX. A terrible vision is painted of the existing DOS environment, i.e. a loader and simple file system, being allowed to call itself POSIX by making functionality optional like process control.

Coming relatively recent to the IEEE POSIX standards development world, (I've only been mired in the muck for two years,) I quickly learned to use the term "POSIX" to mean a family of standards development projects. Indeed, this is the informal definition discussed in the POSIX.1 standard's introduction. If one wants to discuss POSIX.1 interfaces, one talks about "POSIX.1". Likewise, if one is discussing POSIX.6 extensions, one refers to "POSIX.6", and so on. This level of naming is required to remove ambiguity.

There is also no way to keep marketing departments clear on this. They will continue to misuse and abuse the term "POSIX". There is no protection for an ignorant consumer. The standard legislates what needs to be provided to claim POSIX.1 conformance. It cannot police the market place.

There is a genuine technical issue with dividing up the POSIX.1 standard. The division would need to be extremely clean and concise. Relevant definitions from one section would need to be carried with functionality discussed in the other sections. Any cross referencing would need to be handled extremely carefully. Considering the current organisation and wording used throughout the POSIX.1 standard, it would not be an easy job to pick out new options.

This discussion will likely carry on for some time as other profiling work attempts to impact POSIX.1. Already there are profile working groups for supercomputing, transaction processing, real-time systems, multi-processor systems, and a general multi-user this-is-what-UNIX-looks-like profile.

I don't believe anything should prevent a future IEEE POSIX standards working group, POSIX.42 for example, developing some

narrow but well defined application environment profile for their application domain which leverages off of the POSIX.1 domain. Implementors will implement and claim conformance to POSIX.42. Applications developers will build portable applications using the interfaces in the POSIX.42 profile. If POSIX.1 can subset carefully enough to cleanly and clearly describe pieces of functionality, it should. Future applications development/ procurement profiles, such as the U.S. government FIPS 151-1 on POSIX.1, may indeed be based on the POSIX.18 general multi-user profile. Simple. But then nothing is ever simple in POSIX.

OPTIONS and options

POSIX.1 allows implementation variance in a number of ways. The primary implementation options are a well defined, explicit method of allowing implementation variance. These are all named (e.g. _POSIX_CHOWN_RESTRICTED), and have thus been nick-named "Big-O" options.

The term "implementation defined" clearly allows an implementation to do anything it chooses, although there are documentation requirements. Even "unspecified" and "undefined" are well described. Then things get muddy fast.

Some facilities are not necessarily mandatory, and the choice of whether to implement or not is indicated by the use of the term "may". For example, an implementation may define certain environment variables, and if it does, they *shall* mean a certain thing.

For other features, an implementation is allowed a restricted choice between two behaviours, indicated by phrasing similar to "may do A or B". For example, the behaviour of an interrupted *read()* after successfully reading some data is to either return -1 and *errno* is set to [EINTR] *or* the *read()* returns the number of bytes read to that point. These later two examples are deemed "messy", by the POSIX.3.1 (Test Methods for POSIX.1) working group, and are being documented. They've been nick-named "little-o options", as some people feel they should more explicitly be called out as selectable options.

The other side of the discussion argues that these are items which no portable application should care about. The wording of the standard allows for the implementation to choose its own path. It acts as a caveat to the application developer to not depend upon the exact nature of the behaviour.

This discussion is likely to heat up over the next meeting or two.

P1201 Continues

When last we left our story, the Project Management Committee (PMC) of the IEEE POSIX sponsor executive committee (SEC) was left to recommend a suitable fate for the P1201 project on graphic user interface standards. A motion had been made to withdraw sponsorship from the project.

P1201 consists of two projects. P1201.1 is developing a windowing user interface toolkit specification. It has a long history of bloodshed between Open Look and OSF/Motif supporters. P1201.2 is developing a guidelines document on recommended drivability practice. A style guide for "feel", rather than "look".

Separate competing project requests were brought forward in the Spring 1991, one to standardise the OSF/Motif API and Style Guide, and the other to standardise the Open Look API and Style Guide. The IEEE Standards Board (POSIX's parent committee) considered that the functionality overlap between the two proposed work items and also with the work of P1201.1, was an insufficient reason to refuse project sponsorship.

At the July 1991 meeting, the POSIX working groups refused to undertake sponsorship of these competing projects for other reasons. The work was considered too immature to standardise it. It was then moved that sponsorship be removed from P1201 because it suffers the same flaws. This motion was handed to the PMC to recommend action for the October meeting.

P1201.1 requested a revision of its scope of work, and is working toward a simpler higher-level API specification. They have made real progress over the past two meetings (July and October) with the contentious members off chasing their own project requests. (Incidentally, these project requests are still being pursued in dank dark corners of the IEEE standards hierarchy.) The revised scope was recommended for approval, and the project will be reviewed again in three meetings.

P1201.2 has been making steady progress all along. They are hoping to go to ballot in the not too distant future. Continuation of their work was recommended. P1201 is allowed to proceed. It all seemed entirely to quiet and straight forward, considering the history of the past two meetings. Maybe we're maturing.

To pull the plug on these working groups at this point would be a mistake. In a few years time, there will clearly be a need for such a standard, and the technology will be very mature (read: ripe for standardisation). It will be very difficult to get employers to support such an effort, spending the money to keep people involved, if the initial effort is closed without anything tangible to show for it.

European Meeting in October 1992

The IEEE has a desire, as a "transnational" organisation, to have its standards development groups meet every other year somewhere other than the United States. The intent is to gather international input into its standards development process. This is very important to POSIX, considering its ISO development stream as an international standard being developed by a national member body. (The American National Standards Institute, ANSI, delegated responsibility for the POSIX standard's development back to the IEEE.)

There was a lot of resistance to meeting in Europe in the Fall of 1992. The problems centre around corporate approval and money.

Many corporate authorities don't view trips to Europe as "work". They still think POSIX is some kind of conference. They don't appreciate that it is a standards development working group.

Many American hotels, large enough to support the 350 working group members with about 25 to 30 meeting rooms, do not charge for the meeting rooms. They make their money from serving lunch and coffee. European hotels apparently charge for meeting rooms regardless of lunch and room bookings. This means that the meeting registration will likely be at least twice the normal $250 US. Add to this a trans-Atlantic airfare, and a higher per deum, and managers start getting really uncomfortable approving the funds.

Historically, the last meeting was held in Brussels in October, 1989. It did not draw a lot of European response. The European response that it did draw was not sustained past that meeting in many cases. Many working groups are loath to go the effort of gaining corporate approval for a European meeting, if it's not going to bring in the desired European participation.

On the other hand, if a large European contingent does show up, some POSIX working groups are concerned that their work agendas will be affected, while they adjust to bringing an influx of new blood up to speed on the issues. They want to have their cake and eat it too.

The current decision is to continue to pursue a meeting in Europe. A likely candidate country is the Netherlands. Ideas to defray some of the costs include:

- holding seminars prior to the POSIX working group meetings on POSIX related topics,
- requesting the IEEE Computer Society pick up the additional expenses, (on the order of $70,000 US.)
- charging individual attendees the additional money to attend.

