EurOpen

News Letter

Number 2

Summer 1992

Volume 12

EurOpen: The European Forum for Open Systems

Volume 12, Number 2, Summer 1992

ISSN 1011-4211

Editorial Team

Pierre Scheuer
Publication Executive

E-mail P.Scheuer@europen.euonet

Gina Baikenycz, The Instruction Set
Typesetter

E-mail gina@hoskyns.co.uk

Joe Di Mascio, Rank Xerox Business Services Ltd
Printer

Alain Williams
Editor

Parliament Hill Computers Ltd
7 Prospect Street
Caversham
Berkshire RG4 8JB
United Kingdom

Telephone +44 734 461232
Facsimile +44 734 474194

E-mail addw@phcomp.co.uk

Sue Milliken
Advertising Manager

8 Hill Close
Crowborough
East Sussex TN6 1LP
United Kingdom

Telephone +44 892 663174
Facsimile + 44 892 664568

Publisher

EurOpen
Owles Hall
Buntingford
Hertfordshire SG9 9PL
United Kingdom

Telephone +44 763 73039
Facsimile +44 763 73255

E-mail europen@EU.net

THE SINGLE MOST IMPORTANT PAN-EUROPEAN EVENT TO SHAPE THE FUTURE OF OPEN SYSTEMS

Utrecht

Holland

November

25-27

1992

UniForum

Reed Exhibition Companies

ROYAL DUTCH FAIRS

If there's one event in the 1992 calendar that's unmissable for anyone seriously interested in Open Systems in Europe, it has to be OpenForum 92.

A top-flight strategy and business conference will be combined with a high-level technical conference and a set of tutorial seminars to challenge the most demanding participants. And all this alongside a major exhibition showcasing Open Systems products.

OpenForum 92 is the only event backed by two of the world's largest associations dedicated to Open Systems, EurOpen and UniForum and managed by Europes' foremost organisers of exhibitions Royal Dutch Fairs and Reed Exhibitions.

Note the time and the place.

And be there: absolutely, positively be there!

. . . attending the conference/exhibition.

CONTACT IN U.S.

Conference : **UNIFORUM**

Tel:+1 800 255 5620

Fax:+1408 986 1645

Exhibition : **PEMCO**

Tel: +1 800 323 5155

Fax: +1 708 260 0395

CONTACT IN EUROPE

Conference : **Mrs Bea Smink**

Tel: +31 30 955 466

Fax:+31 30 955539

Exhibition : **Mrs Marjolein Jacobs**

Tel: +31 30 955662

Fax:+31 30 955539

Editorial

Alain D D Williams

Parliament Hill Computers Ltd
Caversham, United Kingdom

E-mail addw@phcomp.co.uk

Open Software Foundation

Please welcome Mark Laureys of OSF to our pages.

He starts writing a regular feature on the OSF version of UNIX to complement the USLE column that we have had from Gill Smith for a while. Our aim is to present a more balanced view of the commercial world.

Conferences Past

The EurOpen/USENIX Conference Workshop held 6th - 9th April in Jersey was a successful and interesting event, with the conference having portability as a strong theme.

The keynote speaker was Mike Banahan who presented the point of view that most UNIX tekkies have an obsession with software portability whereas most real end users believe that data portability is far more important. Don't make your mind up until you have turned the page and read what he said.

The collated Proceedings are now available from the EurOpen Secretariat at a cost of £25 or 35 ECUs (members) and £65 or 95 ECUs (non-members). I would recommend to any organisation that is concerned with portability that it should obtain a copy of the proceedings, the conference abstracts are on page 56.

Jacob Hallen was our intrepid reporter in Jersey and writes for you on page 19. Keep reading and you will see that Mick Farmer went ever further - to Moscow.

Conferences Future

There is a major event in the autumn that you should not miss. This is OpenForum '92. This will be a major exhibition and conference of pan-European relevance, it will be *the* European Open Systems show.

For more details see page 32.

Please note that the date for the 1993 conference in Seville, Spain, has been moved back a week to prevent a clash with Feria, a Spanish carnival week. The new date is 3 – 7 May. I suggest that you book your holidays now so that you can go to the Feria and then go to the conference!

EurOpen Governing Board

The EurOpen Spring Governing Board meeting was held on 4th and 5th April, 1992 at the Hotel de France, Jersey, Channels Islands. There were 44 attendees representing 22 National Groups and the Executive Committee. Countries represented were: Algeria, Austria, Belgium, Czechoslovakia, Denmark, Finland, France, Germany, Hungary, Iceland, Ireland, Italy, Netherlands, Norway, Portugal, Russia, Spain, Sweden, Switzerland, Tunisia, UK and Yugoslavia.

An application for membership from the Luxembourg UNIX User Group was accepted.

During the meeting voting took place for the Chairman and new Executive Committee using the single transferrable vote and the following were elected:

Michel Gien – Chairman
Frances Brazier
Norman Hull
Ernst Janich
Nigel Martin

Thanks were expressed to Kim Biel-Nielsen whose nomination for election was unsuccessful this time, but who has done so much work as a member of the Executive in the past.

However, the Governing Board expressed great disappointment that no new nominations had come forward from the National Groups, and to encourage new blood on the Executive Committee it was agreed that the mandate for the newly elected Executive would be for one year only and that new elections will take place at the Spring 1993 meeting.

The overall organisation and management and therefore the role of the Executive is undergoing changes by virtue of the fact that much work done in the past by the Executive is passing to sub-committees of the Governing Board and also to the Executive Director, Pierre Scheuer.

Governing Board sub-committees have been set up to:

- examine income strategy and fees,
- look at the implications of moving EurOpen operations to operate as an International Association under Belgian Law,

- formulate the needs and requirements of EurOpen Users for networking services in the medium and long term,
- formulate a future perspective for the Newsletter,
- propose the directions for future EurOpen events.

In addition, ongoing actions discussed at the meeting included proposals for:

- different types of future events such as educational workshops,
- investigations of whether readers of the Newsletter might be interested in receiving abstracts in English of interesting papers from other National Groups,
- examining the possibilities of holding member and e-mail directories on central computer site,
- creating a framework in which the National Groups can establish a better rapport with their Backbones and foster a spirit of cooperation.

The sub-committee members are drawn from those with relevant experience and interest in the national groups. The use of such members helps to reduce the large work load of the EurOpen executive committee, and will bring in fresh new ideas. It is hoped that the fruits of this initative will be a EurOpen that better serves it's members.

EurOpen Working Groups

Due to pressure of work Mr. Jean-Michel Cornu has had to stand down as co-ordinator of the EurOpen Working Groups and thanks are expressed to him for his work in the past. A new co-ordinator is being sought. The formation of Working Groups is being actively encouraged within the National Groups as is the participation of those local Working Groups at the EurOpen level.

We warmly thank him for his splendid efforts in starting the EurOpen Working Groups and providing a solid foundation for their future development. Read his last report on page 47.

USENIX

Representatives of USENIX were also present at the Governing Board meeting and Barry Shein expressed the view that it was very worthwhile and important for UniForum, USENIX and EurOpen to meet together regularly.

He informed the meeting that USENIX had recently held an election and now had a new Board as follows:

President Steve Johnson, Vice President Michael O'Dell, Secretary Evi Nemeth, Treasurer Rick Adams

Directors Eric Allman, Barry Shein, Lori Grob, Tom Christiansen.

Next Governing Board Meeting

The next EurOpen Governing Board meeting will take place during the OpenForum Conference in Utrecht in November, 1992, in the week-end preceding the OpenForum conferences in Utrecht.

Erratum

In the last issue the wrong contact point for the UK contact for the League for Programming Freedom was given. The UK contact is:

Clive Jones
Christ's College
Cambridge CB2 3BU

E-mail crj10@phoenix.cambridge.ac.uk

Portability

Mike Banahan
Principal Consultant
The Instruction Set
London
United Kingdom

E-mail mikeb@inset.co.uk

Mike is well-known for his leading role in UNIX, C, C++ and Object Oriented technology and for his popular books, 'The C Book' and 'UNIX The Book'. As Chairman of the European C++ User Group and together with his teaching and consultancy roles at The Instruction Set (for example, technical consultant to X/Open over much of the last 5 years), Mike is well equipped for his role on The Instruction Set's courseware design team, with special responsibility for the Programming Languages and Object Oriented Technologies curricula.

Introduction[1]

There is, of course, a very old joke about UNIX and portability. If you want to hear me repeat it, then you have to listen to the live session of this talk - I have no intention of spoiling it (more) by writing it down.

It's sometimes hard to remember, now that UNIX has grown up and developed a lot of middle aged spread, that once upon a time it was seen as being more or less revolutionary. An operating system that could be run on more than one kind of computer? No, stop messing about. We all know that that's impossible ... or even if it isn't, it will be a joke system that runs too slowly to be useful.

In the bad old days - up to the late seventies and including the early eighties - there wasn't much at issue. If you wrote a piece of software, you knew full well that it was more or less a waste of time trying to do anything other than get it working on just the one kind of machine. In full knowledge of that, you could indulge yourself in all kinds of tricks to save memory and screw the last bit of performance out of the machine; fragments written in assembler, use of overlapped I/O, careful selection of disk sector placement - all of which were great fun and allowed a degree of macho posturing: "I can squeeze the bootstrap into 13 bytes".

Of course, it is easy nowadays to look back and to laugh at that. Unless you have recently left the mainframe world, that kind of thing appears increasingly curious and quaint. (Much of mainframe programming is still like that: ask the people at the British Driver Licensing centre how long ago it was that they stopped programming in assembler). In some respects it was hard to justify: a lot of the "tricky" coding didn't do a lot for performance and the effect on portability doesn't bear thinking about. Still there were reasons for it. Let's think about the typical hardware of the day.

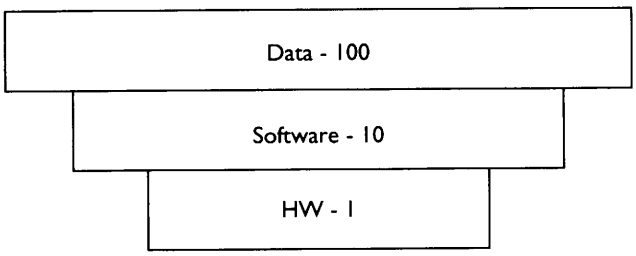
My first "large" UNIX system was at the University of Bradford, in 1978. We had just installed a very powerful DEC PDP11/60, with extensive disk storage. The machine was equipped with an almost unimaginable amount of memory: 256K bytes in total. There were three exchangeable disk drives: RK07s, each the size of an industrial washing machine and holding 24Mb per pack - 72Mb on on-line storage. The processor was fast for its day; a 16-bit machine that delivered probably in the order of 0.25 - 0.5 VAX MIPs. That machine served a user community of 1500 students (that's what wc reported the number of lines in /etc/passwd to be), of whom about 25 would be logged-on at any one time. Mind you, with 25 students all compiling Pascal, the performance was grim. Less than 10 it was bearable, and with only one or two it was very good indeed. Given the amount of memory, disk and MIPs available in modern hardware, the surprise is what bad use they seem to make of it; but that's another matter.

Hardware used to be very, very expensive indeed. I seem to recollect that the system described above cost something like ten times my annual salary as a young lecturer. There was some justification for the argument that it was worth spending programmer time on tricks to make the best of the hardware.

DEC had a pretty fair share of the market for computer science departments in those days; PDP 11 machines were widespread in the UK and very very common in the US. A lot of academic work was being done on

them. In some cases, researchers were able to exchange programs and data between different departments or faculties and take advantage of each others' work in a way that had not been possible before. The PDP 11 was becoming a kind of commodity product in those circles. Unfortunately, even though there was just one hardware architecture, portability suffered from the fact that there were some four or five different operating systems in use: UNIX, RT11, RSX11, DOS (nothing like PC-DOS) and Mumps. Naturally, they completely incompatible with each other. Moving software from one to the other required a major rewrite.

The academic community was having enough trouble with portability. What about the commercial world? There, the situation was much worse. Whilst academics could (at a pinch) kid themselves that rewriting a piece of software was an activity that in itself had value of a kind (there*might* be a paper in just writing up that activity), there has never been much mileage in that argument as far as the commercial sector is concerned. The commercial folk have a different problem too. The software is only part of it. For many organisations, there is a pyramid of values which looks something like this:



In words: most commercial organisations are a lot less worried about their software than their *data*. It's data that's the life-blood of large organisations. The programs are just a way to access it. As a result, any improvement in software portability is of only marginal interest if it means that the data portability is not correspondingly addressed. We'll come back to that later, because it is important.

The prominence of software portability has come in phases. The first one is more obvious by what it was not, rather than what it was. This is the phase where getting software to work at all was challenge enough. Portability was a non-issue: "To what? Why?" would be the response. High-level languages were not primarily concerned with portability; the reason for them was *productivity*, with productivity and efficiency of the resultant software being the two most important tradeoffs to worry about. We can point to the two important languages: COBOL and FORTRAN as examples of tradeoffs in one direction or the other.

The emergence of high-level languages had a curious side-effect though. Despite the fact that they were not much concerned with portability, they became enormously important as *enablers* of it. Once COBRAN had become established as important languages, the standards machinery rolled into play. There was no way that a standard for either language would be accepted if any of the significant hardware manufacturers voted against it, so we ended up with specifications of languages which were quite highly portable. Since these were the common subsets that everyone could agree upon, most vendors felt free to adorn their own implementations of the standards to "make them better implementations". There is no doubt that the resulting lock-in was seen as an advantage, but realistically, most major customers were locked-in to the vendor anyway, so the justification on efficiency or performance grounds was often a reasonable one. Most customers preferred performance over portability and used the extensions.

As ever, not everyone does the obvious thing. A few strange, deranged souls looked at the high-level languages and said to themselves "We can write software which is independent of the target system!". Thus was born the second age of portability.

The second age of portability peaked in the mid to late seventies. This was marked particularly by the "Software Tools" book written by Kernighan and Plauger. Here was a set of standardised software tools intended to be usable across a wide range of target environments, relying on the portability of programs written in standard FORTRAN (although FORTRAN is hideous, so the tools were written in RATFOR, with a RATFOR to FORTRAN translator as part of the toolset). Data portability was a non-starter, so the tools relied on the FORTRAN model of the world: all data is broken into logical records, logical records are packed one to a card - oops, sorry - one to a line. Unsurprisingly, the software tools work profoundly affected the UNIX utilities.

The second age of software portability is characterised by a profound belief in the importance of being able to port software from one kind of computer to another. Remember, the reason for caring at all about portability in those days was principally because it couldn't be done easily. There were numerous significant computer vendors, all selling very different machines with their own operating systems. Portability of any kind was a real challenge. Once it had been demonstrated as being realistic, a whole new group of believers were spawned and various ways of achieving it were investigated. Principal amongst them are the Mohammed Vs. The Mountain techniques.

The problem is this: there is a large installed base of disparate machines and operating systems. You want to write software which delivers the "same" results on any of them. What do you do? You can accommodate their peculiarities (Mohammed goes to the mountain). You pick a widely-available high-level language, and use a carefully selected set of coding techniques for avoiding known hardware dependencies. You take care to insulate yourself from word-length and byte-ordering problems where they appear. You invest a lot of time and effort in isolating the known portability problem areas and push them off into libraries, with carefully abstracted behavioural characteristics. You implement the library for each target machine/OS combination. With care and a lot of experience, you can write

extremely portable software in this way - although there is rarely *no* work to do when porting to a new target.

The alternative is to make the mountain come to Mohammed. You define a virtual machine and operating environment, then write your software to work in that environment. For each target, you implement that environment - then all the software works. Classic examples of this approach are the BCPL Ocode environment, or perhaps more widely known, the UCSD Pcode implementation of a Pascal programming environment. Because both of these are based on interpreters, they tend to suffer from poor run-time performance. There's no doubt that they work though. The ultimate in moving the mountain must be the astonishing work done by Insignia Solutions with Soft-PC: a complete software implementation of the IBM PC-AT environment, running on any UNIX platform. An impressive piece of work.