I would love to hear from EurOpen members any suggestions, ideas and preferences regarding this subject. How high is European interest in an IEEE POSIX working group meeting?

New Work Items for Consideration

A number of new topics are being raised as potential candidates for standardisation by the IEEE's Technical Committee on Operating Systems – Standards Subcommittee (TCOS-SS). This is the full proper name of the committee responsible for the POSIX working groups.

Two of these candidates come from outside of the IEEE realm. The SPECmark consortium has approached the IEEE with the idea of making the SPECmark performance benchmarks an industry standard. There may be a formal presentation by the SPECmark consortium at January's IEEE POSIX meetings, but the general feeling was that standardisation of this work belonged elsewhere.

A second outside proposal was from the Rock Ridge group. These people are standardising on an optical disk standard. (This work should not be confused with the ANSI X3B11.1 WORM standard work. Please see Andrew Hume's X3B11.1 working group report in the Jan/Feb 1992 issue of ;login: or in comp.std.unix.) There is the desire to place a POSIX.1 file system on a Rock Ridge disk, and therefore an interest in input from the POSIX working groups. It was again felt that this work would better be done at arm's length, with appropriate liaisons in place.

The final new work item came from within. The Distributed Systems Steering Committee (DSSC), which overseas all of the network related standards in TCOS-SS, has proposed forming a working group to develop a standard for secure distributed computing. Depending upon who you talk to, this means Kerberos or definitely-not-Kerberos.

It will likely start as a simple Birds-Of-a-Feather meeting at the January IEEE POSIX meeting. If there's enough interest, a real study group will be formed to meet at the April POSIX meeting for the week. The result of such a meeting would be a Project Authorisation Request (PAR) to be presented to the Project Management Committee (PMC) in July. The PMC is a sub-committee of the Sponsor Executive Committee (SEC), which is the controlling committee of TCOS-SS.

If the PMC recommends sponsorship, the SEC will ratify it with a vote. The study group would become a real standards working group of TCOS-SS, with its first formal meeting as "early" as the October POSIX meeting. Thus are standards born.

EWG News

Jean-Michel Cornu
69 rue de Seine 91130 Ris Orangis
France

E-mail Jean-Michel.Cornu@enst.fr
E-mail Jean-Michel.Cornu@afuu.fr

The EurOpen working groups (EWG) are starting their job. They communicate through the mailing lists to exchange ideas and prepare their first outputs for the next EurOpen conference in Jersey (13-17 April 1992).

If you want to join one of these groups, send a mail to the chair to be put on the mailing list. If you don't have an Email address, you can call the EurOpen secretariat or send a fax (+44 763 73255). You will receive all the messages by fax.

Here are some news from the first five EWG:

Benchmarks

This group is chaired by Nhuan Doduc The mailing list is: ewg-bench@eu.net You can send mail to the chair at: ewg-bench-request@eu.net

The group met during the last EurOpen conferences in Budapest. It planned to prepare a starter kit on benchmarks.

Since then, some events have occurred. SPEC, a vendor consortium on benchmarks in the US, has proposed a PAR (new project) in IEEE, proposing to standardise the SPECMARK benchmark. There is some strong opinion in the EWG that a benchmark must be property of the users rather than the vendors. Benchmarks must be controlled by the users rather than the vendors. The benchmark developed by AFUU (the SSBA) should be a good proposal for this kind of standardisation. The SSBA may be enhanced by the EWG on benchmarks, using the work done in several user groups. The EWG is preparing its reaction to the SPEC proposal.

Internationalisation

This group is chaired by Keld Simonsen The mailing list is: ewg-i18n@eu.net You can send mail to the chair at: ewg-i18n-request@eu.net

For those who don't know what I18N means, it is an abbreviation for Internationalisation ("I" then 18 letters then "N").

The POSIX standards (systems calls and command language) have several options tohandle internationalisation. Each country, and each culture inside a country may fix these options to adapt the standards to its specific needs (character sets, date and currency formats...) The document which fixes these options for a specific country is called a local or a national profile, depending on which standard it is related: ISO/IEC 9945-1 POSIX base system interface (also called IEEE P1003.1) or ISO CD 9945-2 POSIX shell and utilities (also called IEEE P1003.2)

Today, The Rapporteur group on internationalisation of the POSIX expert group in ISO has produced a guide to produce locales and national profiles, and to first locales and national profiles: Denmark and Japan.

The EWG on Internationalisation may focus on producing other European locales or national profiles. With its wide diversity of culture, Europe is certainly the best placed to go ahead with internationalisation.

Security

The group is chaired by Herve Scheuer The mailing list is: ewg-secur@eu.net You can send mail to the chair at: ewg-secur-request@eu.net

One of the POSIX projects within IEEE is on security. The group is called IEEE P1003.6. It will produce two documents which will be submitted at the international standardisation level in ISO as extensions to the two standards 9945-1 (POSIX system interface) and 9945-2 (POSIX shell and utilities).

The EWG on security is preparing comments on these drafts and will establish the EurOpen ballot in this topic.

Copyright Legislation

The group is chaired by Jack Jansen The mailing list is: ewg-cprght@eu.net You can send mail to the chair at: ewg-cprght-request@eu.net

The group has planed to draft a statement on the copyright legislation, mainly in the user interface area.

Anton Gerold who was the first chair of this group is no longer able to continue this job. Jack Jansen is the new chair. He will start the discussion on the mailing list.

Thank you to Anton to have initiated this EWG.

Market/Marketing

The group is chaired by Pierre Scheuer The mailing list is: ewg-market@eu.net You can send a mail to the chair at: ewg-market-request@eu.net

The group is under the process of defining its scope. There is two main possibilities:

- Starting a letter presenting the major barriers for the extension of open systems and the answers of the group. This is what do the French group on market with its publication Panoram'X.
- Extending a survey which may be sponsored by the CEC, about the drivers and inhibitors in some countries. The EWG on market may extend the survey to other topics and to other countries.

National Working Groups

9 National user groups have established national working groups. These are:

AFUU, AIUUG, BUUG, DKUUG, GUUG, i2u, NLUUG, SUUG, UUES

Several other groups have planed to start WGs.

Many of the countries participate to the EurOpen Working Group through their members.

If you need more information about WGs in your country, please ask your national contact point for EWGs. The list of the contacts for any country and the list of the chairs of the EWG are can be found at the end of this newsletter.

i2u Annual Convention 1992

20-22 May 1992

Milan, Italy

There will be 3 components: Tutorials, Conference and Exhibition.

Tutorials – 20 May

The tutorial subjects were not available at the time of going to press, please contact the organisers for up to date information.

Conference – 20-21 May

The conference will be divided into technical and application sessions.

The technical sessions will focus on the latest developments including:

- Standards for binary compatibility
- Open System Architectures
- Distributed Environments
- On Line Transaction Processing
- Object Oriented Techniques
- Multimedia
- System Network Management
- Multiprogramming and Parallel Programming
- Evolution of UNIX Standards

The application sessions will focus on results achieved by end users; both the users themselves and those responsible for finding innovative solutions to meet their requirements. They will relate to their experience in using the Open System architecture; this will lead to comments on the resulting technical and organisational impacts.

Exhibition – 20-22 May

This will provide an important update on products and market developments in Italy. Leading hardware and software vendors will be present demonstrating their latest technology.

For further information please contact:

Convention Secretariat
Viale Monza, 347
20126 Milano
Italy

Telephone +39 2 2520 2672
Facsimile +39 2 2520 2508

E-mail i2u@italy.EU.net

USLE Column

Morris Schwartz
USLE
London
United Kingdom

Morris Schwartz has overall responsibility within USLE for commercial sales and marketing activities throughout Europe.

In this column Morris looks at the business needs for UI-ATLAS, and then examines the products that USL already has in place and plans to provide.

For further information on this column, please contact Gill Smith on gill@uel.co.uk or +44 81 567 7711, Gill is Marketing Manager at USLE.

UI-ATLAS and USL's Distributed Computing Framework

The computing industry has undergone a number of significant changes in order to provide computing capabilities and technology appropriate to the evolving business model. There are already standards that partly address the problem of connectivity and interoperability between systems, but undoubtedly the most important recent achievement is the comprehensive distributed computing model defined by UNIX International, the UI-ATLAS™ architecture. The UI-ATLAS model is the combined work of representatives of tens of different companies and therefore takes many diverging needs into account. In defining UI-ATLAS the UI Workgroup had a number of clear business objectives which their model had to address.

Firstly, it was clear that a comprehensive and effective distributed computing model needed to have a long life span (which meant at least ten years). It had to be able to accommodate the inevitable advancements and innovations, such as compatibility, which would occur over the medium term. The next problem was the provision of interoperability for a variety of disparate systems without requiring extensive changes to these systems. This will be the key to rapid adoption and widespread deployment of any distributed computing environment. UI also addressed the issue of investment protection, by ensuring heterogeneous interoperability.

One of the common themes that emerged from discussions was that all the members felt that object orientation was a critical component of a distributed computing architecture because it allows for the separation of the various dependent processes.

The participants in the UI Workgroup represented both competing and complementary interests. They included, for example, several companies which plan to compete with each other as well as others which aim to build on top of these products. However, the overall importance of standard interfaces was clear to both sets of companies. Everybody agreed that they could only benefit if they were able to truly make distributed computing work well for all users.

The UI-ATLAS model builds upon existing UNIX System V networking facilities and utilises *de facto* and *de jure* standard technologies already available in the market. It is, however, designed to provide an open and portable platform that can be extended to allow for consistent growth.

UI-ATLAS takes a layered approach to building the complete computing environment. This means that functions of a common type are contained in a common level of the overall hierarchy and are visible to the other layers only through well defined interfaces. In that way, one layer can evolve independently with both performance and functionality improvements without requiring the other layers to change simultaneously. The architecture has five principal layers which build on top of each other. The architecture incorporates standards such as ISO models, and follows industry standards like X/Open and POSIX. It will also provide the transition from products such as TCP/IP to OSI compliant products. Within this outline, we at USL have produced our own Distributed Computing Framework, to map USL products onto the UI-ATLAS architecture. Figure 1 shows how USL is providing products to fit the framework, it outlines those

products which are currently available from USL, and those which are partially in place.

I will now look in turn at each layer of the Distributed Computing Framework.

Base OS Services and Security

The Operating System builds on UNIX System V because of its unchallenged price/performance advantage. No other operating system has been ported over such a wide range of computers with scalability from small PCs to supercomputers.

USL obviously provides the SVR4 product family as the base of this layer. With the recent availability announcements of UNIX System V Release 4 Multi-Processor (SVR4 MP) and UNIX System V Release 4.1 Enhanced Security (SVR4.1 ES), the features and functions available from the base operating system have greatly increased. SVR4 MP gives vendors, value added resellers, integrators and end-users the first widely supported, standards compliant, high performance platform for multiprocessing applications.

SVR4.1 ES provides the highest level of security available today (B2+ level security as defined by the US National Computer Security Centre). You will see from Figure 1, that with SVR4.1 ES, USL already provides the following security elements of the framework; MAC/DAC, the audit package, trusted facilities management, secure RPC and NFS. The authentication piece is also already partially in place with an authentication of users logging in. The release amalgamating security and symmetric multiprocessing elements, UNIX System V Release 4.2 Enhanced Security/Multiprocessing (SVR4.2 ES/MP) will be available in the 1993 time-frame.

Network/Communications Services

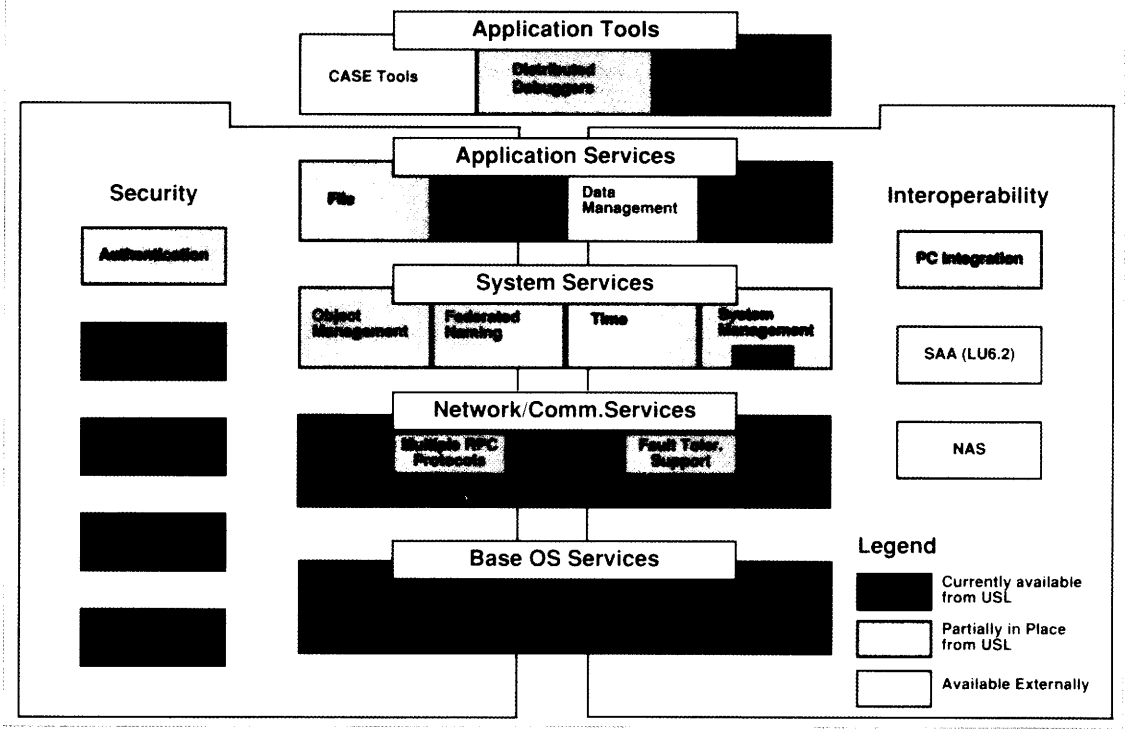
Within the networking element of this layer, USL already provides TCP/IP, and the OSI elements are satisfied with our Open Networking Platform Release 1 product. Remote Procedure Call (ONC TI-RPC) is available now with others to follow, and SVR4.1 has already been re-engineered to provide improved Fault Tolerance support.