Between those two extremes lies UNIX. UNIX offers a standardised environment, even though the underlying hardware platforms may be different. The UNIX equivalent of Pcode is the C programming language (or some other appropriate HLL) coupled with standards for filenames, contents of certain important files and a standard way of representing textual data. Minor variations between derivatives of UNIX have to be addressed, as may some hardware characteristics which still show through the programming languages. It's clear that C was not designed with portability as its principal objective.

The early eighties saw the first widespread realisation that software portability could really be achieved. High-level languages by themselves were never enough to deliver everything that people wanted. Leaving aside the data portability matter, practical barriers that still remained were points like the ones mentioned above. Filenames are the bane of the portable software author. Lack of standards for record-structured file access, file length, file locking and other mechanisms are essentially show-stoppers. Systems which distinguish between file types (binary, text, display, source etc.) are a disaster. The significant contribution of UNIX was that, like a language, it did not provide "solution" level services, but it *did* give a target environment which was a) usable with some effort, and b) itself portable with some effort to most hardware families. The approach was unusual for its time: give the user the tools to finish the job rather than attempting to do the job itself. This was the opposite of trends in commercially-oriented operating systems of the day.

The UNIX approach does suffer from a significant problem. Full compatibility or Pcode techniques allow for software "shrink-wrapping" to a degree that cannot be achieved in UNIX environments at present. The PC-DOS market is the most successful example of what is possible with a "standardised" environment. UNIX systems still do not reach that level of compatibility for two reasons:

- Most HLLs do not give 100% portability
- Not enough of the UNIX environment is fully standardised yet

Binary-compatible systems have proved over the years that a software developer can reasonably assume that a product tested on one such system will work on another one. Source-level portability still fails to deliver that, so we notice a real cost to the vendor:

- Each port needs to be re-tested
- There is bound to be "version skew"
- There is a significant inventory control problem

There are now two environments in widespread use largely because they offer portability in some shape or form: DOS and UNIX. DOS represents a homogeneous target for software vendors where there is no confusion of responsibilities: almost without exception, the hardware manufacturers make and sell hardware, the software is provided by organisations which concentrate only on software. There is a true mix-and-match marketplace where a large number of customers have reached the level of technical sophistication which allows them to do their own systems integration work.

The UNIX market is more of a hybrid. Systems integration is not generally within the range of the individual purchaser and the software is still more closely coupled to the system than it is in the DOS marketplace. Some software vendors do manage to target nearly all of the mainstream hardware platforms, but naive early expectations of a DOS-like marketplace have been tempered with reality. Many smaller software vendors sell only on a restricted range of platforms. Key for them is that they at least have the *option* of porting at relatively low cost. The portability potential of UNIX has acted like a the theory of the nuclear deterrent: you don't have to use it to gain the benefit! Because the software authors know that software portability is possible, they target UNIX. Occasionally they do actually port their software, but as often as not they target only the platforms which are important in their market niche.

By far the most important effect of the portability that UNIX introduced has been that it *has* created a commodity hardware market. For the first time, because of the existence of some widely available portable packages, various vendors' hardware offerings can be directly compared one against one. We now have innovation and a price/performance regime that would never have happened without UNIX to drive it. The majority of UNIX system purchasers buy it not for its portability, but because it costs less than the alternatives.

Now we need to consider the third phase of portability; data portability. As the diagram showed, for most commercial organisations and a smaller number of others, the principal investment is not in software but in data. That is one strand which must be kept in mind at all times. The second strand is the fallout from the second phase of portability. Remember, the driver for the second phase was the proliferation of incompatible

environments WHICH ALREADY EXISTED. UNIX and DOS have now marginalised many of those incompatible environments; the industry is moving to a state where the number of incompatible environments is much reduced. Software portability is now rapidly fading as a major issue: that problem has been "solved" and becomes uninteresting.

The industry's focus is now shifting to connectivity and compatibility of data formats. It is still early days for this subject, but straws are in the wind already. Large-scale storage of high-value data is the most important thing to manage and it is in that area that the first signs are visible. Until recently, SQL has been the only standard relevant to the database area. However, the work of SAG (the SQL Access Group) is getting closer and closer to becoming reality. Here, we have standards for database front- and back-ends, so that the back-end also starts to become a commodity product. There are now specialist hardware vendors who focus particularly on building systems to support database back-ends. Once interfaces for the back-end servers are fully established, the server becomes a network component in its own right. Whether it is running UNIX or not becomes irrelevant. In fact, it is highly unlikely that UNIX will be running on the server except perhaps to support the administrative and network utilities. All the evidence points to the database package bypassing the operating system to the extent that it can no longer be said to be using the operating system at all. Application-to-application integration is likely to be in the province of the Object Request Broker (ORB) from the Object Management Group. Here, interfaces for inter-application communications and data formats begin to dominate.

And so the wheel rolls on, finally leaving UNIX behind it. The process started late in the Seventies, where portability was demonstrated as a real hope for the first time, moves to a phase where the main enabler is itself increasingly a marginal component. We see instead distributed systems using technology such as TCP/IP, NFS, X, SAG and the ORB. One day, someone may even get OSI networking to do something useful! The portability of software becomes less and less interesting; what matters is whether it can access and process the data.

The bulk of computer users have never cared much about the software and still less about the hardware. The more that the emphasis shifts to what really matters, the data, the more the industry moves from adolescence to its maturity.

Portability in a Research Environment

Andrew Hume
Howard Trickey
AT&T Bell Laboratories
Murray Hill, New Jersey, USA

E-mail andrew@research.att.com

Software developed within a research environment is both easier and harder to make portable than software developed within a industrial environment. Projects utilising unique aspects of the underlying environment may prove difficult or expensive to port. On the other hand, access to the system source can sometimes solve portability problems locally.

This report describes some of the approaches used within the Computing Science Research Center at AT&T Bell Laboratories for writing portable software.

Introduction

This report describes some aspects of how portability interacts with the research done within the Computing Science Research Center at AT&T Bell Laboratories. For some center members, portability is irrelevant – all they need is TeX or a Fortran compiler. For others, particularly those who investigate new directions in paradigms, portability is simply not an issue, as their work is not portable. A third group has a much harder job; their work is mostly portable but is made less so every time they use a novel part of our environment. They are very much aware of the tension between research and standards; standards are fine when your research can build upon them, but when your research explores new ways of doing a particular thing, standards are often a millstone around your neck.

Our Environment

The hardware supporting our Center's computing environment is fairly ordinary: we primarily work on VAX 8550s and SGI multiprocessors. There is an ECL MIPS system (6280), a token Sun, a Cray X/MP-28 and some VAX 11/750s we can't seem to get rid of. For terminals, we use mainly MIPS Magnums and Gnots (a terminal designed in our center by Bart Locanthi), although there are several Nexts and NCD X terminals.

Our networking is split between Datakit, a virtual circuit based network, and Ethernet. (There is also an experimental 45Mbit/s link to a few other sites.)

The software environment is rather more mixed. The VAXen run 10th Edition Research UNIX [Hum90a], most of the SGI machines and the Cray run the manufacturer's version of the UNIX system, and the rest run Plan 9, a new operating system developed in our Center [Pik90a].

So How Do I Write Portable Software?

It depends a lot on how big the thing is you are trying to make portable, and to how many different systems you wish it to be portable to. There is a spectrum of portability; we give some examples below.

Approach 1: Use a trivial UNIX system environment

There is a class of programs that just don't need much from the environment. Take as an example *gre*, a new version of *grep*. The sole requirements it has on the environment are

<ctype.h>	**read**	**printf**	**longjmp**
open	**write**	**malloc**	**setjmp**
close	**fwrite**	**fflush**	

(The names above are library functions or system calls; names in <> are header files.) Until recently, programs of this kind were extremely easy to port. However, a growing need to work with old C compilers, ANSI C compilers (particularly with strictness flags turned on) and C++ compilers has made the source uglier and more complicated.

The advantages of using function prototypes are too great to throw away, so we write new programs with function prototypes. Rather than litter up the source with `#ifdef`'s to accommodate non-ANSI C compilers (and C++), we use an awk script to convert files to use old-style function headers before distributing the files outside of our center.

Approach 2: Use a rich UNIX environment

Another class of programs uses much more of the UNIX environment, or in current parlance, the environment defined by ANSI C [ANS89a] and ISO 9945-1 (which was POSIX 1003.1). Take as an example *mk* [Hum87a], a new version of *make*. Some of the routines it needs from the environment include

<ctype.h>	**execle**	**getuid**	**print**	**strchr**
_exit	**exit**	**lseek**	**printf**	**strcmp**
access	**fflush**	**malloc**	**read**	**strcpy**

atoi	**fgets**	**memcmp**	**readdir**	**strdup**
atol	**fopen**	**memcpy**	**regcomp**	**strlen**
close	**fork**	**memset**	**regexec**	**strncmp**
closedir	**fprintf**	**mktemp**	**regsub**	**system**
creat	**fputc**	**open**	**signal**	**time**
dup2	**fwrite**	**opendir**	**sprintf**	**unlink**
environ	**getgid**	**perror**	**stat**	**utime**
errno	**getpid**	**pipe**	**strcat**	**wait**
write				

Except for the regular expression routines, these are all covered by the ANSI C and POSIX 1003.1 standards. (The regular expression routines are covered by POSIX 1003.2.)

Most, if not all, providers of UNIX have promised to embrace POSIX in some way or another, so eventually programs written to these standards ought to be very portable. Each new release from the major manufacturers comes closer to POSIX conformance, but most are not quite there yet. And even if the vendors started to ship POSIX conformant systems, there are many users out there who won't have upgraded their operating system and, in fact, may not even do so until they buy a new computer. Thus, any exported code is likely to run into any of the following levels of POSIX and ANSI C conformance:

- Older BSD-based UNIX systems: Most core functions are there, though some string functions have different names. Headers are mostly missing or wrong. Typically, the C compiler is pre-ANSI.

- Older System V-based UNIX systems: Most functions are there. Many header files are there, though they lack function prototypes and may have extra things in them. Typically, the C compiler is pre-ANSI.

- First Pass attempt at POSIX compliance: All or almost all functions are there. Header files are there, but not up to par in various ways (perhaps no prototypes, may include extra stuff that shouldn't be there).

- POSIX and ANSI C compliance. Any extra stuff in header files is either allowed by the standard, or properly protected by some sort of extension feature-test symbol. It appears that SGI's IRIX release 4 reaches this level.

We have tried to anticipate the industry trend towards standard languages and environments. On all of our machines, we have built an ANSI C/POSIX environment called APE, consisting of an ANSI C compiler (including preprocessor) which searches `/v/ape/include` instead of `/usr/include`. The files in `/v/ape/include` are mostly the same across machines, but there are some machine-dependent ones too. Only ANSI C headers may be included if no feature-test symbols are defined. If `_POSIX_SOURCE` is defined, then POSIX headers may be included and more (POSIX-defined) symbols are visible in the ANSI C headers. Note that this is deliberately picky; our goal is that software that runs under APE locally will at least build on any conforming ANSI C/POSIX environment. The downside is that software developed under more user-friendly environments may have a few easy to fix problems, and some potentially more awkward problems, compiling under APE. The easy problems involve using the feature-test symbols, which APE applies as strictly as the standard allows. The harder problems involve system parameters which, in principle, are only known at runtime. One such example is the number of files that a process can have open at once. The symbol `OPEN_MAX`, if present, defines this maximum. Even though the value of this is known (and fixed) in all our implementations, we do not define this symbol because it is not compulsory according to POSIX and thus, we force our programmers to cope with the hardest case (where it is not a compile-time constant). Most often, this just means changed statically declared arrays into `malloc`'ed arrays. The APE compiler loads against a library which is constructed by extracting many functions from the system `libc.a` and providing others from source of our own.

One problem with the POSIX solution is that ANSI C and POSIX 1003.1 don't quite provide enough functionality to implement some programs. Some omissions, such as `popen` and `ftw`, are easy enough to simulate, but it is annoying that they are not there. Standard network access functions are a bigger problem. Eventually, we expect various POSIX subcommittees will solve these problems. Until then, we find it necessary to maintain a small additional library to go along with APE. The feature test symbol for these additional routines and their headers is `_RESEARCH_SOURCE`.

Approach 3: A little piece of England

During the latter part of the 80s, there was an explosion in the number and variety of UNIX systems. Every day seemed to bring another system with a small (sometimes more) number of gratuitous differences to existing UNIX systems. Rather than design a system to a common subset of these systems, our developers would assume a specific environment and rely on a system-specific library to implement this environment.

This style of portability works quite well with largish subsystems such as *upas*, our Research mail system, and the File Motel, our file backup system. The success of this technique depends on how well the support environment is designed and how much effort is needed to port the support library. In practice, this has never seemed to be much of a problem, but there is a lot of interest outside our Center on automating this process. People seem to like trusting baroque programs, like *config*, which act as oracles on what UNIX system you have and thus, you can (in principle) write one support library conditioned by *#ifdef*'s supplied by such an oracle. (This is largely unused in our Center, mainly because even if there were a reliable way of automatically determining the appropriate *#define*s, and there never seemed to be one that worked on our UNIX system, a single library of source festooned with *#ifdef*'ed code seemed rather less attractive than multiple copies of the same library tuned to various systems.)

Perhaps the most elaborate form of this self-configuring approach we know of is that used by some colleagues in another group at AT&T Bell Laboratories. Applications are built using a fairly extensive library common to their group. The installation process first builds this library piece by piece, probing for each routine by compiling a little test program, and either using the installed routine or some portable source. Any installation errors or problems are automatically mailed back to the owner of that piece. This, of course, takes time; it took well over 45 minutes on a VAX 11/750 to install some tree-walking software (equivalent to *ftw*)!

Note that the difference between approach 2 and approach 3 is small but important; with approach 2, the need for a separate environment eventually will go away (except, of course, for target systems which do not support ANSI C and POSIX). With approach 3, you are stuck with your support libraries forever, although with luck, you may be able to implement your support library under POSIX.

Approach 4: Just say no

There comes a point, though, where you simply cannot port programs. This can stem from an ideological or aesthetic point of view, or from a simple mismatch of system capabilities. An example of the former might be programs that depend on features peculiar to Plan 9, such as a file system configured to and shared by a process group, or by the way the environment is shared by a process group through the file system. (In principle, these could be nearly simulated, albeit with considerable effort.) An example of a mismatch might be programs that depend on the normal UNIX system semantics of linking and unlinking files. Plan 9 simply doesn't have links (of any kind). And under some circumstances, these programs might fail if the underlying file system is an NFS file system.

Some Real Examples

Here are some real examples from our center.

gre

Gre is a fast replacement for *grep*, *egrep*, and *fgrep*. It consists of a sophisticated pattern matching library, including single string, multiple string and regular expression routines, of 3349 lines of 17 C source and header files. The *gre* source, excluding this library, is 977 lines of 9 C source and header files. The program has proven to be very portable; the reasons why are

- It requires very little from the system; the most worrying ar `setjmp` and `longjmp` (although these have caused no problem so far). Actually, the routine that caused the most porting problems was `getopt`, so we now just ship it with *gre*.

- The source is designed to be compiled with either ANSI C or C++. However, by adhering to a specific style of declarations, an *awk* script converts the source into old-style C.

- One thing that varies across our computing environment is the preferred buffered I/O system. On Plan 9, we use *bio* by Ken Thompson; on 10th Edition machines, we use *fio* by Andrew Hume; and on the other machines, we use *stdio*. These are handled by a compilation flag (say `USE_STDIO`) set within the makefile and an appropriate header file:

```
#ifdef USE_STDIO
#define     PR  printf(
#define     EPR fprintf(stderr,
#define     SPR sprintf(
#define     WR(b,n) fwrite(b, 1, n, stdout)
#define     FLUSH fflush(stdout)
#endif
```

(*Gre* doesn't use buffered input; it uses the system call *read* for all its input.)