The client/server model is already on its way to becoming a standard method for providing distributed computing services with RPC as the foundation for communications.

The System Services Layer

The system services layer comprises the core of the distributed computing model. These technologies provide the foundation upon which the components that support distributed applications and tools are built. The model integrates support for standard file systems, offering local and remote access to both UNIX-based and proprietary systems, enabling applications to access across heterogeneous networks. Any available system or network resource can be referenced from within the distributed environment. This allows for the easy migration of existing applications written under a variety of naming schemes. USL already supplies part of the System Management element of this layer as part of SVR4.1. The Object Management Group is addressing the standards required for Object Management, and the Federating Naming and Time elements are being addressed.

Figure 1. USL's Distributed Computing Framework



Applications Services Layer

The next higher layer provides applications services such as user interfaces, a transaction monitor and network management, all of which, based on object technologies, offer a highly functional and efficient method of systems and application design.

Perhaps the most significant product in this layer is the USL Open On-Line Transaction Processing System, TUXEDO. With the new extensions to the product, /HOST and /Workstation, a secure and stable foundation is provided on which to build distributed transaction based applications accepting input from traditional proprietary mainframes as well as local workstations.

User interface services provide users and systems developers with a consistent view of the computing environment, be it local or distributed. In the file systems section, NFS and RFS already support USLUs VERITAS Commercial File System and Logical Volume Manager Products.

The application services also build the foundation for application tools used to develop transaction processing applications and graphics based applications.

Application Tools

Within the application Tools layer, Object Oriented Languages are extremely important. USL has just announced the most up to date C++ product, Release 3 which now includes templates. Work is also being undertaken on distributed debuggers.

Interoperability

And finally to interoperability. The framework supports a wide range of network services which allow open and proprietary systems to interoperate. Transport independence ensures that a great number of existing applications can function within the framework without change. Both existing TCP/IP networks and the growing number of OSI compliant networks will also be able to co-exist and interoperate.

Summary

The challenge for corporate information management departments in the 1990s will be to provide truly transparent information access. Open, flexible, distributed technologies for the management, assignment and access of information services will provide companies with a competitive advantage and an opportunity to enhance their market position.

We expect three major trends in the development of corporate information technology. Firstly, the requirement will be for high performance and reliable communications networks, secondly for extensive and accessible distributed databases and thirdly, for enhanced human interface workstations.

All these requirements are addressed by the UI-ATLAS and USLUs Distributed Computing model in a highly cost-effective way. By building on top of de jure and de facto standards rather than imposing new interfaces the architecture preserves existing investments in people, training, equipment and organisation.

Through its philosophy of interoperability, the model does not require changes to be made to already published interface specifications or Application Programming Interfaces on the target systems. This again provides large economical advantages.

A standards based distributed environment enables global and shared access to resources such as printers, data and applications thereby minimising the need for peripherals. Besides these traditional shared resources distributed computing introduces a new level of economy with respect to specialised resources such as computation or database engines, traditionally too expensive for one user.

In conclusion we view UI-ATLAS as a model that enables the deployment of cost effective computing without disrupting current computing environments. It offers a rich set of features to take advantage of improved system and application services. It also provides the tools needed to develop new applications which will run over a diverse set of systems, providing a new way in which to utilise existing and future information technology investments. UI-ATLAS is based on technologies available today to solve the business needs of today and to provide the competitive advantage for the future.

At USL we are working hard to develop products within our Distributed Computing Framework to feed into the UI-ATLAS environment. We are well down the road to doing just that, and will keep you informed as soon as further developments become available.

Book Review

UNIX for Fortran Programmers
Mike Loukides
O'Reilly & Associates, Inc., August 1990
ISBN 0-937175-51-X
Price $24.95

Reviewed by Nhuan DODUC of Framentec-Cognitech, Paris, France

E-mail ndoduc@framentec.fr or doduc@afuu.fr

The 244 pages of this new volume from the very well-known *Nutshell Handbooks Series* once again deserves high level of merit. The book is rather well balanced, easily readable and above all, very practical and useful. I heartily recommend it to each and every Fortran user who wants to migrate towards a UNIX system (albeit a BSD flavour system, which happens to be my favourite). I strongly suggest it to all Fortran DP centres' library as a first-aid help-kit to the above-mentioned: Fortran-old-hands but UNIX-newcomers.

The book's avowed intention is to introduce UNIX to experienced Fortran programmer, so the titles of the 8 chapters appear not only self-explanatory but natural.

Chapter 1 lays down the minimum knowledge of UNIX that befits any real UNIX beginner; most of us browsing through EurOpen Newsletters are probably heavy-handed regarding UNIX and will probably skip this chapter, as suggested in the book; however, I am still glad of it in its role as an "aide memoire".

Chapter 2 is a thin but good enough tutorial on **vi** which is a not-too-bad text editor but whose main and only value is its availability on every UNIX system.

Chapter 3, "Compiling and Linking Fortran Program", is probably the starting chapter for most of us. The hurried users may allow themselves a first stop here since the target is reached, at least for casual, standard-conformant and machine-independent Fortran source. I do note however that the BSD flavour is heavily dominant here (for instance, the -N?n switch in pages 61-62) and regret that no warning is issued towards possible strange behaviours of (performant) compilers such as Ipi or svs ones ...

Chapter 4, "Fortran Working Environment", is just the next logical and chronological step after the successful execution of simple program(s). This chapter should be one of the most important of the book, since any real application has to really interact with the real world (e.g. environment), but, paradoxically, it's one of the thinnest. I take it as granted that interaction with the medium means heavy OS dependency, and it might not be a bad decision to keep this chapter thin and lean.

The next chapter, the 5th, is the mandatory one when your program does not want to run any more; my bad experience with non-UNIX debuggers forbids me to mingle with UNIX ones and so I dare not say anything here.

Chapter 6, "Automatic Compilation with make", is the one that I appreciate most, if only because I value *make*, as one of UNIX's most useful tools. Truly, if you only have time for two chapters, then Chapters 3 and 6 must be your choice. The next chapter, 7, "Source Management with RS", need not be here. Experienced users usually find ways to manage their code, and RCS, which is *not* part of UNIX, in the meaning that **vi** is, is not necessarily a good tool or a good example or introduction.

The 8th and last chapter deals with "Program Timing and Profiling". Its usefulness needs not be discussed but its not-present-here is not a big loss, especially since people are not going to exercise this part daily, and since vendors can supply much more sophisticated tools.

Appendix A is one of the most interesting and important part of the book, because it tells about specificities in UNIX.

Appendix B, "Program mixing Fortran and C", is a natural part of this book, probably the *other* reason why people migrate to UNIX[1].

Appendix C is a casual reminder about "Data Representation": Data representation usually is known and understood for a specific (and often proprietary) hardware family; UNIX means ability to change among hardware platforms (and not mentioning distributed computing or remote processing), so this reminder, although simple, is not useless.

Appendix D is a table of error numbers.

As a whole, this book will be helpful not only to people moving towards UNIX, but as well to the majority of programmers for whom fortran is the main interest while UNIX is just a shell albeit an interesting one. One last word: this book has helped me to better introduce UNIX to fortran users who are convinced that UNIX is an unavoidable gateway but who are just looking for a good path to UNIX. So, while there are excellent books about UNIX or Fortran, this one does act as a good summary between the two subjects, definitely a good buy.

1 *a very good public-domain package from CERN has appeared lately in comp.lang.fortran*

Book Reviews

Four books for System Administrators reviewed by

Alain Williams, Parliament Hill Computers Ltd

E-mail addw@phcomp.co.uk

These books are being reviewed together so that they can be contrasted.

UNIX System Security

Rick Farrow, Addison Wesley, 1991,
ISBN 0-201-57030-0.
(UK) Price £20.65, /Soft Back, 278pp.

This book starts with a short chapter which discusses "why" security, and outlines the US DOD "Orange Book" and it's security levels (C2, B1, ...) and a Trusted Computing Base.

The next chapter is aimed at all users. It explains how UNIX protection works, how to choose good passwords, how to set up your environment (PATH, umask, directory/file permissions) to avoid unwanted access, and ends showing how to encrypt/decrypt files. It is easy to read with lots of examples, and should be readily understood.

The rest of the book is aimed at System Administrators and Programmers. First the practices of sound administration and control are explained - much of this is user creation and password files, but also permissions on system files.

Network security is examined, from UUCP to TCP/IP based NFS, rlogin, ... In addition to good practice, pitfalls are pointed out.

Finally there is a section on how to behave when a break in is attempted, not only on a technical but also at a legal level.

In addition to examples, shell and C programs are given to illustrate points and to provide useful tools - for instance a script to check a list of files for changes in owner, mode and perhaps size and checksum.

That different versions of UNIX exist is recognised, those discussed are mainly System V releases 3 and 4, SunOS, AIX and BSD.

UNIX System Security

Patrick Wood and Stephen Kochan, Hayden Book (Simon & Schuster),
6th Edition 1990,
ISBN 0-672-48494-3.
(UK) Price £32.50, Soft Back, 299 pp

This book, outline, follows the much the same pattern as the last book with sections clearly aimed at the user, the programmer, and the administrator. It takes a lower level viewpoint (for instance by explaining disk layout at the block level). As above good solid advice is given in the art of writing secure and setuid programs.

The book was first published in 1985, and has, unfortunately, not been updated to take account of new developments. For instance the chapter on networking deals (very well) with the original and Honey-Danber UUCPs, briefly mentions RJE (for IBM dinosaurs), 3B NET and NSC (something proprietary). TCP/IP, NFS and RFS are not even mentioned.

Good clear examples are well given, and listings of useful programs are given - for instance more programs to check file permissions against a list, and one to secure a terminal with a password (but without logging out).

One nice feature is that numbered list of points to check are given where appropriate. The book also contains many references for further reading.

This book is from the System V (.2 ?) viewpoint only, the existence of other versions of UNIX is not recognised.

Practical UNIX security

Simon Garfinkel and Gene Spafford, O'Reilly & Associates, 1991,
ISBN 0-937175-72-2
(US) Price $29.95, Soft Back, 481 pp.

As with the above two books this one starts with a security overview and discussion of philosophy - how secure do you need to bother to be ?

There is no nice well defined section aimed clearly and directly at users, while not excluding a user audience this book is for administrators.

Users, setuid, umask, good password choice, etc, are explained.

Once you have the outline there is much on putting it into practice. This book does not have the Shell and C programs of the other books, but does wisely expand the concept of security to include backups and denial of service.

Networks from UUCP, to TCP/IP, FTP and NIS. The concepts are explained and the obvious loop holes (eg .rhosts) pointed out. There is a good chapter on why and how to set up a "Firewall Machine" for communication to the outside world - very well worth reading.

There is an interesting section on Kerberos and Sun's Secure RPC.

Physical security is also well covered as well as what to do (practical and legal) when you discover a break in.

The book clearly highlights appropriate differences between versions of UNIX; which *usually* means System V (up to V3.2), Berkeley UNIX and occasionally SunOS.

Good references and contact points (if you live in the USA) are also given. The book is well indexed.

Essential System Administration

Æleen Frisch, O'Reilly & Associates, 1991,
ISBN 0-937175-80-3.
(US) Price $29.95, Soft Back, 440 pp.

This book has a different aim than the other three. Security is covered and it provides an introduction to security concepts, and what the primary areas to look at are.

What this book does cover well is how to get a system running and smoothly. Most of the areas covered are the day to day tasks of a system administrator.

It starts with booting and system shutdown. Adding/removing user accounts, is followed automation of administrative work - this means cron and shell scripts.

The section on system monitoring covers process load, memory use, disk quotas.

Next come chapters on file systems and disks (mount/umount, disk space monitoring, fsck, ...), backups, terminals (how to wire an RS232 cable, and then get the stty settings right), printers, networks (TCP/IP, NFS and NIS), and accounting.

A strong point of this book is that it highlights differences between different UNIXs, explaining the different file structures, commands, operation, and very importantly how to interpret the output of the different commands - such as pstat, swap, lsps. The different UNIXs covered are Xenix, System V (V.3 and V.4), BSD (release differences are noted), AIX 3.1 and sometime SunOS.

This book has a good index. It is a book that I shall keep and use.

Summary

Note that System Administration and Security are two areas where many different vendors have put a lot of effort to differentiate themselves. You will not find a description of C2 extended systems (eg SCO and HP), neither will you learn how to use the various system administrator's shells as the books concentrate on the underlying files and commands.

If you want a book to train your users in good habits (such optimists **do** exist), choose either of the first two books for them.

For an administrator I would recomment both Garfinkel and Farrow/Spafford - books are cheap compared to the potential cost of a security violation. If I had to choose one it would be the Farrow/Spafford.

The System Administration book is just that, worth getting if you have just been given the job (possibly using various UNIXs) and want to make things work quickly. Get a book on security later.

Readers who are interested in security issues should note that there is an EurOpen Working Group on the subject. Please see the Working Group report in this newsletter.

Abstracts

Here are the abstracts from the UKUUG Winter meeting on 16-18 December 1991 in Edinburgh, Scotland. Copies of the proceedings may be obtained from the UKUUG at £15.

Thanks to Stuart McRobert <sm@ic.doc.ac.uk> who typeset the proceedings and provided the abstracts.

UKnet 1991

Tim Goodwin
UKnet Backbone

E-mail tlg@uknet.ac.uk

In the last year, UKnetsmembershiphasgrownsteadily.'At the same time we have expanded the services we offer. In October, the UKnet operation was moved to a new backbone machine, and functionally separated from the email systems of the University of Kent. This paper gives some facts and figures relating to UKnetsactivitiesin'the last year.