- Compilation is controlled by *mk* on our Plan 9 and 10th Edition machines, but an extremely simple makefile is provided for other systems.

- *Gre* comes with an extensive test suite of 212 simple tests and 13 complicated tests. (Every bug generated at least one test case.) The test suite is comprehensive, convenient to run and takes only 35 seconds to run on a VAX 8550.

File Motel

The File Motel [Hum88a] is an example of a larger system. It is a file backup and restore system based on a central server and multiple clients running on heterogeneous systems. It solves modestly well the main problems facing such systems: heterogeneous compilers, libraries and include files; heterogeneous networking interfaces; heterogeneous backup media; and configurability issues. To give you an idea of the size of the system, a client has a configuration file, 3 shell scripts and 15 executables; one script is in `/usr/bin` and all the other files are in `/usr/lib/backup`. The server has an additional 7 shell scripts and 16 executables, supporting amongst other things a compressed B-tree database. The source for the 10 shell scripts is 376 lines, and the C source is 5065 lines in 38 source files.

The porting strategy chosen for the File Motel is typical for large systems in our Center; the source has almost no `#ifdef`'s (when used, they control features or functions rather than alternate implementations) and instead uses system or feature specific libraries to hide system differences. Unlike *gre*, the File Motel relies upon a specific make tool *mk*, and a specific I/O library, *fio*; it was easier to port both of these than it was to deal with the variability and inefficiency of *make* and stdio. The support libraries are

library	files	lines	contents
libc	28	2158	fio, getopt, regcomp/regexp
libfm	24	1214	db interface, logging, etc
libcbt	14	2045	compressed B-tree library
sys/*sys*	9	754	(see below)

To some extent, the system specific library reflects when this system was implemented (1986 and 1989). (If this system were reimplemented today, we would assume POSIX 1003.1 compliance and several of the routines would move into `libc`.)

The routines in the system specific library are

`dateadjust`	set dates on a file
`dirtoents`	return member names from a directory
`ftw`	file tree walk routine
`nofile`	determine number of file descriptors available
`rx.`*ipc*	establish a bidirectional link to a <machine,service>
`serv.` *ipc*	service requests for a given service
`sysname`	determine the system name
`username`	determine the user's name

We decided, as POSIX has, that you can't unify current IPC interfaces; we simply cope with them. On some systems, we had to implement some other routines as well, such as *mkdir*, *rmdir*, *lstat*, and *dup2*. To support a new system, one simply copies the directory for a similar system and tweaks the implementation. This admittedly crude technique has proved manageable in practice; we have not found a bug common to the system-specific routines.

Configuration issues are controlled by two files; one is on all systems and designates the name of the central server machine and other system configuration parameters, and the other is used in compilation and is included by all the various mkfile's. Nearly all the various configuration options are selected in the latter file; however, the server machine's name is dependent on the underlying IPC mechanism. Here is an example of the compilation options file:

```
# version 10
RANLIB=ranlib
IPC=v10
IPCLIB=-lipc
SYS=v10
FMLIB=/usr/lib/backup
FMBIN=/usr/bin
LIBTYPE=a
NPROC=2
CFLAGS= -DSTRINGH="'<string.h>'"
COMPAT=.compat
WORMFACE=uda
```

The Research UNIX System

The research UNIX system has not been ported (except experimentally) to a new architecture or machine since it was ported to the VAX in the early 80s. Furthermore, it seems unlikely it ever will be; the job is large and tedious, vendors no longer routinely provide enough information to do such a job, and the people who are most likely to do it are working on Plan 9 instead. However, there is continuing interest and support for the user level programs such as *grep*, *awk*, *troff* etc. So we are leaning towards a solution which is essentially porting the user level commands and libraries (and not the operating system) to the other systems. The problem is maintaining such programs across the different machines and operating systems in our Center. The components of the solution are

APE	As described above. The compiler `lcc` is by Chris Fraser and David Hanson [Fra91a], and the ANSI C preprocessor is by Dennis Ritchie.
mk	*Mk* is a replacement for *make* that runs on all the machines in our Center. It is currently being rewritten so as to better fit within the Plan 9 environment.
dist/ship	These programs aid in the distribution of database files, source and executables across the machines in our environment. *Ship* [Koe84a] works well between the 10th Edition machines. *Dist*, written by Mike Haertel, solves the more difficult problem of handling the non-10th edition machines and Plan 9.
rc	Tom Duff's new shell is the only shell common to (nearly) all our machines. Actually, the issue of a common shell, at least where it impacts installation scripts and *mk* recipes, is one of the main unresolved issues.

Plan 9

Plan 9 is a *research* project, whose goal is to find a good way of working with modern computers, networks, displays, and storage media. A deliberate attempt was made to question each element of the programming environment, looking for better ways than the traditional UNIX way wherever possible. Software compatibility between Plan 9 and UNIX was not an important issue. So for Plan 9, the only portability question is dealing with the various machine architectures that Plan 9 runs on. This has lead to a style of programming that trades CPU costs for generality. For example, when 32 bit integers are transmitted through communication channels, a byte order, say MSB, is chosen and implemented in the obvious portable way:

```
long v;
unsigned char *p;
v = *p++ << 24;
v |= *p++ << 16;
v |= *p++ << 8;
v |= *p;
```

rather than using an `#ifdef` to determine if we can do it in some faster way for a specific system. We actively use identical source to compile all of the utilities running on MIPS, 68020, 68040, Sparc, and 386 machines.

Plan 9 has its own set of system calls, libraries, header files, compiler, shell, editor, and other utilities. There is some similarity to UNIX, but there are also differences. One difference is that each process group can have a custom namespace. This makes it easy to handle heterogeneous architectures by arranging that, for instance, `/bin` contains binaries appropriate for the architecture that the process group is running on.

What are the advantages of the Plan 9 environment over POSIX? From a programmer's perspective, the big advantage is library and header file simplicity. Most of the system calls and library functions used in normal programming are in `libc.a`, and they are all declared (with function prototypes) in `<libc.h>`. There is a small architecture-dependent file called `<u.h>` that defines types used in various header files, but otherwise Plan 9 sticks to the rule: one library, one header file. Most utilities use only `libc` and `libbio`, a new buffered I/O library. Contrast this with the situation in POSIX, where you typically need six or eight header files, and a manual to remember which ones. For instance, to use the `open` system call, you must include `<sys/types.h>`, `<sys/stat.h>`, and `<fcntl.h>`.

Of course, there are occasions when we want to port software to or from Plan 9. For importing, there is an APE library for Plan 9 that simulates most of POSIX. For exporting, we can use the 'little bit of England' approach, and provide a simulation of the Plan 9 environment assuming a POSIX one. In particular, the graphics model is much simpler than other systems and has been implemented using the X window system.

However, the simulation in both directions cannot be perfect. For example, there is no notion of an 'effective user' in Plan 9, so `seteuid` has no effect (you can't even fake it). Conversely, the Plan 9 system calls for modifying the namespace would be hard to simulate in a POSIX environment. So far, the problems in importing programs have not been due to the veracity of the POSIX simulation, but rather, with things that POSIX doesn't yet define (networking, for instance). Exporting programs is harder as the more interesting programs tend to use the novel aspects of the Plan 9 environment. For example, the Plan 9 window system, 8.5, relies heavily on how the file system namespace is built and shared by process groups and cannot be easily ported to a normal UNIX system.

Conclusion

There is no magic bullet for solving portability problems. Standards such as ANSI C and POSIX have made the chore of porting significantly easier than it used to be, and the situation will continue to improve as these standards become ubiquitous in the market place.

The approaches outlined above describe some ways that applications can be structured for portability. One part of the Plan 9 scheme that deserves special mention is the structuring of header files (an initial machine-dependent file followed by one per library). This is a far superior scheme to that in the standards (although they address rather more concerns than we do).

References

[ANS89a] ANSI, "Programming Language – C" ANSI X3.159-1989 1989

[Fra91a] Christopher W. Fraser, and David R. Hanson "A Retargetable Compiler for ANSI C" SIGPLAN 26(10) October 1991

[Hum87a] Andrew Hume, "Mk: A Successor to Make" Summer 1987 USENIX Conference Proceedings June 1987 USENIX

[Hum88a] Andrew Hume, "The File Motel – An Incremental Backup System for UNIX" Summer 1988 USENIX Conference Proceedings June 1988 USENIX

[Hum90a] Andrew Hume, and Doug McIlroy "Unix Research System (10th Edition)" 1990 Saunders College Publishing Philadelphia

[Koe84a] Andrew Koenig, "Automatic Software Distribution" Summer 1984 USENIX Summer Conference Proceedings June 1984 USENIX

[Pik90a] Rob Pike, Dave Presotto, Ken Thompson, and Howard Trickey, "Plan 9 from Bell Labs." Summer 1990 UKUUG Conference Proceedings London UKUUG July 1990

On-Line Documentation and Publication: The Davenport Group

Dale Dougherty
O'Reilly & Associates, Inc.
Sebastopol, California
USA

E-mail dale@ora.com

Dale Dougherty is writer and editor for O'Reilly & Associates, Inc. He has co-authored with Tim O'Reilly "UNIX Text Processing" (Sams/Hayden Books) and written "Sed & Awk" and other Nutshell Handbooks published by O'Reilly & Associates. He also formed the Digital Media Group within O'Reilly & Associates to focus on digital delivery of published information. He founded the Davenport Group in 1991.

Introduction

The Davenport Group brings together UNIX system vendors, software vendors and technical book publishers who have something at stake in the development of standards and technology for online documentation and publication. The initial efforts of the Davenport Group are focused upon the establishment of a common interchange format for online technical documentation. The Davenport Group supports an interchange format based on SGML, the Standard Generalised Markup Language (ISO/IEC 8879) and "HyTime" Hypermedia/Time-based Structuring Language (ISO/IEC Draft International Standard 10744).

This article discusses why the Davenport Group was formed and what problems it is attempting to solve. It describes recent activities of the group. Emergence of On-Line Documentation Systems

If you buy a UNIX workstation these days, chances are good that instead of a shelf of UNIX manuals, you'll get a single CD-ROM containing the same amount of material and more. On-line documentation systems are being introduced as an economical way for system vendors to deliver documentation to computer users. These systems are either replacing or supplementing the delivery documentation in print.

Of course, it is not only a CD-ROM that a manufacturer must supply but a software program called a browser that displays the documents on the computer screen. The browser offers different ways to navigate through bodies of information, allowing the user to place notes and bookmarks along the way. Users can follow hypertext links to jump instantly from one place in a document to another. The browser also usually features a fairly sophisticated search engine that allows users to locate documents based on particular word matches.

On-line documentation systems provide a solution to the growing volume and complexity of technical documentation. While many prefer to receive printed documentation, others are realising the benefits of having the documentation available online. They don't have to hunt through stacks of books or walk to the system administrator's office to find out that someone else has checked out the manual.

As sophisticated as these "first-generation" online documentation systems are, most are hard-wired to the structure and format of the vendor's own documentation. For instance, one vendor uses Interleaf to produce print documentation based on a specific style sheet and other conventions. Their online system accepts as input only Interleaf documents that use the same set of styles. Another vendor handles troff documents that are marked up using their own internal macro package.

The current generation of online documentation systems are not extensible, general-purpose information systems for retrieving and browsing technical information. If they were open systems, a user might go from a computer programming manual to a scientific journal to a professional reference book to a corporate memo, all online in a single software environment.

At O'Reilly & Associates, Inc., we publish books on UNIX and the X Window System that are shipped as documentation by numerous UNIX system manufacturers. As more manufacturers move to online delivery of documentation, we have begun working with manufacturers to get our books into their systems. Indeed, we think it makes a lot of sense for books about computers to be read on computers.

As a publisher, however, we knew it wasn't practical or economical to supply our books in multiple formats to a wide range of UNIX manufacturers. Yes, we could write filters to go from one format to another, or the manufacturers could write these filters. But it is impractical to maintain a large number of filters. What's more, our documents were marked up for a print-specific formatter, troff, and we needed some way of adding information for online display, such as hypertext links.

Happily, most UNIX manufacturers recognise the limitations of existing online documentations. Now they rely more and more on outside sources for computer documentation just as they use third-parties to supply hardware or software components to build a system. These outside sources include publishers, software vendors, and other manufacturers. It also includes industry consortia such as the Open Software Foundation (OSF) and UNIX International (UI) who distribute documentation in source form to their members. Those organisations had already learned from experience that it was difficult to send and receive complex documents in a print-specific format such as troff.

All of this serves as background to the formation of the Davenport Group. It was founded on the hope that by providing an open forum for discussing issues specific to online documentation and publication, UNIX manufacturers, software vendors and publishers might find ways to solve problems that we had in common. (As an aside, the name Davenport has no particular significance -- one of us liked the name, that's all -- and we all agreed upon it instantly, providing a handle for identifying the group.)

After several meetings, we agreed that we would focus our efforts on establishing a common interchange format for online documentation in source form. That would allow publishers to put information in a single format for distribution, and it would make it easier for UNIX manufacturers to import other documents into their systems. We agreed that the interchange format would be based on SGML, the Standard Generalised Markup Language (ISO/IEC 8879).

SGML differs significantly from a print-specific markup language such troff or TeX. SGML is concerned with the semantic structure of a document, not its typographic appearance. That structure is described as a formal grammar in a Document Type Definition (DTD). The DTD describes the names of individual structural elements and their relationships to one another. For instance, a DTD might define an element for first-level headings named "HEAD1" and state that it contains one other element named "TITLE." In a document, first-level headings are identified by the tag "HEAD1." An SGML parser can validate any SGML document against its DTD and report, for example, that a document has a HEAD1 element but is missing a TITLE element.

In a print-specific format such as troff, a macro usually marks semantic elements such as a first level heading, although there is no mechanism such as a DTD for describing the document's structure. In addition, formatter-specific markup is mixed with semantic markup, specifying indent levels, page breaks and vertical spacing. We also agreed that we would use "HyTime" Hypermedia/Time-based Structuring Language (ISO/IEC Draft International Standard 10744) to define links in a standard way. It was important to many of us that if were going to go to great expense to add linking information to our documents, that it be done in a portable way.

An interchange format makes it possible to represent all the value of the information in a way that is independent of any particular software application or interface. This could be a very significant result, one that may have impact outside the UNIX community.

Most publishers realise that the information published in books is also valuable in digital form. The difficulty is delivering the information in a form that can reach a wide audience. As mentioned earlier, software is required to provide access to the information and display it on a screen. Most CD-ROM products today bind the information to a specific piece of software. The user who buys an encyclopedia on CD-ROM gets the software "for nothing." This may seem like a bargain, but it is not. For one thing, the software is usable only for that particular body of information. You can't integrate new sources of information. If you buy a dictionary on CD-ROM, you will likely get a different piece of software, good only for use in reading that dictionary. When a body of information is bound to a particular software, the user must learn a unique interface to access the information. If the user has to deal with multiple sources of information (a manual, a journal, a dictionary), then it can only be confusing and distracting to change among different programs to perform the same tasks on different bodies of information. I believe it seriously constrains the market for online information products.

If there were a common interchange format, then one can imagine that a browser might be re-usable. Indeed, consumers, not producers, of information could choose the software and interface they want to use. They could maintain a consistent software environment independent of the information that they access. One can even imagine that an online documentation system, supplied by a computer manufacturer, might be open to all kinds of information, including that developed by users of such systems.

Recent Activities of the Davenport Group

At our December meeting following Hypertext '91 in San Antonio, we established a working group to examine the application of architectural forms in SGML to problems of interchange in technical documentation. This working group has met in January and March for four days. Both meetings have been quite focused and technical and the people who attended were, by and large, selected to represent their company's interest in this particular area.

Thus, we seem to have come upon a two-tier meeting structure that other organisations have used successfully: general or plenary meetings and independent working groups. With this structure, general meetings will be held less frequently, perhaps twice a year. At the general meeting, we entertain proposals for new working groups and grant or deny their charter. In addition, we receive reports on the activity of the existing groups and comment upon their progress.

A working group can be established to provide a set of interested individuals a forum at which they can propose, discuss and review work in a particular subject area. The structure of the group, the frequency and duration of its meetings and, indeed, its work product, can be determined by the group's participants. Each working group should be able to operate autonomously, without outside interference or bureaucracy. A working group offers a chance to draw upon specific expertise and to increase the level of participation of people of similar interests.

The only requirement is that a group have a champion, someone who makes a proposal at the general meeting to describe the objectives of the group and who is willing to organise the effort. A group exists as long as it has contributors, which is to say that the work must be valuable enough for others to want to participate in it.

There are many possibilities for working groups: converting from troff to SGML, standardising man pages, formalising linking strategies, user testing of online technologies, studying similarities and differences of multiple DTDs, licensing and copyright issues, etc.

Each working group need not have a technical focus. One might be educational, sponsoring a seminar on a particular area. Another might demonstrate the application of particular standards or technology, along the lines of a "proof-of-concept" demonstration. Similarly, one might develop a set of criteria for analysing or testing related commercial products or applications, and produce a summary such as Craig Boyle's group at Texas A&M produced on hypertext browsers. A working group might study and review an existing or emerging standard that has an impact on online documentation systems. We have planned a general meeting for 11-13 May 1992 in Seattle, WA, USA, hosted by Digital Equipment Corp. At that meeting, we will discuss proposals for establishing new working groups.

Working Group on Architectural Forms

In January, 1992, the working group on architectural forms met in Emeryville, CA, USA, hosted by Open Systems Solutions, Inc. A second meeting was held in March, 1992 at Ft. Collins, CO, USA, hosted by Hewlett-Packard. One or both of these sessions were attended by representatives from Avalanche, Electronic Book Technologies, Hewlett-Packard, IBM, Novell, O'Reilly & Associates, Open Software Foundation, Open Systems Solutions, SCO, Silicon Graphics, Sirius Technologies, TechnoTeacher and UNIX System Laboratories.

This working group is designing a template for an SGML Document Type Definition (DTD) that document architects can use when developing a DTD for technical documentation. The template offers a set of guidelines for declaring elements in a DTD such that the information in the document can be made interoperable among various applications. The SGML DTD template is described in the Davenport Group Draft Advisory Standard (DGDAS). Its editor is Dr. Steven Newcomb of Florida State University and co-chairman of the committee developing HyTime.

- DGDAS Excerpt:

"The DGDAS defines a model and language for the representation of online and printed technical documentation of the kind usually supplied by UNIX and other open systems vendors to their customers. The DGDAS will allow such vendors to "bundle" technical documents provided by a variety of publishers, and to give their customers access to all this information via user interfaces that they may define independently. The DGDAS will allow documents of dissimilar SGML document types to be delivered as a single-compound document with whatever aggregated indexes, tables of contents, cross-references, and webs of hyperlinks the systems integrators may wish to generate, without any special assistance from the originators and/or publishers. Document originators and publishers will be able to create and maintain a single version of each document without regard to the modes and styles of its ultimate delivery to users, since the DGDAS makes it possible for the same original document to be formatted for online delivery as hypermedia as well as for printing on paper."

The Davenport Group Draft Advisory Standard is available via anonymous FTP as a compressed PostScript file on the following hosts: mailer.cc.fsu.edu (pub/SGML/DAVENPORT/dvnport1.ps.Z); ifi.uio.no (SGML/davenport.dgdas.1.ps.Z); src.doc.ic.ac.uk (/text/sgml/DAVENPORT/dvnport.1.ps.Z). By the time this article appears, a new version of this document may be available at the same locations.)

- Organisation:

The meetings and workshops of the Davenport Group are open to all interested participants and observers. For more information, write to Dale Dougherty, The Davenport Group, c/o O'Reilly & Associates, Inc., 103A Morris Avenue, Sebastopol, CA 95472, USA. You can also send e-mail at dale@ora.com. There is an e-mail and print mailing list.

The Steering Committee of the Davenport Group consists of Dale Dougherty, O'Reilly & Associates; H. Garrett Long, Managing Editor of UNIX Press, the technical publishing arm of UNIX System Laboratories (USL); Paul Hammerstrom, the Publications Manager for the UNIX System Software Group (USSG) at Digital Equipment Corporation; Rich Pusateri, Project Lead of the On-line Information Access System (OLIAS) technology development at HaL Computer.

Representatives from the following companies have participated in the Davenport Group:

- Avalanche Development
- Convex
- Cray Research
- Data General
- Digital Equipment Corporation

- Digital Press
- Electronic Book Technologies
- Exoterica
- Frame
- Fulcrum
- HaL Computer Systems
- Hewlett-Packard
- Hypertext Engineering
- IBM
- Information Technology Group, Texas A&M University
- Interactive Systems
- Interleaf
- Knowledge Systems
- Knowledge Workshop
- O'Reilly & Associates
- Open Software Foundation (OSF)
- Open Systems Solutions
- Open Text Systems
- Prentice Hall
- Prime Computer
- Santa Cruz Operations (SCO)
- Silicon Graphics
- Sirius
- Sun Microsystems
- Technical Arts Group
- TechnoTeacher
- Tivoli
- UNIX System Laboratories (USL)
- Usenix
- Verity

Book Review

Programmer's Supplement for Release 5, of the X Window System Version 11.
David Flanagan, O'Reilly & Associates, Inc., November 1991,
ISBN 0-937175-86-2.
(UK) Price ??.??, Soft Back, 367 pp,

Reviewed by Loek Schoenmaker, Vrije Universiteit, Amsterdam.

This book addresses the differences between release 4 and release 5 of the X11 Window System. It describes in detail the new features available in release 5.

The book is written for X (application) programmers. The author assumes (as stated in the preface) that the reader is somewhat familiar with Xlib and Xt programming in the C programming language: it is not a stand-alone book, it is a supplement to other manuals.

The structure of the book is very good. In addition to a clear introduction, each chapter starts out with a summary of the chapter and an indication for whom the chapter is meant. Furthermore, the book contains quite a few sample programs, illustrating the use of the described functions.

What is described in the book? Well, it describes the new features of release 5 which include:

- Font Service and Scalable Fonts

In release 5 it is possible to introduce a font server: all fonts are available from the font server and every client (and printer) uses the font server. The advantage is, that the fonts need not be duplicated for each server so many more fonts can be supported without running out of disk-space. Furthermore, scaled fonts can be computed without affecting the performance of the X server.

- Device-independent colour

Until release 5, X has been intrinsically device-dependent. Now, the X Colour Management System (Xcms) has been introduced to support the description of device-independent colours in a variety of colour spaces. In this book Xcms is only partly described, leaving the discussion of the more difficult part to a forthcoming book.

- Internationalisation

Internationalisation is about writing software that can be run, unmodified, in any language desired by the user. This release tries to solve the problems that arises if one wants to represent a language with more than 256 characters, if one wants to input ideographic text from a standard keyboard.

This part of the book is divided in (1) an overview of the issues and problems of internationalisation and (2) an overview of X11R5's possibilities of internationalised text input.

- Resource Management

Some changes have been made to the Xlib and Xt resource management. The book describes the changes.

Furthermore, some other changes are described and a large part of the book is devoted to reference pages of most of the new functions.

The only error I discovered was in the preface. The specified ftp address for the retrieval of example programs, results in an error message indicating that the ftp-address is out of date (and the new address).

The book is very clearly written and well-structured. Apart from who should read what, it is also very clear what is described of a subject and what is not. In the latter case, references are given to books describing the missing topics.

Bearing this all in mind it is a good book: if you are familiar with X programming it'll be easy to use. If you are not, it will almost be worthless because of the assumptions made in the book.

My final remark: I like the book, especially since it is printed on recycled paper.

Jersey Conference Report

Jacob Hallen
Chalmers University
Goteborg
Sweden

E-mail jacob@fy.chalmers.se

Jacob Hallen is a somewhat overaged student at the Chalmers University of technology in Goteborg, Sweden. He started out as a student of Physics, but quickly discovered that Computer Science was a lot more fun. He is severely hampered in his studies by teaching, developing irrigation systems, scuba diving, mud and the necessity to eat.

Conference in the Channel

As I am sure most of you know, the Spring Europen Conference was held on Jersey. At these events there are a lot more things happening than what you can read in the conference proceedings. This is an account of my impressions from the conference.

Minus Two

Airplane fares are a jungle. By leaving home two days before the conference and spending one day in London, I save over 250 pounds. Since there are worse things to do on a Saturday than drooling over the computer science bookshelves in Dillons bookstore and getting lost in the Egyptian part of British Museum, I have no complaints about the extra days.

In the afternoon a one hour subway ride and 45 minutes on an airplane brings me to Jersey. One can see the entire island as the plane goes down for landing. The island is quite small, about twice as long as it is wide, rising above the waves as a flat table. It looks very green and inviting in the afternoon sun.

On the bus into St. Helier I met the first delegates to the conference, Anette Bradefors and Bernt Budde from the computer centre at Uppsala university in Sweden.. I also met an Irishman, who like me needed a place to stay. We decided to go looking together, so after turning one rather scruffy bed & breakfast place down and asking the local people in a pub, we ended up in a very nice nice little hotel near the beach.

Minus One

My Irish friend, who came to Jersey on vacation and to look for a job, reported that he had met a very nice Jersey girl the night before, and that she might come to pick him up at the hotel. I decided to hire a bike and tour the island in the nice and sunny weather. Going on the left side of the road felt rather awkward in the beginning, but after a while I got the hang of that too. I decided to go counter clockwise all around the island. This turned out to be a bad decision, because I ended up going slightly uphill almost all the day with a short and steep slope at the end of the ride.

Jersey turned out to be be filled with small villages, farmhouses and cottages with a few of the famous mansions (or should I say palaces) in between. I was quite surprised to see that potatoes were almost the only crop being grown on the island. Even more amazing was that the southern slopes were less than a month away from picking.

My tour led me past several ancient watchtowers that protected Jersey in times of unrest to the village of Gorey, which is dominated by the magnificent medieval castle Mont Orgeuil. The castle itself is very interesting and the view from the castle over the east coast of Jersey, the sea and the French coast less than 20 km away is really splendid.

After Gorey I soon left the sea and began the slow climb I mentioned before. The trip went on narrow roads between high earth banks and hedges through all the eleven parishes of the island. To give you a clue about how small Jersey is, I spent four hours effective time at a leisurely tempo on the bicycle. I must confess that I'm not used to such large quantities of fresh air, and it was with a sigh of contentment that I sat down with a pint of bitter in the hotel pub.

Later that night I went to a Portuguese restaurant to have dinner. During the meal I noticed a party of six people who were in very high spirits. Since two of the men in the party had beards and the other two wore moustaches I decided that they simply had to be conference delegates. When I approached them, they invited me to their table and introduced themselves as Jim Reid, Mick Farmer, Sue Small, Hazel Tomlinson, Sunil Das and Jan (whose last name escapes me at the moment). They all turned out to be from the UKUUG and the rest of the evening was spent discussing

everything from wine temperatures to Glaswegian dialect.

Takeoff

Monday morning saw the start of the tutorials, and I went to Mr Stallman's tutorial on gcc 2.0. The lecture covered lots of ground and quite a bit of the stuff went over my head. I think you need to know quite a bit about the internals of gcc before going to a tutorial like this. At the tutorial there were mainly people from central Europe, like Antal Komar from Hungary, Jurgen Schmidt from Germany and Michel Munnix from Belgium.

During lunch and the coffee breaks I managed to have a look at the Hotel de France, where the conference was situated. It is a tall white building that looks like it was built in the early years of the 20th century. On the ground floor it has a plethora of banquet halls, dining rooms, conference rooms and a huge bar, all with 5 meter high ceilings and deep wall to wall carpets. The hotel was staffed with a multitude of porters, waiters and servicemen. It is said to be the poshest hotel on Jersey.

Monday was also the day when we were told that the machine that was to serve as mail-machine and IP-link had been lost in the air freight. This was received with calm and even small sighs of relief by some people, while others were seen walking the corridors with worried faces and muttering 'I should never have come here'.

In the evening I sampled the traditional British cuisine at my hotel (with soggy boiled vegetables and all) and made an early night.

My Irish friend reported that he had gone sightseeing with his Jersey lass, and that he was invited to dinner with her parents the next night.

StageTwo

When I went to the tutorial on Tuesday morning there was a drizzle and temperatures were not much above freezing, so it felt nice and cosy to go to Berry Kerchevals tutorial on X11 in a windowless conference room. This lecture was at just the right level for me, and I left it with a feeling that I had grasped in a half day something that would have taken me a week or more to find out by myself.

During the morning coffee break I had my face saved by the UKUUG Facesaver project, who were operating just by the registration desk. The resulting pictures are very compact and can be downloaded by FTP from somewhere in Britain, but the procedure for saving a picture seemed much too complicated to me. A shellscript to do the routine tasks would probably do wonders.

After a splendid lunch in the green-house-like hotel restaurant it was time for the conference proceedings to start. The conference hall could seat about 300 people and I estimated it to be about half filled at the start of the conference.

The main topic of the conference was portability and keynote speaker Mike Banahan took us back to his days with the BBC, when a piece of equipment was portable if it had a handle and to the good old PDP11/ Unix V7 days when portability wasn't a problem. He was followed by a number of interesting speakers, whose presentations are very well covered in the conference proceedings. Most papers were very well presented, but some people would be much better received if they learned how to use a microphone, and asked an assistant to operate the overhead projector.

Among the conference attendants you could see people from almost all the Europen countries, from Iceland to Tunisia and from Portugal to Russia. There were also a number of American speakers present. Ed Gould and Andrew Hume probably arrived the same morning from a much warmer climate, because they wore shorts in the rather chilly Jersey afternoon. Personally I used double sweaters before the cold conference hall was properly warmed up.

The evening took me to a small Italian restaurant that served huge pasta dishes. I had a very good hot seafood spaghetti and then I went back to Hotel de France to chat with people. To my surprise the bar was almost empty so some other place must have been the centre of activities this night, or perhaps everyone went to bed at ten o'clock.

Back at my hotel I learned that it was fully booked the next night by a French school class and that my Irish friend was to stay in the girls home, and that her father was offering him a job in the brewery.

Stage Three

Wednesday morning I changed to a small and rather scruffy B&B quite near the conference hotel. With the exception of three speakers the entire day was spent on portability. I was rather surprised at the crude methods that are state of the art on this field. The only software tool that was presented or even mentioned in a presentation was a tool to check for standards compliance that was presented by Derek Jones of Knowledge Software. The three speeches not covering portability were a rather complicated presentation on security and encryption, a rather frightening presentation on software patents and a rather heartening presentation on the state of the EUnet.

After the programme a rather large crowd left for an EUnet BOF in the bar, while a smaller crowd stayed behind to discuss patents and copyrights with Mr Stallman. He was rather disheartened by the small number of people who stayed, but after a bit of persuasion we got an interesting discussion under way. I think several of the people who stayed were concerned enough to do something, and that is what really counts.

Torpedoes Away

Wednesday evening was the time for the conference banquet and when the buses left the hotel for La Mare vineyards they were full of expectant buzzing. Style varied from tennis shirt and sneakers to tuxedo, with about the same number of people at both extremes, so everyone was rightly dressed for the occasion, and I think no one felt uncomfortable in the company.

After a 15 minute drive straight across the island we arrived at the vineyards and were shown into a large white tent. A band played in the corner and we were welcomed with a drink of sparkling white wine. In the tent we were seated at large round tables with about 8 people at each.

After this the proprietor held a little speech on the subject of wine tasting, and a blind test of 6 wines began. Each table was supposed to choose between 3 different countries, 3 different districts and 3 different grapes for each wine and finally guess which year the wine was made. Our table turned out to contain very little in the way of wine knowledge, but Berry Kercheval turned out to know more than he confessed to begin with. The contest was quite difficult, and the wines served were of really good quality. I was quite ashamed to mistake a Mosel Spatlese for a Sauternes, but then, I wasn't the only one. Among the other wines in the tasting there were a nice Rioja, a very good Australian claret and of course the very nice white wine that La Mare produces. The tasting provoked much discussion and was a really fun event.