Managing the International X.500 Directory Pilot

Colin J. Robbins
X-Tel Services Ltd

E-mail C.Robbins@xtel.co.uk

For over two years now there has been an X.500 Pilot Directory Service spanning many countries. The operational management of the pilot service has been ad-hoc, coordinated by the author. In March this year the .I PARADISE .R project, which is part of the European .I COSINE .R initiative, started to manage the top level country data for the participating .I COSINE .R countries, and to coordinate with the North American and Australian pilots. This paper discusses the problems with the service before the .I PARADISE .R project, the steps that have already been taken by .I PARADISE .R to manage the top level DSAs and the work that will be needed in the future to manage the expanding X.500 pilot project. Extensions to X.500 needed to keep the pilot running reliably are discussed.

Setting up a Domain Name Service

Piete Brooks
University of Cambridge
Computer Laboratory

E-mail pb@cl.cam.ac.uk

The first step in getting connected to the Internet is to get an IP link. The next .B crucial step is to ensure that as well as being physically connected, other sites can find your hosts (by name), and can find out who you are (to perform some very rudimentary verification). This fundamental service depends upon the Domain Name Service (DNS) which provides a distributed and hierarchically-structured database of Internet names and addresses. Proper operation of a Name Server (NS) is a vital prerequisite for any site joining the Internet.

This paper describes some lessons learnt while steering new ShoeString – the JANET IP pilot – sites through the tangled, uncharted wilderness.

The First 9 months of UKnet IP

Peter Houlder
UKnet Backbone

E-mail Peter.Houlder@uknet.ac.uk

1991 has been the year IP started in the UK. Both JANET and UKnet net have started their services. This paper descibes the startup and first 9 months of the service. It gives some idea of why and how IP started in the UK and why it is so useful. It then describes the types of service and routing strategies employed. Finally it gives some idea of how the world InterNet interworks and the various registrations required.

Some Worrying Aspects of British Naming

Peter Houlder
Head of UKnet – the UK branch of EUnet

E-mail Peter.Houlder@uknet.ac.uk

This is a statement based on the UKnet teamsexperience'at the sharp end of international gatewaying for a country, where the networking authorities seems to be unconcerned that they are out of step with the rest of the world. We are speaking out now because we see developments occurring that could make things a lot better or a lot worse. If you do not have the time to read this fully please stop now as we consider that the issues raised cannot be considered by scan reading.

Summary of UKIC developments

Brian Tompsett
Chairman
UK Internet Consortium

E-mail bct@cs.hull.ac.uk

This paper contains a brief summary of the past year of activities for the UK Internet Consortium and illustrates some of the work we will be doing in the next year. It has been compiled from articles previously posted to the ukipnet mailing list. It is not

intended to be a definative or authorative statement on UKIC policy or activity, but as an informal guide to what has and is happening.

The JANET IP Service

Bob Day
The Joint Network Team

E-mail R.A.Day@jnt.ac.uk

An overview is given of the JANET IP Service, which has recently been launched as a value-added service on JANET. A description is given of the basic architecture of the IP carrier service provided. The introduction of the service has highlighted a number of technical issues related to the co-existence of this, and the new application services it enables, with the existing JANET-based services. The more immediately important of these are discussed in more detail.

Monitoring Heterogeneous Local Area Networks With a View Towards Efficient Network Management

Martin BeerShaun Hovers
Department of Computer Science
University of Liverpool

E-mail {mdb | shaun}@compsci.liverpool.ac.uk

The widespread adoption of Local Area Networks has meant that network performance measurement, which was previously the sole preserve of large installations with expensive monitoring equipment, must now be performed by many more computer installations without such equipment and expertise, if acceptable levels of performance are to be consistently achieved. This paper will discuss the design and use of a monitoring and management system which by analyzing the actual network stations to collect data on the current status of network traffic, uses the real data collected to provide a full description of current network performance. In addition, allows changes in both the physical network configuration and traffic densities can be simulated, by providing suitable additional data.

A suite of software to perform these functions is currently being developed in the Department of Computer Science, at the University of Liverpool. By providing much more accurate information about the present and anticipated future performance of the network, the approach discussed in this paper will considerably improve the quality of management decision making.

The X Window Training Project Sponsored by the ITTI Initiative

Jean Ritchie
Bob McGonigleJohn Wexler
University of Edinburgh

E-mail jeanr@castle.ed.ac.uk

E-mail Bob_McGonigle@edinburgh.ac.uk

The UK Universities Funding Council (UFC) is sponsoring many projects in the IT area, and one of them, at Edinburgh University, is in the field of X Window Training. The project is to produce a training pack, available at nominal cost to all centres of higher education and research, which will be used for self-teaching, assisted learning or traditional lectures.

The training pack will cover principles and use of the X Window system for managers, system administrators and users. It will also cover the use and tailoring of the standard window managers.

Performance Evaluation of Workstation-Supported X-terminals

K.J. Chan
Colin C. CharltonJ. Little
Department of Computer Science
University of Liverpool

E-mail ccc@cs.liv.ac.uk

In recent years, the marked increase in performance specification of even basic model workstations has now made it feasible to consider configuring networks of graphics terminals supported by a single workstation. This provides a cost-effective solution to the difficulties of providing responsive facilities for a large number of users within a limited budget. In adopting such a scheme however, it is necessary to determine an appropriate configuration for the application environment, in terms of the number of users such a system can effectively support and the memory capacity required. To address these issues, a series of benchmark tests were performed at Liverpool UniversitysDepartmentofComputerScience.'By varying the configuration and performing both compilation and windowing tasks, a set of performance figures were obtained which allow these issues to be resolved to a considerable extent. This paper describes the tests performed and presents the results obtained.

Tracing a Security Hole in BSD UNIX

Rob McCron
Edinburgh University Computing Service

E-mail R.McCron@ed.ac.uk

Computer security is an issue which provokes hysteria from the media, secrecy from the industry, and both anger and frustration from users. This paper deals with some of the issues which arose while running a system which was under security attack from an unknown remote source. Techniques for tracking the source of the attack are discussed, as are some post attack policies which were evolved. The specific attack with which we dealt, and eventually overcame, was related to a bug in the BSD UNIX line printer software. This paper deals with that attack in particular, and also broader security issues in general. An important feature of the information contained here is that it is of a very practical nature, being derived from experience of sustained attack over a period of several months. The extreme lengths to which we went while under attack, and the precautions which we now take on an everyday basis are discussed. A final issue, and one of great importance, is that of our dealings with CERT during and after the event; the apparent keenness of that organisation to do as much as possible to put an end to the ongoing attacks, and their obvious inability to do so in a timely and effective manner.

Campus Network Security Attitudes and Experience

Brian Tompsett
Veerle Van Laere
Department of Computer Science
University of Hull

E-mail bct@cs.hull.ac.uk

This paper describes a campus computer security study and the results obtained. The aim was to look at the overall state of computer security on the campus and to look at areas that needed improvement. A particular focus was made on system manager and user attitudes to security, as this was seen as the lynch-pin of a good security system. If you have the hearts and minds of the system managers and the users then you stand a chance of having a secure campus network. Without their support the situation is much more difficult.

Security and Open Working in the Networked Academic Community

Denis Russell
University of Newcastle upon Tyne
England

E-mail Denis.Russell@Newcastle.ac.uk

Even though academics thrive on the publication of their work, there is still a need for security services. This paper summarizes some work that has been done in assessing the perceived threats in the UK, and looks at possible defences against such threats. The emphasis in this paper is upon defence against attack rather than recovery from damage caused by a successful attack.