After the wine tasting we were served a very nice cold buffet. By then spirits were very high and it was more or less impossible to converse with other people than my two nearest table mates; Donald Lewine from the USA and Joakim Gezelius from Sweden.

Before the desert our host managed after a few tries to make his voice heard and presented the winners of the competition. Mike Banahan headed the winning table with an entire table of Norwegians in second place. I think I noticed several French people at Mikes table, which may explain their remarkable score. At our table we were very happy that we managed to do quite a bit better than pure random guessing.

After the prize ceremony the chairmen of the days sessions, Berry Shein and David Tilbrook, were thanked for their participation, and were given sweatshirts as presents. Due to certain size incompatibilities they were offered 2 and 3 shirts respectively. There was also an ad from Richard Stallman that he handed out LPF buttons (Apple logo in the form of a snake and the text 'Keep your lawyers off my computer'). There was a steady stream of visitors at his table and when the evening ended a majority of the participants wore a button.

The rest of the night passed on very quickly, and when I had finished my second helping of the dessert it was already eleven o'clock and time to go. I still haven't grasped how the evening managed to pass so quickly.

On the bus back to St. Helier there was much talk and one could easily tell from the general atmosphere that it was a bus full of very contented people. I think I heard people singing in chorus on one of the other buses, but that may just be products of a too vivid imagination. Jersey laws set the closing time for pubs at eleven o'clock with no drinking up time, so when we were back in town the evening was more or less finished.

Landing

Thursday turned out to be a bit hectic. If I wanted to get to Goteborg the same day I had to take a flight at 1330, so I missed the last speakers. However, I managed to hear an interesting talk by Andrew Hume on portability in the Plan 9 environment, an overview of X by Berry Kercheval, and a presentation on programming fashion that turned out to contain some very interesting and novel ideas for how to make programming for GUI environments easier as well as more portable.

After that it was time to rush, and I shared a cab to the airport with Per Wilfor from Umea in Sweden. He was called home in a hurry, because the UNIX system he manages had crashed. Looks like we still have a long way to go until the computer industry becomes really mature.

Debriefing

All in all I think that the Jersey conference was a roaring success, and we should all give our special thanks to Bill Barrett and Jane Morrison who took care of all the practical arrangements with calm and kindness. The only cloud in the sky was the lack of IP connection, which was a problem for some of the participants.

SUUG Workshop on UNIX and its Applications – Report

Mick Farmer
Birkbeck College
Malet Street
London WC1E 7HX
England

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 interests he can be found teaching at Birkbeck College.

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

Abstract

This is a short report of my trip to Moscow to attend the above workshop and present a paper entitled "Initial Experience with the UKUUG FaceSaver Project". All views expressed are my own.

Sunday, 24 November

Air flights from London to Helsinki and from Helsinki to Moscow are cheaper, but take longer, than flying direct. Never mind, we arrive at 21:00 to be met at the gate by our host, Peter Brusilovsky. He's expecting Mike Fadden (UNIX Today) on the same flight so we went in search of a drink. Can't find a bar so we change 25 GB pounds into 2000+ roubles, remembering to keep the form for showing when we leave. Small counter at one end of the grey arrival lounge turns out to be a bar. Most people buy a vile-coloured juice but we pay 25 roubles (= 30 pence) for imported lager.

By 23:00 we convince Peter that there's nobody left to come through the gates so we leave for the hotel. Our driver is a maniac who drives along the middle lane of three (each way) overtaking on the left or right depending on how he's feeling, mostly involving changing down in order to hear the sound of the engine more clearly. We get to see the Kremlin walls as he swerves onto the Moscow bridge on two/three wheels.

The Nitsevt Hotel turns out to be a block of flats with the ground floor converted into a lobby and buffet. Our room on the top floor is clean and consists of two rooms plus kitchen (no utensils), bathroom, and toilet. There appears to be hot water and the towels consist of strips of linen; no soap.

Monday

The breakfast on offer is pretty limited – frankfurters, fried eggs, tea or coffee. There's been no milk for weeks. Still it only costs two roubles (= 2.5 pence). Norman Hull had arrived the day before as had the Algerian delegation we first met in Budapest. Two of the American visitors proclaim that they can't possibly eat "this stuff". Another American family from Minnesota complain that they've been ripped off by the Russians (something to do with the fact that they paid for their accommodation in advance in dollars).

The conference starts slowly, allowing all day for registration. We may need it as the bus which is meant to take us to the conference venue hasn't turned up. I finally ring Peter who tells me that the bus has disappeared and we should travel by metro. The metro is excellent – fast and frequent and you can go anywhere for 15 copeks (= 3/16 penny). We detour through one of their amazing stations to take photos and wonder why the Northern line of the London Underground can't be like this.

It's a good half-mile walk from the nearest metro to the conference venue so we're looking forward to a hot drink. No. There's no coffee and the tea has been delayed. Never mind, the Institute is warm and the lecture theatre looks impressive. Peter tells us that there's more interest in this workshop than in their previous meeting (just after Budapest) so there's nearly 500 people expected. The only thing on sale is more vile-coloured juice so we hold out until lunch talking to old friends. Lunch at 14:00 consists of two, smali, helpings. However, here we can buy coffee with ice cream to put in it to make it milky! The Americans from

Minnesota never stop talking about themselves which is driving most of us up the wall.

The SUUG have arranged a reception for their visitors this evening so we arrange to meet on the platform of one of the stations at 19:00. We go looking for water and possibly some biscuits in case we get hungry. This is not an easy task. First, the shops don't have signs. Second, they all have curtains across the windows so you can't see in. Third, most are closed. Fourth, those without queues don't have anything to sell! The odd packet of tea, some cooking oil, some very tired looking apples if you're lucky. I've never been anywhere where I'm a millionaire (in their currency) but can't buy anything. Nothing. No wonder Moscow seems crowded with people going from A to B looking glum – there's nothing else to do! They also have a habit of walking straight through you.

The SUUG reception is in the flat of one of the SUUG secretariat and they've laid on sandwiches, vodka, and some white wine. This must have cost them a small fortune. The Minnesota Americans haven't turned up – good. The flat is large because it's actually two flats knocked into one. Previously, there were three families living in it (our family plus each of their parents) which must have been cramped in three medium-sized living rooms, kitchen, bathroom, and toilet. All flats are owned by the state. Apparently you can only move into Moscow if you can arrange a swap with someone who wants to move out. Very few people do. It's very difficult for young couples to obtain a flat of their own and most live with one of their parents, taking over the flat when they die.

Norman and I gravitate to the kitchen drinking vodka with Sergei Kuznetsov and Leo Tomberg, the Russian network people, who are worried about the dominant position of a company called Relcom who provide most of the computer networking links in the USSR. Interestingly, they are expanding into general telecommunications. It's a bit like ICL taking on BT and Mercury. Apparently Relcom were very helpful during the recent coup and now have a friend reasonably high up. They can obtain phone lines within days whereas most Russians wait years! Nothing gets resolved and we leave when the vodka runs out.

Tuesday

They've found the bus so it's ready to take us to the conference. On the way, our guide shows us a few of the sights, mainly in the middle of huge traffic jams. Nothing starts on time so it doesn't matter that we arrive late. Peter tells me that they're unable to find a 35m slide projector so I'll have to rely on the overheads. Luckily I've also brought some snaps of the FaceSaver kit. They're arranging simultaneous translation of the English talks so I go through some of my jargon with my translator. Porting software is a new concept, so is PostScript and I'm beginning to wonder whether I've got the right guy.

Norman kicks off explaining the advantages of EurOpen, most of which aren't to this audience as only a few have e-mail and the rest have stand-alone machines with card readers and line-printers. Mike Nowlan then described EUnet in detail. This was followed by a Czech computer scientist explaining that their government had decreed that only UNIX systems would be used in future, etc. Time for the tea break, but the tea failed to materialise.

Now my talk. The simultaneous translation consisted of me saying a sentence, or nearly a sentence, followed by my translator's version in Russian. First, this doubles the time of the talk. Second, it ruins my rather dry sense of humour which relies to a fair extent on timing. However, there were a few laughs and they appreciated seeing pictures of Dennis Ritchie, Rob Pike, and the like which, apparently, they hadn't seen before.

The next talk was by Daniel Berry, an Israeli with a speech impediment, so I excused myself with many more of the delegates. Lots of interest from the delegates outside the lecture theatre in obtaining information from the west. I've also got leaflets advertising our UNIX video course which some people got excited about, but they can't afford to pay for things in hard currency.

The afternoon was all in Russian so we went looking in the shops – I don't know why because they were mostly empty. One exception was the Arbat Irish Shop which only accepted hard currency. It also had a bar so we paid for some Guinness in sterling, receiving our change in dollars. Someone's making a tidy profit there.

Moscow natives can obviously tell we're Westerners (perhaps it's my deerstalker hat) so we're constantly pestered by young boys offering roubles (over 100 to the pound), Russian dolls, Russian hats, army watches, and metal badges (over 100 for a dollar).

This evening we're trying one of the recommended restaurants. As we approach you tell something's funny because there are twelve men inside the door in dinner jackets wearing hard expressions. Still, in we go. Apparently a restaurant in Moscow is a place where you go to eat and watch a floor show. No, we don't want to buy tickets for the show, just eat. Ok, but we have to pay in hard currency. The food was passable (especially after the conference lunch) and the only drink is vodka with Russian champagne. They just come together for about five pounds. As it turns out we stay long enough to see the show, have a dance, and get drunk on the vodka!

Wednesday

After frankfurters and tea it's sight seeing day. We do the normal tourist round, Red Square, Lenin's tomb, St Basil's cathedral, the Kremlin, and GUM department store. I knock down the price of ten postcards from 70 roubles to 50 roubles (= 60 pence) only to be told later they sell for three roubles in the shops (but which shops I retaliate!). Naturally, there's nowhere to have lunch and we end up buying two cans of lager from a small kiosk which also sold combs and second-hand clothes.

We arrive back at the hotel ready for our evening entertainment – a trip to the circus. Our guide says that we can't buy food at the circus so we get some more frankfurters from the buffet and eat them in the coach. I'm beginning to loath Russian frankfurters. The circus is great fun and we buy ice cream for a couple of roubles. Sliced sausage and glasses of that vile-coloured liquid are on sale!

Thursday

Over frankfurters and tea we learn that the Minnesota Americans have been up half the night getting their networking demonstration wired up. It consists of thick-wire ethernet, thin-wire ethernet, and twisted-pair ethernet connecting some machines. They sound as if they'd invented the whole networking scene single-handed.

At the conference we talk to a lot of people while waiting for the Minnesota American's talk. He starts with lots of dense maps showing connectivity, reliability, availability, which probably doesn't mean much to an audience that doesn't even have decent e-mail. During and after tea, they give their demonstration which is a disaster. The Russians in charge of the computers don't know about ethernet addressing and a whole hour is spent screaming at one another while plugging and unplugging ethernet transceivers. Still, the Russians seem generally interested in looking at all the kit and cables.

After lunch we get into conversation with the people selling the Amoeba micro-kernel who tell us about a Spanish restaurant in the Moscow Hotel (hard currency naturally). The talks are in Russian so we look for presents; the dolls are ridiculously expensive so we decide to get a decent drink at the Arbat Irish Shop – thwarted because it's closed for some reason. However, we get to the Spanish restaurant and have a reasonable evening meal with the Amoeba people (they've brought along their own translator for the week – impressive).

Friday

This is a free day so we visit the famous cemetery and monastery and take pictures of a few graves. We know it's Chekov because we follow a guided tour and then interpret the cyrillic afterwards. By now it's getting cold and wet so we admit defeat (in finding anything to eat or drink) and go to the bar in the Arbat Irish Shop and during Guinness and eat a toasted sandwich!

One of our Russian friends has booked us a table at an Armenian restaurant and we have to phone Sergei about the time and venue. Local calls within Moscow are free. You can see why – it takes five or six tries just to get through. Anyway, we do so in the end. Apparently the restaurant will accept roubles for the food but only hard currency for drinks, and we've now run out of hard currency. We go to the Intourist Hotel to get some money. No way sunshine; the Intourist desk no longer changes money (in retrospect, this was probably because of the imminent change to a market economy when prices were expected to rocket). The bar at the Intourist has run out of beer so we spend our last dollars on coffee. The bar staff make the Krays positively helpful by comparison.

Outside we negotiate with a black market cab driver to take us to this restaurant, which is way out west on the outskirts of town. I think I've done well to beat him down from 2000 roubles to 500 roubles, but the laughs from the other drivers probably means we've been had. At this point the electricity supply to the trolley buses failed so the taxi driver drove some 200 yards up the road on the pavement and then down a one-way street the wrong way. The policeman who stopped him at the end turned out to be a friend! The taxi driver was more of a maniac driver than the one we'd had on Sunday, overtaking through small gaps, inventing a new traffic lane at the lights. He finally finds the restaurant and curses that we can't pay in anything but roubles.

Sergei was waiting for us and he agreed that we'd paid too much for the taxi. He'd come by metro and trolley-bus for 30 copeks! The meal consists of some cold starters followed by chicken kebabs of dubious quality. We can't pay for alcohol so we drink the vile-coloured juice which isn't that bad. 450 roubles for three (= 6 pounds) seems good value. If only we'd found such places sooner.

Saturday

Our original driver takes us to the airport. On the way he points out the bus that was used to bar the way of the tanks in the recent coup. Already, it's in the forecourt of one of the museums. Peter meets us at the airport where he gives us letters for us to post outside Russia.

The rate of exchange in the duty free shop is approximately three roubles to the pound! We don't buy much and end up paying in DMs through Visa.

On the way back it's so nice to be able to get currency at Helsinki airport and buy a drink!

Summary

On the negative side, I've never been to a place that made me so depressed and looking forward to getting away. Moscow can be summed up in one word – grey. The weather is grey. The buildings are grey. The people are grey. The economy is grey.

On the positive side, we've met some interesting people who work under extremely difficult conditions. We look forward to meeting our Russian friends again.

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 Alain Williams at *addw@phcomp.co.uk* 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

June 2	DKUUG C, System Development, 4GL, CASE, *Odense, Denmark*
June 3	DKUUG C, Networks & Communication, *Odense, Denmark*
June 2-4	UNIX EXPO West, *Anaheim CC, Anaheim, California, USA*
June 8-12	USENIX, *Marriott, San Antonio, Texas, USA*
June 18	DKUUG C, Multimedia, *København, Denmark*
June 21-24	Sun Users Group, *Washington DC, USA*
July	PCI, *Boston, USA*
July 3	UKUUG, LISA, *Imperial College, London, UK*
July 13-17	IEEE 1003, *Chicago, Illinois, USA (location tentative)*
August 10-13	USENIX C++, *Portland, USA*
August 18	DKUUG C, The UNIX market, *Helsingør, Denmark'*
September 8-11	AUUG C T, *World Congress Centre, Melbourne, Australia*
September 14-17	USENIX, C, Security, *Baltimore, Maryland, USA*
September 24	DKUUG C, Office Automation, *København, Denmark'*
September 22-24	GUUG, C, *Wiesbaden, Germany*
Autumn	USENIX Security III, *Location unknown*
Autumn	ISO/IEC JTC1 SC22 WG15, *Denmark*
Autumn	NUUG, C, *Norway*
Autumn	SUUG, C, *Soviet Union*
October 5-9	NLUUG C, Efficiency fair, *RAI Amsterdam, The Netherlands*
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*
October 29	DKUUG C, Workstations, *Odense, Denmark*
November 25-27	EurOpen/UniForum, *Utrecht, Netherlands*
Non 26	DKUUG C, Annuam Meeting, *København, Denmark'*
December	TNUUG, C, *Tunisia*
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)*
May 3-7	EurOpen, *Seville, Spain*
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, US*

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
AUUG	The Australian UNIX systems Users Group
DECUS	The Digital Equipment Computer Users Society
ECUG	The European C++ User Group
EXUG	European X User Group
FNUG	Federation of NCR User Groups
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
NSF	National Science Foundation
SAB	Standards Activities Board
SERC	NSF/Purdue/Forida Software Engineering Research Center
Sinix	The Singapore UNIX Association
USENIX	The Professional and Technical UNIX Association
UniForum	The International Association of UNIX Systems Users

TuNisan UNIX User Group - TNUUG

Mondher Makni
IRSIT
BP 212, 2 Rue Ibn Nadime
1082 Cite Mahrajane
Tunis, Tunisia

E-mail Mondher@Tunisia.EU.net

Mondher Makni is the Chairman of TNUUG. He is also the Administrative manager of the EUnet TNnet, the Tunisian branch of EUnet, in Tunisia. Mondher Makni works at the Regional Institute for Informatics and Telecommunications as the Network Manager. He holds a Master of Sciences Degree in Computer Engineering from Rensselaer Polytechnic Institute (RPI) in Troy, New York, USA.

The TNUUG was created in March 1991. TNUUG was voted a member of EurOpen in the Tromsø conference in May 1991. Since then the initial effort of TNUUG has been focused in setting up electronic mail service and full IP connectivity to the rest of the world. Since July 1991. Tunisia has had full IP connectivity worldwide offering standard electronic services (email, Telnet and FTP). The next step is to develop a TNUUG customer base. Currently, TNUUG has six full members (fee subscription) in addition to five temporary (no fee subscription) members. The membership categories are:

- Private institutions
- Government institutions
- University
- Individuals

TNUUG is currently housed by the Regional Institute for Informatics and Telecommunications (IRSIT).

Objectives of TNUUG

- Promote Open System concepts within the computer user community in Tunisia
- Promote UNIX as an Open System at the user level and at the Computer Vendor level
- Demonstrate the Internationalisation components of UNIX mainly for the Arabic Language support
- Promote networking activities within Tunisia

TNUUG Activities

In this regard, a one day UNIX Openhouse took place on 10 December 1991. This open house was sponsored jointly by a local SCO vendor and TMI a local Systems Integrator. About 150 visitors from differents user classes (end users, System Integrators, Computer vendors, etc...) participated at this Openhouse. Several papers were presented about various topics (UNIX & Networks, Distributed Databases in UNIX, UNIX and Security, UNIX and Normalisation, etc...) as well as a UNIX equipment exhibition. During this Openhouse, the official membership of TNUUG with EurOpen was announced. In addition, the official opening of the Tunisia Internet node (TNnet) was announced.

Based on the participants comments and reactions, It was felt that a more detailed discussion of some topics about Open Systems and UNIX was necessary. In this regard, a three day seminar about UNIX and Open systems is planned in December 1992. This seminar will be organised by TNUUG.

TNUUG is also currently working on its newsletter as well as setting up an archive server for some popular public domain software.

Future Plans

In 1992 TNUUG will work on extending its customer bases. Our goal is to double the customer base in 1992. We hope that after the 1992 Seminar, more people will be willing to join TNUUG. We are also currently talking to computer vendors to more generously help with the organisation of this first TNUUG sponsored seminar.

Contact Points

TNUUG
IRSIT
BP 212
2 Rue Ibn Nadime
1082 Cite Mahrajane
Tunis, Tunisia

Telephone +216 1 787 757
Facsimile +216 1 787 827

E-mail TNUUG@Tunisia.EU.net

UKUUG Column

Mick Farmer
Birkbeck College
Malet Street
London WC1E 7HX
England

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 interests he can be found teaching at Birkbeck College.

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

The UKUUG is currently in the process of improving the services we offer to our members. We are increasing the number of local groups and starting up special interest groups in topics like IP and MINIX. The remainder of this column is devoted to other services provided by the UKUUG.

Newsletter

Our own newsletter, news@UK, has finally gone into production; the first issue coming out on 1st March, a modest eight pages. We intend to publish six issues per year, building up the content as our members tell us what they would like to see. The editor is Sue Small sue@cs.bbk.ac.uk, who is interested in receiving articles, feedback, letters, etc. from members.

The newsletter is also available from the UKUUG archive. PostScript and ASCII versions are available, the ASCII version is also posted to the newsgroup uk.ukuug.

FaceSaver

The system is up and running again after serious problems with the Acorn hard disk. During the recent EurOpen conference in Jersey, the FaceSaver system was used to generate delegate badges as well as the final delegate list. Although many new faces were captured in Jersey, there are still many people whose faces are not on the UKUUG archive.

If you are interested in having the FaceSaver service at your conference or exhibition please contact me at the usual address.

LISA Workshop

This one-day workshop replaces our traditional summer meeting which we have had to cancel at short notice. The meeting will concentrate on issues associated with being connected to a network, whether via UUCP, IP, or high bandwidth wet string. The workshop is being organised by Neil Todd neil@gid.co.uk.

The aim of the workshop is to allow attendees to understand the significant issues and problems that may be encountered when network connectivity is obtained. Topics currently planned include the following:

- Name server configuration and bootstrapping.
- Message router configuration, e.g. Sendmail, PP, etc.
- Security aspects, e.g. firewalls and system monitoring.
- Living with a WAN connection — pluses and minuses.
- Network monitoring and network services.

The workshop will be held at Imperial College, London on 3rd July. Further details can be obtained by sending electronic mail to the list ukuug-wshop-lisa@uknet.ac.uk. Alternatively, contact our secretariat and ask to be put on the mailing list.

IP Workshop

This workshop will now be held this Autumn. Further details can be obtained by sending electronic mail to the list ukuug-wshop-ip@uknet.ac.uk, or by contacting our secretariat at the address given above.

Call for Papers and Announcement

USENIX Winter 1993 Technical Conference: The Challenge of Innovation

San Diego, California, 25-29 January 1993

UNIX and its cousins find themselves in increasing use throughout the industry. Succeeding in the challenge of meeting the world's expectations of software is an increasingly difficult task.

This USENIX conference is looking for innovative papers in a variety of areas, for example:

- Computing in the very large
  - Global connectivity
  - Coping with explosive growth
- Distributed environments
  - Client/server
  - Location independent computing
- Sociological and societal impacts
  - Connectivity vs. Security
  - New base applications
- Utilising state-of-the-art hardware
  - Storage systems
  - Communications systems
  - Large networks
- Exploiting increasingly layered software
  - Object oriented systems
  - Creative new interfaces
  - Complexity management
  - New development techniques
- Visionary base systems software
  - New building blocks
  - Novel leverage of standards

At the USENIX Winter 1993 Technical Conference, systems researchers and developers, systems administrators, software professionals, programmers, applications developers, support staff, technical managers, and educators tackle questions of immediate importance to advanced computing systems development and management.

The Program Committee solicits new work on all topics related to UNIX or UNIX-inspired programming and technologies. Vendors are welcome to submit technical presentations, but the program committee will reject product announcements.

Form of Refereed Paper Submissions

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

A Good Extended Abstract Should Contain:

- Abstract (same as in the final paper)
- Introduction (to the problem and its importance)
- Solution (details on the problem and its issues, design decisions, tradeoffs,
- motivations, implementation details)
- Evaluation
- References to previous work

Every Submission Should Include:

- The extended abstract
- One contact author with a daytime phone number and surface mail address

- Email address, if available
- Home phone number (volunteers work at night!)
- Indication if any authors are students
- List of audio/visual equipment desired beyond microphone and overhead projector
- Please note: presentations are usually scheduled for 25 minutes

Where to Send Submissions and Make Inquiries

Six paper copies of each submission should be sent to:

Rob Kolstad
7759 Delmonico Drive
Colorado Springs, CO 80919, USA

Make inquiries regarding submissions to Rob Kolstad at +1 719 593-9445 or via email to kolstad@bsdi.com.

Conference Program Committee

General Chair: Rob Kolstad, Berkeley Software Design, Inc.

Technical Program Chair: Dan Geer, Geer Zolot Associates

Matthew Blaze	Princeton University
Tom Christiansen	CONVEX Computer Corporation
Clement T. Cole	Locus Computing Corporation
James Duncan	Pennsylvania State University
Dick Dunn	eklektix
Peter Honeyman	CITI, University of Michigan
Daniel V. Klein	Software Engineering Institute
Steve McDowell	Exlog, Inc.
Dinah McNutt	Tivoli Systems, Inc.
Kent Peacock	Intel Corporation
Gretchen Phillips	State University of New York at Buffalo
David S. H. Rosenthal	SunSoft, Inc.
Jeffrey R. Schwab	Purdue University
Mary Seabrook	Open Systems Solutions, Inc.
Dave Taylor	SunWorld Magazine
Saul G. Wold	Sun Microsystems

Awards for Best Papers

A cash prize will be awarded by the conference program committee for both the best paper and the best paper by a full-time student at the conference. With your submission, please indicate if you are a full-time student.

Relevant Dates for Refereed Paper Submissions

Extended Abstracts Due:	20 July 1992
Notifications to Authors:	19 August 1992
Final Papers Due:	20 November 1992

For More Conference Information

Materials containing all details of the technical and tutorial program, conference registration, hotel and airline discount and reservation information will be mailed in September 1992. If you wish to receive the pre-registration materials, please contact:

USENIX Conference Office
22672 Lambert St., Suite 613
El Toro, CA 92630, USA

Telephone +1 714 588-8649
Facsimile +1 714 588-9706

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 October 19-23, 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
- Standardisation 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 summarises 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

THE SINGLE MOST IMPORTANT PAN-EUROPEAN EVENT TO SHAPE THE FUTURE OF OPEN SYSTEMS

Utrecht

Holland

November

25-27

1992

EurOpen

UniForum

Reed Exhibition Companies

ROYAL DUTCH FAIRS

If there's one event in the 1992 calendar that's unmissable for anyone seriously interested in Open Systems in Europe, it has to be OpenForum 92.

A top-flight strategy and business conference will be combined with a high-level technical conference and a set of tutorial seminars to challenge the most demanding participants. And all this alongside a major exhibition showcasing Open Systems products.

OpenForum 92 is the only event backed by two of the world's largest associations dedicated to Open Systems, EurOpen and UniForum and managed by Europes' foremost organisers of exhibitions Royal Dutch Fairs and Reed Exhibitions.

Note the time and the place.

And be there: absolutely, positively be there!

. . . attending the conference/exhibition.

CONTACT IN U.S.

Conference : **UNIFORUM**

Tel:+1 800 255 5620

Fax:+1408 986 1645

Exhibition : **PEMCO**

Tel: +1 800 323 5155

Fax: +1 708 260 0395

CONTACT IN EUROPE

Conference : **Mrs Bea Smink**

Tel: +31 30 955 466

Fax:+31 30 955539

Exhibition : **Mrs Marjolein Jacobs**

Tel: +31 30 955662

Fax:+31 30 955539

OpenForum 92

First Pan-European "Open Systems" Conference and Exhibition

Royal Dutch Fairgrounds, Utrecht, The Netherlands

23-27 November 1992

The Conferences

There are two conferences "Business and Strategy" and "Technical".

Dr Arno Penzias

Dr Penzias will be the keynote speaker at the opening session on Wednesday 25 November. He is vice president of research at AT&T Bell Laboratories. Dr Penzias shared the Nobel prize for physics in 1978 for his work in radio astronomy supporting the "big-bang" theory of the origin of the Universe.

Doug Michels

Doug Michels will be the keynote speaker on the second day. He is executive vice president and co-founder of the Santa Cruz Organisation (SCO), a pioneering company in the Open Systems arena.

Many good papers have been received, the programme committees will have made their selections in mid May.

Tutorials

A set of high quality tutorials is being prepared. This continues the long standing tradition of using top level experts to present leading edge material to those privileged enough to attend.

The Exhibition

Exhibitors have been keen to promote themselves in what they see as *the* Open Systems event in Europe, bookings for space have flowed in.

Interest has come from the European headquarters of companies, not just the Dutch offices.

It is expected to exceed 3000 square meters of exhibition space.

This is an excellent response in a climate which has seen many of the existing, well established, Open Systems Shows suffer a decline of interest - a process that OpenForum '92 has accelerated.

Keep yourself informed

You are busy and have little time to spare. You need to see what new solutions are being provided in the Open Systems market.

You thus wisely attend *the* show which will give you the information that you need as quickly as possible.

The Sponsors

The sponsors, EurOpen and UniForum, represent about 90% of open systems users world wide.

OpenForum '92 is the first fruit of a growing spirit of cooperation between the two organisations.

Remember - Be there or be left out

Call for Participation

Conference Theme

The theme of UniForum 1993 is "Open Systems, Open Opportunities". Manufactuers, system vendord, software developers and end users all benefit from the features and capabilities provided by the UNIX and Open Systems environment.

This conferenc will examine how each of these groups is utilizing UNIX and Open Systems to increate their productivity - and will provide solutions to specific problems encountered by their organisations.

Conference Schedule

UniForum 1993 is scheduled from 15-19 March. All day tutorials will be held on 15 and 16 March. Conference sessions, including individual presentations and panel discussions as well as technical paper sessions, will run 17-19 March in conjuncion with the vendor exhibition.

Conference Format

In addition to keynote addresses by top industry executives and plenary seesions conducted by industry leaders, the UniForum Conference is composed of four main elements:

Individual Conference Presentations - industry experts will provide personal insights on either the management aspects or technical considerations of a particular subject.

Panel Sessions - moderated panels of experts provide multiple points of view on topics of interest to users, developers and marketers.

Short Technical Paper Presentations - provide a more technical view of an important techological advancement within the industry.

All Day Tutorial Sessions - training in specific topics on interest to end users, developers and administrators.

Conference Sessions

Five separate tracks or subject specific categories covering both business and technical issues are planned for:

- Interoperability
- Network Computing and Management
- MIS and Commercial Issues
- ISVs and Dealers - The Distribution Channels
- Architecture and Applications

Sessions within these tracks will run concurrently from Wednesday to Friday, each session will be 90 minutes long. Your proposal must include the topic, content and indended audience for your session, and include the names of three qualified panelists (unless you are proposing an individual presentation).

Sumitting Abstracts

A submissions form is available from UniForum. It must be received no later than 1 July 1992, proposals will not be accepted by 'phone, but may be sent by fax or email.