International standard solutions will not be available, for a long time and may well not properly recognize the multi-protocol nature of international academic networking, and Kerberos seems to be the best available system.

Even Kerberos has its problems and limitations, and this paper discusses some of them, including password guessing and non-Internet protocols such as X.29.

A system such as Kerberos is just a tool to aid in the implementation of a security policy. The role of a security policy is examined.

Investigation of Current Implementations of SLIP for SunOS and Personal Computers

Richard A. Coop
Department of Computer Science
University of Hull

E-mail rac@cs.hull.ac.uk

SLIP provides an easy and inexpensive method of connecting TCP/IP hosts with serial lines. It can be utilised to incorporate isolated hosts devoid of LAN technology into a network, provide dialup connectivity where traffic load does not justify use of a leased line and can provide local redundancy in the event of network failure.

Based upon the authorspracticalexperience,'this paper discusses current commercial and public implementations of SLIP and dialup SLIP. These include SLIP for PCs(PhilKarn'sKA9Q'and FTPsPC/TCP)andSLIPrunningunderstreamsandnonstreams'based SunOS architectures.

It highlights implementation and interoperability issues, as well as the importance of Van Jacobson header compression to improve interactive response at lower link speeds.

A Distributed Concurrent Implementation of Standard ML

David C. J. Matthews
University of Edinburgh

E-mail dcjm@lfcs.ed.ac.uk

Standard ML is a functional programming language used extensively in universities and increasingly in industry. This paper discusses a concurrency mechanism which has been implemented in the Poly/ML implementation of Standard ML and has been used on uniprocessors and shared memory multiprocessors. It is now being implemented on a distributed network of UNIX workstations. Each of these implementations is described.

The aim of this work is to produce a distributed system that will allow large ML programs to be run on a network of processors. Although eventually such a network might be a closely-coupled network of transputers, the initial design is intended for the sort of system that many organisations have, namely personal workstations on a local network. Making use of these out of office hours will provide a substantial improvement in the computing power available.

EurOpen and EUnet Addresses

An e-mail list address is an address to which if e-mail is sent, the e-mail will be forwarded to all the members on the list. This is a convenient way of sending mail to a whole group of people who share a particular interest or do a particular job, without needing to know exactly who they are, or if they have moved recently.

Several e-mail addresses have seen set up to make communication with people within EurOpen.

europen@EU.net	EurOpen Secretariat, Owles Hall
europen-jersey@EU.net	Jersey Conference
europen-exec@EU.net	EurOpen Executive Board Members
europen-gov@EU.net	EurOpen Governing Board Members
europen-papers@EU.net	Where papers for conferences are to be sent
europen-tapes@EU.net	Requests for Tape Distribution

EurOpen Working Groups

Each group has a mail address that expands to all the members of the working group. To be included in the expansion (ie to receive mail sent to the Working Group's address), add "-request" to the name; eg to join the benchmarks group (address "ewg-bench@eu.net") send mail to "ewg-bench-request@eu.net".

Working Group	*E-mail Address*
Internationalisation	ewg-i18n@eu.net
Benchmarks	ewg-bench@eu.net
Copyright Legislation	ewg-cprght@eu.net
Security, UNIX and TCP/IP Security	ewg-secur@eu.net
Marketing	ewg-market@eu.net

National Contact Points for Working Groups

AFUU Christophe Binot	binot@afuu.fr
AIUUG Mohand Tafat	Facsimile +213.2.63.87.75
BUUG A. Wambecq	wambecq@ub4b.buug.be
CHUUG Georges Schild	gs@tech.ascom.ch
CSUUG Zdenek Jirkovec	csuug@Czechoslovakia.EU.net
DKUUG Keld J. Simonsen	keld@dkuug.dk
EUUG-S Bjorn Eriksen	chairman@EurOpen.se
FUUG Kai Keinanen	kmk@fuug.fi
GUUG Anton Gerold	gerold@informatik.tu-muenchen.de
HUUG Elod Knuth	h86knu@ella.uucp
i2u Joy Marino	Joy.Marino@dist.unige.it
ICEUUG Magnus Gislason	magnus@rhi.hi.is
IUUG Brian O'Donovan	odonovan@cs.tcd.ie
NLUUG Emile Van Dantzig	emile@tmu.uucp
NUUG Arne Asplem	aras@nuug.no
PUUG J. Legatheaux Martins	jalm@di.fc.ul.pt
SUUG Sergei D. Kuznetsov	kuz@ivann.delta.msk.su
TNUUG Mondher Makni	Facsimile +216.1.787.827
UKUUG Sunil K. Das	sunil@cs.city.ac.uk
UUES Javier Romanach	jromanach@tid.es
UUGA Friedrich Kofler	kofler@tuvie.can.ac.at
YUUG Milan Palian	Milan.Palin@ninurta.fer.yu

The EurOpen Working Group Coordinator is Jean-Michel Cornu who may be reached as follows:

69 rue de Seine	E-mail Jean-Michel.Cornu@afuu.fr
91130 Ris Orangis	Telephone +33 1 69.43.48.47
France	Facsimile +33 1 69.06.83.16

EUnet Working Groups

High Level Services Cormac Callanan eunet-hls-wg@eu.net	Public Relations Cormac Callanan eunet-pr-wg@eu.net

Archiving
Martijn Roos Lindgreen
archive-wg@eu.net

As before add a "-request" to the name to request inclusion, eg "archive-wg-request@eu.net"

The EUnet European Office in Amsterdam is located at:

Office
EUnet
NIKHEF
Kruislaan 409
1098 SJ Amsterdam
The Netherlands
Telephone +31 20 5925124
Facsimile +31 20 5925155

Postal
EUnet
c/o NIKHEF
Postbus 41882
1009 DB Amsterdam
The Netherlands

EUnet national assistance can be obtained by sending mail to any EUnet country (see list below) at address:

E-mail postmaster@<country>.eu.net

Thus to reach the Autrian postmaster mail:
postmaster@Austria.eu.net

Austria
Bulgaria
Denmark
France
Greece
Iceland
Italy
Norway
Spain
Switzerland
United-Kingdom
Yugoslavia
Belgium
Czechoslovakia
Finland
Germany
Hungary
Ireland
Netherlands
Portugal
Sweden
Tunisia
USSR