Submit to:

Ed Palmer
UniForum '93 Chairman Committee
UniForum Association
2901 Tasman Drive, Suite 201, Santa Clara, California 95054, USA
Telephone +1 408 986 8840
Facsimile +1 408 986 1645

E-mail ed@director.uniforum.org

Programme Committee

Eileen Coons	Open Software Foundation
Steve Payne	UNIX International
Steve Zalewski	Oracle Corp
Bob Lewine	X/Open Company
Bill Keatley	American Airlines
Bill Bonin	Hewlett-Packard Co
Judith Hurwitz	Hurwitz Consulting Group
Paul Druckman	CAD/CAM Technologies Inc
Mike Millikin	GHA
Richard Fichera	RF Consulting
Ed Palmer	UniForum Association

EurOpen Publications

Publications available though EurOpen

			£	ECU
EurOpen Proceedings	Munich	Spring 1990	20.00	30
	Nice	Autumn 1990	25.00	37
	Tromsø	Spring 1991	25.00	37
	Budapest	Autumn 1991	25.00	37
	Jersey	Spring 1992	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	Medium Size Only	White, Grey, Black	12.00	16
AFUU Publications	Dossier Benchmarks AFUU 1989 Results	Mars 1990	24.00	35
The following publications	(Summary of benchmarks sent on request	Mars 1991	34.00	50
are written in the	Réussir avec UNIX (for administrators)	1988	4.00	5
French language	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
	Guide of Open Systems Actors		5.00	7
	Guide des Acteurs des Systémes Ouverts		5.00	7
	Unix et Systémes overt de la Portability au portage		18.00	27

			£	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
EurOpenD1 R6	42.00	60
EurOpenD2	42.00	60
EurOpenD3	42.00/60.00	60/86
EurOpenD4	50.00/60.00	72/60
EurOpenD5	42.00	60
EurOpenD6	78.00/96.00	112/114
EurOpenD7	24.00/60.00	35/86
EurOpenD8	42.00/60.00	60/86
EurOpenD9	52.00/60.00	75/86
EurOpenD10	33.00/60.00	48/86
EurOpenD11	42.00/60.00	60/86
EurOpenD12	42.00/60.00	60/86

	£	ECU
EurOpenD13	42.00/60.00	60/86
EurOpenD14	42.00/60.00	60/86
EurOpenD15	80.00/120.00	120/172
EurOpenD16	42.00/60.00	60/86
EurOpenD17	42.00/60.00	60/86
EurOpenD18	42.00/60.00	60/86
EurOpenD19	42.00/60.00	60/86
EurOpenD20	24.00/60.00	35/86
EurOpenD21	42.00/60.00	60/86
EurOpenD22	42.00/60.00	60/86
EurOpenD23	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 amount 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 _____

_____

_____

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

Company name _____

Email address _____

Telephone _____

Facsimile _____

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_____

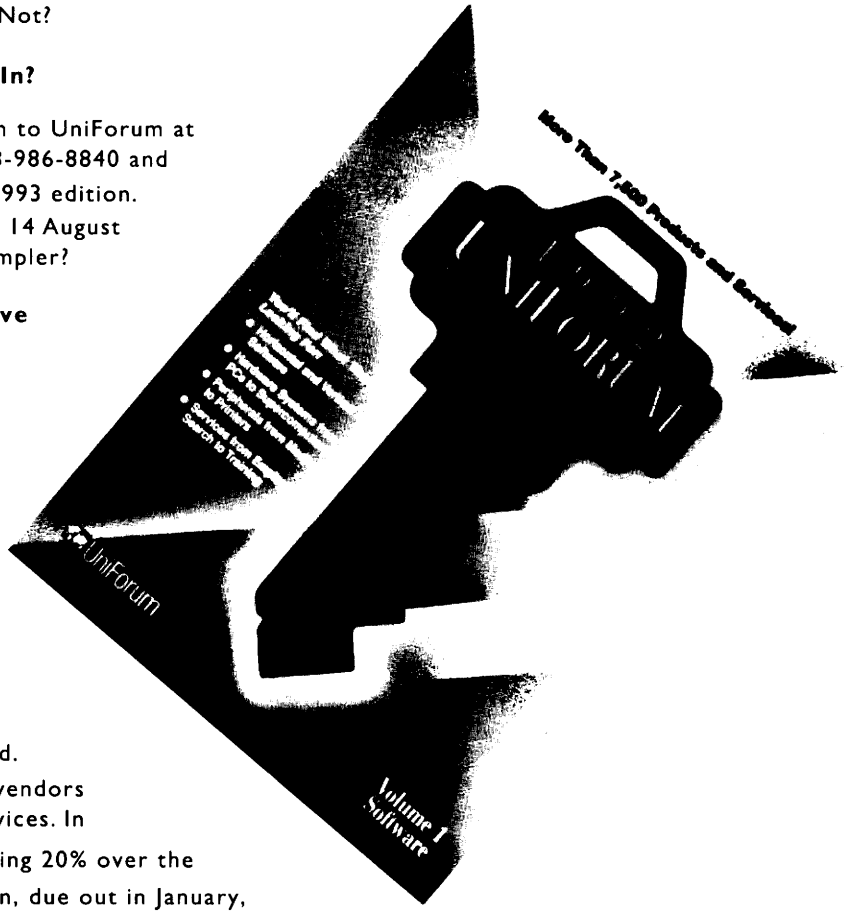
Date of Expiry _____

Signed _____

Date _____

2,100 Companies Listed 7,600 UNIX and Open Systems Products for FREE in the 1992 UniForum Products Directory – Is Yours One of Them?

If you're not in the world's largest catalog of UNIX and open systems products, ask yourself: Why Not?



Then ask yourself: How Do I Get In?

Simple. Fax your company information to UniForum at +1-408-986-1645 or call us at +1-408-986-8840 and we'll send you an entry form for the 1993 edition. Return your completed entry forms by 14 August 1992 and you're in. What could be simpler?

You Can't Find More Cost-Effective Advertising!

UniForum offers you the free opportunity you need to expose your products and services to more than 12,000 UNIX and open systems products purchasers and users worldwide. What could be more cost-effective advertising?

Be A Part Of The Open Systems Action

You can bet your competitors are listed. The 1992 Directory listed over 2,100 vendors and more than 7,600 products and services. In all, the 1992 Directory grew a whopping 20% over the previous edition. And the 1993 edition, due out in January, promises to be even bigger and better.

The two-volume UniForum Products Directory includes: – Software, Hardware, Peripherals and Services in over 300 categories -- Six indexes for quickly and easily finding vendors and products – The only comprehensive listing of products that comply with POSIX and X/Open's XPG2 and XPG3 specifications

So What Are You Waiting For?

Fax your request for a 1993 Directory entry form to UniForum today! UniForum's fax number is +1-408-986-1645. Email requests should be directed to kathy@uniforum.org.

BONUS OFFERS!

Get a copy of the 1992 UniForum Products Directory at 30% off the $95 list price! At just $69.95, the best deal in the industry! Contact UniForum today and mention code 9251077 with your order to take advantage of this offer.

An Even Better Deal!

Place your order today for the 1993 Directory and pay only $59.95. That's a 35% discount! Mention code 9251078 when you call to purchase the 1993 Directory at this great discount.

European Post and Packing add $55.

Orders may be placed with Owles Hall (see the order form on page 37) or directly with UniForum at:

UniForum 2901 Tasman Drive, Suite 201 Santa Clara, CA 95054, USA
Telephone +1-408-986-8840, Facsimile +1-408-986-1645

UniForum
The International Association of Open Systems Professionals

EUnet Report

Glenn Kowack
EUnet Chief Executive

E-mail glenn@mcsun.eu.net

Major new events in EUnet include our joining both the CIX Association and Ebone'92, beginning InterEUnet service in several new countries, obtaining new equipment for Amsterdam operations, and providing services to many new EUnet sites.

Peter Drucker, the famous Viennese-born management theorist, is fond of saying that everything that happens inside an organisation is a fantasy while everything that happens on the outside is real. The great danger to organisations like EUnet, which involve many people (frequently volunteers) in many organisations and countries, is that they can become fatally obsessed with their internal process. EUnet was founded on "providing networking any way possible" under the most difficult economic, technical, and regulatory circumstances. This required and was made possible by a high degree of subtlety, and there has always been a tension between focusing on the inside (to keep everyone working together efficiently), and making things happen on the outside – in the service of network users.

Recent months have seen a sea-change in EUnet. Without abandoning their sensitivity to internal requirements, EUnet activists have become even more focused on user and service-provision issues. This is evident by more focus on new services, by moving more quickly through organisational questions, by greater scrutiny to questions of service delivery and supporting technology. This sort of evolution of attitudes is impressive to see. EUnet continues to become more and more efficient, practical, and professional.

EUnet Site Counts

EUnet grows along two axis: the first is by the number of subscribing sites; the second is by the amount of service and traffic used by each site. Here is a quick snapshot of EUnet site counts.

	January	February	March
Email	2196	2234	2300
News	740	760	779
InterEUnet	113	119	122

(A "site" is a single subscribing organisation, at one site, with one link to its EUnet backbone. InterEUnet is the EUnet IP service.)

This represents good, steady growth, which is roughly according to plan. As more EUnet countries come on board, we should see an acceleration of this growth.

EUnet Joins CIX Association

EUnet joined the CIX Association in April. (Please see the attached press release.)

This is an important step for EUnet: CIX (pronounced "kicks") membership gives EUnet users access to a very large and growing body of users, it reaffirms our leadership role in providing quality services, and positions us for additional growth.

It also makes EUnet a party to an important process: clarifying open access to user-supported networks in an environment otherwise supported (and confused) by government subsidies.

Peer and Cooperating Networks

- RIPE NCC

The RIPE Network Coordinating Centre is now up and running, under the management of Daniel Karrenberg (formerly of EUnet). The NCC will provide a variety of important services to the European IP community, mostly centering around IP network status information services and coordination.

- Ebone'92

In my previous article, I cited the upcoming Ebone'92 initiative. That effort is now in full bloom, providing IP transport services to many regional network service providers across Europe, including EUnet, NORDUnet, HEPnet, EARN, SURFnet and others. The initiative operates on donated resources (EUnet is contributing its share of the Stockholm-Amsterdam and the Amsterdam-CERN links). EUnet formally signed the Ebone Memorandum of Understanding in January of this year. Ebone is managed by several different committees:

The Ebone Consortium of Cooperating Organisations (ECCO) is the "board" of Ebone, and includes a representative from each member organisation. The Ebone Management Committee (EMC) is a small executive committee, in charge of business, financial,

and executive management issues. EUnet is participating in both of these committees (your author is the EUnet representative).

The Ebone Action Team (EAT) is the technical committee, in charge of determining technical directions, resource requirements, and overall topology. The Ebone Management and Operations Committee (EMOPS) is the coordinating body for all sites directly involved in daily operations of the Ebone sub-net. EUnet is capably represented at these last two committees by Stephan Biesbroeck of EUnet-Belgium

EUnet participates in Ebone'92 because of its extremely common-sense, efficient, and cooperative approach to networking. We expect to use Ebone'92 to satisfy a variety of international connectivity requirements in the future.

In parallel with this effort, RARE (the European coordinating body for research networking) is continuing to make progress on it's plans for a company (an "Operational Unit") to manage a 2-megabit pan-European backbone for European research networks. Although a needed function, we are concerned that the OpUnit will be a top-heavy organisation which will not be sufficiently responsive to the needs of the networking community. Many in that community believe that the OpUnit will suffer from the same problems which beset the IXI X.25 network. This would not be a terrible problem were it not for the fact that RARE and the OpUnit expect to absorb both the RIPE NCC and the Ebone'92 structure. EUnet and EARN (since they are not "national research networks") are particularly concerned since they are being excluded from direct participation in the operation of the OpUnit.

The future remains uncertain: will networking see new practical leaps, or will it continue to be encumbered by slow-moving bureaucracies? EUnet will continue to cooperate with any practical network service providers as long as the terms and conditions are reasonable. We hope to see the good example of Ebone'92 continue well into the future. We'll keep you posted on these important developments in future articles.

Changes Among the National Nets

- Luxembourg Progress

The Luxembourg cooperating site is making steady progress toward full backbone status.

- New Directions in the United Kingdom

The UKUUG and UKC recently began a round of extensive UKnet planning discussions. Although a bit premature for announcement, we are hearing of interesting new plans and expect to see some exciting future directions announcements soon.

Austria Operations Transition

The UUGA backbone began operations during February of this year, and soon thereafter connected their first InterEUnet subscriber. They have also recently concluded a line-sharing agreement with the major universities in Austria and will have a new leased line to CERN by the summer.

- Switzerland

The EUnet-Switzerland NOC installed a leased line to the Ebone EBS located at CERN, Geneva earlier this year, and is rapidly expanding their InterEUnet service. A POP (Point-Of-Presence) in Basel is due to be installed in two months.

- Finland, Ireland, and InterEUnet

EUnet-Finland and IEunet connected their first InterEUnet subscribers in April.

- Algeria

Algeria is making fast progress on setting up their backbone. They have acquired a host machine and are working with other backbones to close on the remaining technical details. We expect them to be up and running by the time this Newsletter is distributed.

InterEUnet Overview

The following countries are now providing InterEUnet services:

Austria	France
Germany	Finland
Ireland	Italy
Netherlands	Switzerland
United Kingdom	

This coverage in nine countries now gives EUnet an extremely wide geographical reach.

New Backbones

EUnet continues to see a flurry of interest from around Eastern and Central Europe, and the Middle East region. We'll have future reports on these areas as they advance to connected sites and beginning backbones.

Technical Developments

- 128Kb Trans-Atlantic Link Traffic

As mentioned earlier, the second axis of EUnet growth is service traffic. Our trans-Atlantic link is a good general indicator or EUnet traffic growth. The load on our 128Kb trans-Atlantic link (to UUnet and from there on to NSFnet and CIX) is increasing steadily. During a typical day in prime time (8am-6pm), US to Europe traffic runs over 80% saturation (~105Kbps), Europe to US traffic runs at about 50% saturation (~70Kbps). There is a very high likelihood that we will need to upgrade that link (or obtain capacity elsewhere some time this summer). The underlying trend is a doubling of trans-Atlantic traffic approximately every 9 months.

This growth is a mixed (but mostly positive) blessing for EUnet. On the one hand, it means that we must manage ourselves carefully to be sure that we have a match of income and expenditures. On the other hand, it is a real opportunity: as line sizes increase, the price-per-bit goes down steadily. And, the larger the line, the better it is able to absorb "bursty" traffic from any one subscriber. Both of which yield better services for our subscribers; our *raison d'etre*.

New Computing Power in Amsterdam

mcsun, although a fine old war horse, is showing its age. It's computing power, and price and availability of new interfaces and components are starting to lag behind our steadily growing computational requirements. Early this year we added two new hosts to the Amsterdam NOC (Network Operations Center): two SUN ELC workstations, each configured with 24 MB of memory and a 1.2 Gb disk. We have moved the nameserver function from mcsun to the first ELC ("ns.eu.net"). This move will free a significant amount of memory on mcsun; it will also guarantee reliable and steady name services to EUnet and the network community. We are beginning to move other functions to the second ELC ("news.eu.net"). The latter will be supporting Network News very soon; this will likely include an upgrade from "Bnews" to the "Cnews" system.

Since the two machines are identical, in extreme emergencies one could be substituted for the other (although our maintenance contracts should make this unnecessary).

The arrival of these two new "small" workstation machines is significantly improving our traffic throughput, transmission times, and substantially simplifying operator effort.

Distributed Archive Service Progress

As planned, EUnet hired Martijn Roos Lindgreen in December to design a distributed EUnet archiving system (explained in detail in the Winter '92 Newsletter). Work is now progressing according to plan. We would like to thank the University of Amsterdam for providing the umbrella under which this work is being performed.

We are presently looking for an EUnet archivist. This role will concentrate on archive content and will complement Martijn's activities.

An Addition to the Executive Committee

At our Tunisia meeting, we decided to add Simon Poole (of the CHUUG and EUnet-Switzerland) as Exec member-at-large. His appointment is designed to help us address the increasing activity in EUnet and also to improve internal communications: Simon is now the official internal "town-crier" of EUnet. By issuing regular internal bulletins, he will ensure that the Backbones are aware of Exec actions and other activities.

Progress on EUnet Structure and Directions

At the Jersey '92 EurOpen meeting, the Governing Board passed a resolution asking for a proposal to be voted upon at the Utrecht Governing Board meeting in November of this year. The resolution calls for further analysis of EUnet service and organisational directions. If the business case calls for it, we will call a vote on establishing EUnet as a separate legal entity.

Internal Cost Structure and Billing

The new EUnet internal price and cost structure was put in place at the beginning of this year. The backbones and the European-level organisation find it a real improvement: it is simple, appears fair, and is bringing in the right amount of funds.

We have also completed the previously-announced transition of billing from CWI to Owles Hall. This will give us better control of our finances.

Administrative Changes

Daniel Karrenberg has moved from EUnet to his new position as manager of RIPE NCC. Although we will miss Daniel's contributions, we are very happy to see competent people working at this very important service. And, we especially want to thank Daniel for many years of contribution to EUnet. We are presently looking for a new staff members to fill Daniel's former role.

The EUnet office in Amsterdam is also looking for an office assistant. This person will help with the myriad tasks required to make an organisation like EUnet work smoothly.