Addresses of national backbones

Belgium	Dept. Computer Science K.U.Leuven Celestijnenlaan 200A B-3001 Leuven
Bulgaria	Digital System Struga 25, ap. 174, Varna - 9000
Czechoslovakia	Comenius University Faculty of Mathematics and Physics Computing Centre Mlynska dolina 842 15 Bratislava
Denmark	DKnet, c/o DKUUG Kabbelejevej 27 B DK-2700 Bronshoj
Finland	Finnish UNIX Users' Group Perustie 23 A 11, 00330 Helsinki
France	FNet, c/o CEDIA-PASSERELLES INRIA, centre de Rocquencourt Domaine de Voluceau, BP 105 78153 Le Chesnay CEDEX
Germany	EUnet Germany University of Dortmund FB Informatik, IRB EUnet Postoffice P.O. Box 500 500 D-W-4600 Dortmund 50
Hungary	Computer and Automation Institute Hungarian Academy of Science (SzTAKI), Victor Hugo u. 18-22 H-1132 Budapest
Iceland	University of Iceland Dunhaga 5, 107 Reykjavik
Ireland	IEunet Limited, O'Reilly Institute Trinity College, Dublin 2
Italy	DIST, Universita' di Genova Via Opera Pia, 11a 16145 Genova
Netherlands	Stichting NLnet t.a.v. Martijn Roos Lindgreen p/a CWI, Kruislaan 413 1098 SJ Amsterdam
Norway	NUUG Backbone, c/o USIT Postboks 1059 Blindern N-0316 OSLO 3
Portugal	PUUG Avenue 24 de Julho, 134, 7 1300 Lisboa
Soviet-Union	Demos 6, Ovchinnikovskaya naberezhnaya, 113035 Moskow
Spain	Depto. de Ingenieria Telematica Escuela Tecnica Superior de Ingenieros de Telecomunicacion Ciudad Universitaria s/n 28040 Madrid
Sweden	SUNET/KTH S-100 44 Stockholm
Switzerland	CHUUG, Zweierstrasse 35 CH-8004 Zuerich
Tunisia	IRSIT BP 212, 2 Rue Ibn Nadime 1082 Cite Mahrajane, Tunis
United-Kingdom	UKnet, University of Kent Canterbury, Kent
Yugoslavia	FER Trzaska 25, YU-61000 Ljubljana

National Group Addresses

Algeria - AIUUG
Nour Eddine Oulmi
4 Rue Beaufort
Alger 16000
ALGERIA
Telephone +213 2 744112
Facsimile +213 2 638775

Austria - EurOpen.Austria
Friedrich Kofler
Schottenring 33/Hof
A-1010 Wien
AUSTRIA
Telephone +43 222 34 61 84
Facsimile +43 222 310 44 62
E-mail kofler@at.ac.can.tuvie

Belgium - BUUG
Edgard Nyssen
BUUG Secretariat
VUB-FAC. T.W. ETRO/IRIS
Pleinlann, 2
B-1050 Brussels
BELGIUM
Telephone +32 2 641 2930
Facsimile +32 2 641 2883
E-mail buug@etro.vub.ac.be

Czechoslovakia - CSUUG
Sekretariat CSUUG
Výpočetni Centrum VSE
Nám. A. Zápotockého 4
130 67 PRAHA 3
CZECHOSLOVAKIA
E-mail csuug@Czechoslovakia.EU.net

Denmark - DKUUG
Inge & Mogens Buhelt
Kabbelejevej 27 B
DK-2700 Brønshøj
DENMARK
Telephone +45 31 60 66 80
Facsimile +45 31 60 66 49
E-mail sek@dkuug.dk

Finland - FUUG
Outi Nyman
Finnish UNIX Users' Group
Perustie 23 A 11
00330 Helsinki
FINLAND
Telephone +358 0 489 016
E-mail fuug@fuug.fi

France - AFUU
Domonique Bidou
AFUU
11 rue Carnot
94270 Le Kremlin Bicetre
Paris
FRANCE
Telephone +33 1 46 70 95 90
Facsimile +33 1 46 58 94 20
E-mail secretariat@afuu.fr

Germany - GUUG
Prof. Burkhard Stork
GUUG-Vorstand
Elsenheimerstr 43
D-8000 MÜNCHEN 21
WEST GERMANY
Telephone +49 89 570 7697
Facsimile +49 89 570 7607
E-mail sto@aic.de

Hungary - HUUG
Dr Elôd Knuth
Computer and Automation Institute
Hungarian Academy of Sciences
H-1502 Budapest 112, P.O. Box 63
HUNGARY
Telephone +361 1 665 435
Facsimile +361 1 667 503

Iceland - ICEUUG
Magnús Gíslason
University Computer Center
Dunhaga 5
IS-107 Reykjavik
ICELAND
Telephone +354 1 694750
Facsimile +354 1 28801
E-mail iceuug-s@rhi.hi.is

Ireland - IUUG
Dr T G Murphy
School of Mathematics
Trinity college
Dublin 2
IRELAND
Telephone +353 1 772941
Facsimile +353 1 772694

Italy - i2u
Ing Carlo Mortarino
i2u
Viale Monza 347
20126 Milano
ITALY
Telephone +39 2 2520 2530

Netherlands - NLUUG
Sandra Scharff
p/a Transmediair BV
PO Box 297
3720 AG Bilthoven
THE NETHERLANDS
Telephone +31 30 291718/281820
Facsimile +31 30 292294
E-mail nluug-buro@nluug.nl

Norway - NUUG
c/o Balder programvare AS
Sentrumsbygget
PO Box 1344
N-1401 SKI
NORWAY
Telephone +47 9 870550
Facsimile +47 9 876766

Portugal - PUUG
Legatheaux Martins
Avenue 24 de Julho, nº 134, 7º
Lisboa
PORTUGAL
Telephone +351 1 395 06 42
Facsimile +351 1 67 18 76
E-mail sec@puug.pt

Spain - UUES
Jose A Manas
Usarios de UNIX de España (UUES)
Telefonica I&D
C/Emilio Vargas 6
28043 Madrid
SPAIN
Telephone +34 1 337 42 22
E-mail jmanas@dit.upm.es

Sweden - EurOpen.SE
Peter Röstin
DynaSoft
Liljeholmsv 10
11761 Stockholm
SWEDEN
Telephone +46 8 726 85 60
Facsimile +46 8 18 11 45
E-mail russin@dynas.se

Switzerland - CHUUG
Patrik Eschle
Zweierstr 35
8004 Zürich
SWITZERLAND
Telephone +41 1 291 4580
Facsimile +41 1 291 4642
E-mail chuug@chuug.ch

Tunisia - TNUUG
Mondher Makni
IRSIT BP 212
2 Rue Ibn Nadime
1082 Cite-Mahrajene
TUNISIA
Telephone +216 1 787 757
Fascimile +216 1 787 827

United Kingdom - UKUUG
Bill Barrett
Owles Hall
Buntingford
Hertfordshire SG9 9PL
UNITED KINGDOM
Telephone +44 763 73475
Facsimile +44 763 73255
E-mail ukuug@uknet.ac.uk

USSR - SUUG
Dr. Kuznetsov
Kosinskaja str, 16-3-65
Moscow 111538
USSR
Telephone +7 95 374 7049

Yugoslavia - YUUG
Milan Palian
ICOS Ljubljana
Titova 118
YU-61000 Ljubljana
Slovenia
YUGOSLAVIA
Telephone +38 61 181283
Facsimile +38 61 183546
E-mail Milan.Palian@ninurta.fe.yu

Associate Members

USENIX
UniForum
PL-Open (Poland)
Bulgarian Group
Turkish Group
Israeli Group

EurOpen Secretariat

Owles Hall
Buntingford
Hertfordshire SG9 9PL
United Kingdom

Telephone +44 763 73039
Facsimile +44 763 73255
E-mail europen@EU.net