EUnet Public Presence

Cormac Callanan (EUnet-Ireland) is now officially EUnet's Public Relations coordinator. Cormac is acting as the first point of contact for persons and organisations interested in general information regarding EUnet, and is also responsible for planning and executing a number of public relations initiatives. To that end, we now have a mail alias Public-Relations@eu.net, which is monitored by Cormac.

The first of these initiatives was the "EUnet Joins CIX Association" press release. This press release was mailed to key publications throughout Europe, and distributed on a number of mailing lists and news groups. It was also quickly translated into many European languages; Cormac now has a team of translators ready to prepare other EUnet communications.

The EurOpen and EUnet glossy brochures have been completed and are now being distributed to EurOpen members, Teus Hagen did the lion's share of work on this effort and deserves credit for service above and beyond the call of duty.

UKIC Conference

Your author gave a presentation to the UKIC "Internetworking '92" Conference in Farnborough, UK along with Mr. James Slater of UKC/UKnet. Our presence there generated significant new interest in EUnet services.

EurOpen Jersey Conference

EUnet held a BOF (Birds-Of-a-Feather) session at the EurOpen Jersey Conference in April, which was well-attended. This was preceded by a talk by your author which gave an overview of EUnet and a personal view of the recent history of OSI networking in Europe.

Kudos

- Tunisia '92

The Winter '92 EUnet Backbone meeting was sponsored this year by Tunisia. Local organising was done with unusual style and flair by Mondher Makni and his crew, providing us with excellent service in the resort town of Hammamet.

This was our first meeting in Northern Africa. The meeting was also exceptionally economical, proving that EUnet need not limit major meetings to "central" Europe for financial reasons.

- UUnet Trans-Continental Link

Rick Adams and the crew at UUnet are allowing EUnet traffic to freely transit the UUnet T1 link from the US East Coast to the CIX West POP. This contribution is critical to our access to CIX West. Our thanks to the staff of UUnet.

PRESS RELEASE – EUnet JOINS CIX ASSOCIATION

Amsterdam, 2 April 1992. EUnet, the pan-European open systems computer network, today announced that it has become the first operating European network to join the Commercial Internet Exchange Association (CIX).

EUnet is Europe's largest subscription-funded network, focusing on users in the research and development community since 1982. Operating in nearly every country from Iceland to Russia and as far South as Tunisia, EUnet connects over 2500 networks and sites and has gateways to every major network in Europe. EUnet services include Electronic Mail, Network (USENET) News, and Archive access. EUnet's InterEunet service, in operation since 1990, provides international and intercontinental LAN interconnection services based on the TCP/IP protocols to over 100 organisations across Europe. EUnet is a service by and for the members of EurOpen, the European Forum for Open System, a not-for-profit organisation.

The Commercial Internet Exchange Association, Inc. (CIX), was formed in 1991 as a non-profit trade association by UUnet Technologies (UUnet), Performance Systems International (PSInet), and General Atomics (CERFnet). The CIX founders and all new members cooperate to provide a non-restrictive packet interchange for TCP/IP and OSI traffic. The Association actively solicits membership and network connections with all commercial and non-profit US national, Regional, and Mid-level networks, Federal and state government networks, and international network organisations.

Mitch Kapor, President, CIX Association says: "I am delighted that EUnet has joined the CIX. The CIX will now truly become an international network connecting European and U.S. network subscribers."

Glenn Kowack, EUnet CEO, states: "There is a large and growing community of users in the United States who, although involved in R & D, do not happen to have access to NSFnet. EUnet's CIX membership complements our NSFnet connectivity by giving EUnet subscribers access to those users. We see CIX functioning in the US in the same manner as does Ebone '92 (European Backbone) in Europe."

The EUnet community is enthusiastic about its membership in the CIX, which will help to advance the evolution of networking services and open systems worldwide. For further information contact:

Cormac Callanan
Telephone +353 1 740 508
E-mail Public-Relations@eu.net

Report on the January 1992 IEEE POSIX Meeting for EurOpen

Stephen R. Walli
EurOpen Institutional Representative

E-mail stephe@mks.com

Standing on Huntington Beach, gazing across the moonlit Pacific after a five hour Sponsor Executive Committee meeting, is a small group of POSIX refugees. It is 10 pm. The oil rigs glow golden in the distance like great Christmas trees. The surf crashes and hisses at our feet.

Jason: Just remember, Stephe, in the grand scheme of things....

Stephe [interrupting]: ...Oil's important.

Okay. So I'm a cynic. It was generally an ugly week. I believe there are still some fundamental flaws in the structure of POSIX. Working group members are still grappling with the enormity of the beast we've created. GUIs, rescued from the jaws of defeat by the IEEE Standards Advisory Board, entered the scene again. Life (and POSIX) goes on.

POSIX is coming! POSIX is coming!

The IEEE POSIX working groups **are** coming to Europe. This Autumn, the POSIX working groups will be meeting in Utrecht, NL, from 19-23 October. The meetings will be held at the Holiday Inn and the Scandic Crown Hotel.

Information will be posted to this column, and on the net where appropriate, as the meetings approach.

There will also be an ISO Working Group 15 (WG15 - ISO POSIX) meeting around the same time. It will either be the week before or the week after, and will also be in Europe. Again, information will be forthcoming.

SICC and the PMC

POSIX often makes use of the fact that a small group of people can go off and accomplish what a large group will spend hours arguing about. Small committees are the work method of choice in many areas. Two committees of the Sponsor Executive Committee (SEC) are the System Interfaces Co-ordination Committee (SICC), and the Project Management Committee (PMC).

SICC is gaining momentum. It is effectively the chairs of all the system interface groups:

- POSIX.1, the base system interface,
- POSIX.4, real-time extensions to POSIX.1,
- POSIX.6, security extensions to POSIX.1,
- POSIX.8, transparent file access extensions to POSIX.1,
- POSIX.12, protocol independent interfaces (Berkeley sockets and X/Open's TLI).
- POSIX.15, supercomputing batch interfaces,
- POSIX.17, directory services (think X.500).

They are a simple subcommittee of the SEC, that meets to resolve areas of conflict between the base standards, and to work out concerns which affect all of the documents.

The other group which is effecting the way POSIX is coming forward is the Project Management Committee (PMC). This was the fourth meeting of the PMC that has existed, and half of its eight members were due to retire. Two elected to stay on (and were voted back in by the SEC) for an additional two year term. Two new members (one of them the EurOpen Institutional Representative,) were also voted into the group.

The PMC is responsible for mentoring existing projects to ensure they are performing according to plan. They are also responsible for ensuring that new project requests (PARs) are complete and have all the attendant paper work before reaching the SEC.

One interesting result of their work this week was the splitting up of the POSIX.7 (System Administration) project. POSIX.7 has been re-organised into:

- a parent project (POSIX.7), which is responsible for co-ordinating the other parts of the POSIX.7 standard
- POSIX.7a to build a standard for print management,
- POSIX.7b to build a standard for software management,
- POSIX.7c to a set of guidelines for user management.

The final subproject is important. There is not sufficient agreement on how to do user administration to build a standard. There is a lot of existing practice

with administering users. Everyone agrees that the current solutions are not good. So they are building a set of agreed upon guidelines. It's really too bad the PMC didn't exist when other project requests were cut to suggest this sort of solution for certain other contentious immature areas of technology.

Multi-lingual Test Assertions

Multi-lingual Test Assertions are test assertions written in such a way that test suites in multiple programming languages could be written. This addresses the problems associated with testing conformance for something like an Ada run-time environment, that supports POSIX.5 (the Ada language version of POSIX.1 functionality) when all the test suites are written in C.

POSIX describes an interface to operating system services. It is based on the historical C interface to UNIX, but should not be restricted to just that language. No one wants to make the same mistakes made with GKS (which demonstrated you could write Fortran programs in any language) by forcing other languages to bind the interfaces in the same functional way.

This was why ISO WG15 (ISO POSIX) wants a programming language independent specification to be written of POSIX.1. Other languages can then bind the functionality appropriately. The semantic description would be kept in a single book, with the appropriate syntactic binding and glue in other books.

Test assertions written to POSIX.1 demonstrate the same problems. Everyone agrees that much of the functional content of the test methods for POSIX.1 should be the same no matter what programming language is being used to write the suite. If the test assertions from which the test cases are built were written to the programming language independent version of the standard, they could be bound to the language syntax at the same time as the functional specification, providing a language based set of assertions from which to build the conformance test suite.

Okay. The cat's out of the bag. We are really discussing language independent specification test assertions. If I used that as the title to this section, you might have turned the page. Hopefully, you now at least appreciate the problem to be solved, even if you still run screaming into the night.

Now all we have to do is solve the resource problem, and find people to help with the specification.

Distributed Security Study Group

A group of about twenty people met for a day during the January POSIX meeting to discuss distributed security technologies and scenarios. The group felt it had sufficient momentum and manpower to form a proper study group, and they will meet for the entire week at the April meeting in Dallas, Texas.

They were a little disorganised to begin with, but agreed to a set of existing specifications and models they wish to begin investigating at the next meeting.

There are those within the group that wish to take some time to investigate the current base of experience before proceeding, while others appear to want to pick a specification and begin tweaking it into a standard. Pretty aggressive considering they haven't cut a project request yet!

PAR Wars II – The Umpire Strikes Back

Since we last looked in on our story, our protagonists, OSF/Motif and Open Look (sometimes known as Rodan and Godzilla) were off chasing their project requests (PARs) up the IEEE Standards hierarchy. Having been told "no" by the Sponsor Executive Committee (SEC), they are now asking for sponsorship at the higher level of the IEEE Standards Advisory Board (SAB).

You will recall the flow.

April 1991, the PARs first surface. The intent is to directly form ballot groups to ballot the current programmers and users guides. The SEC, clearly uncomfortable with the obvious overlap between these two GUI PARs and the current work in P1201, argues for several hours (!) and then tables discussion at that time.

July 1991, discussion is re-opened. A small committee is formed to clearly state all pros and cons of the two projects. The presentation is made, and since all members of the SEC find at least one serious problem with the Motif and Open Look project requests, the projects are voted down.

Some were concerned over the apparent lack of maturity of the technology. No one that has tried the two technologies (with the exception of the PAR presenters) seems to like either of the interfaces. People are concerned that wrapping a standard around them now will prevent them from being improved and matured. Some user representatives clearly wish there to be a single unified standard.

With the apparent "death" of the two competing PARs came a motion to remove sponsorship from the existing P1201 project, arguing it suffers the same flaws. Discussion is tabled to the project management committee (PMC) until October.

October 1991, P1201 is allowed to survive. P1201.1 is making good progress at defining a higher level interface for simple windowing, based on current practice. While possibly not as functional as either the Motif or Open Look toolkits, it is useful to a wide range of applications, and there is consensus forming around it. P1201.2 is preparing a recommended practice document on windowing style, and is making good progress.

The Motif and Open Look projects presenters are off haunting other corridors (SAB). The SAB, pointing to the 802 LAN standards as if they were a "good" example, has said it is an insufficient reason to kill the project requests simply because they have overlapping scopes.

The SEC is responsible for approving projects for the IEEE Technical Committee on Operating Systems - Standards Subcommittee (TCOS-SS). This is where POSIX and the GUI standards all reside.

It seems the SAB is sympathetic to their plight and has agreed to sponsor the projects. Gone are the contentious ideas that the two camps will directly form ballot groups and ballot the current programmers references and users guides. The SAB has said that they must play fair, and play by the rules. (Direct balloting only exists on paper as a method of fast-tracking clearly uncontentious documents.)

The SAB has further offered them **back** to TCOS-SS/ SEC! The projects really **do** belong within the scope of TCOS. The SEC agreed to take a look at the revised sponsored-in-principle projects at the April 1992 meeting. I believe one member's phrase was to "accept the pig in a poke now, and rip the bag off later." Hopefully, it won't be too prophetic an analogy.

We have been cautioned that we can not trivially agree to accept them, then shut them down. They have been sponsored-in-principle by a higher power. The two projects have already each held their first meetings, outside of TCOS's sphere of influence.

I for one want to see them really play by the rules! If accepted by TCOS, they should certainly come into the Steering Committee on Windowing User Interfaces (SCWUI) realm of influence. No reason to exempt them from test assertions either.

Language independent specifications (LIS) are certainly appropriate considering the number of Ada developers that currently need to find messy solutions to working with these two GUIs in their native Ada environments. Indeed, they may discover that they functionally overlap more than they care to admit by writing the LIS. If they rationalise the LIS, as POSIX.12 is doing with C-based sockets and C-based XTI, then maybe Ada could benefit by coming up with a single binding standard!

Of course, if they were to come forward as recommended practices or guidelines, instead of as full use standards, they would not need to provide LIS and test assertions as robustness proofs. Something to think about.

As I said at the end of the last report, "thus are standards born."

Book Review

OPEN SYSTEMS - A Business Strategy for the 1990's
Pamela Gray
ISBN 0-07-707244-8
Price £24.95, 263 pages.

Reviewed by Peter Theobald, Xi Software Ltd

E-mail pbt@xisl.co.uk

One might have expected that a book written on this complex subject by such a well versed person as Pamela Gray would be over complicated for the non technically minded and full of jargon. On both counts this is definitely not the case.

This book could quite easily become a textbook for students embarking on courses relating to computer studies at colleges or universities. It defines the base level of information for all topics without delving into bits and bytes, and yet offers serious comment for the potential commercial user. The historical representation of the progress of the standards which are, or will be, adopted will be useful to students whilst offering the commercial user a pathway for his future progression. All abbreviations are well defined on their first use.

The interaction between the text and the selected illustrations whether discussing simple or complex situations is well thought out. Similarly everyday examples where standards have been adopted for products to increase their market acceptance will be understood by the reader and should be heeded by the supplier of Open Systems.

Anyone who does not yet use Open Systems will find that the approach adopted in this book is balanced indicating areas where they will have to be wary or where they will find that they will not have the facilities to which they have become accustomed as well as those areas where they will significant benefit from a move from proprietary systems. The minefield presented by the lack of de facto standards in the user interface is treated with excellence by the author. Security, the lack of some utilities, standardisation on international character sets, and system administration are all covered in the later chapters giving information available to the date at which the book was written.

No mention was made about the Pick operating system or the availability of Interactive UNIX for the small system. These would have been necessary to give a full Open Systems picture. These omissions should not be cause to impair the significance of this work.

I would recommend that anyone who is contemplating the future direction, or has the responsibility for the strategy, in their company for computing should read this book, and any sequel that may be written by Pamela Gray on this subject, prior to making any decisions on the direction to take.

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

During the last conferences week in Jersey, EurOpen Working Groups (EWG) became more and more active. Two working groups took the opportunity, to hold a meeting. see below the reports from the EWG on security and EWG on internationalisation. Between conferences, the EWGs work on email.

Outputs are the key points about working groups. Being a member of an EWG doesn't mean only to sit and wait for someone else to give information. This means that several people produce an output together. The level of implication and of exchange lead to increases in the level of expertise for each member.

If you want to join a group, please send your name and email address to the chairman through the "-request" addresses below (if you don't have an email yet, send your fax number to the EurOpen Secretariat, you will receive the messages through the fax gateway, but will not be able to send anything).

Working Group:

Benchmarks	
ewg-bench@eu.net	ewg-bench-request@eu.net
Internationalisation	
ewg-i18n@eu.net	ewg-i18n-request@eu.net
Security	
ewg-secur@eu.net	ewg-secur-request@eu.net
Copyright legislation	
ewg-cprght@eu.net	euw-cprght-request@eu.net
Market	
ewg-market@eu.net	ewg-market-request@eu.net

I have acted for 18 months as the EWG co-ordinator to launch the EurOpen Working Groups. It seems that we have raised a first milestone. There are new working groups at national and EurOpen levels. We still have a lot of work to do to consolidate and develop EWGs. But I am now more and more involved as a consultant in various international organisations. Due to time constraints I have decided to resign from my position of EWG co-ordinator. I hope that the job I have done during the past 18 months, will be helpful for the next co-ordinator to continue the development of EWGs.

I would like to take this opportunity to thank all those who have contributed to the creation of the EurOpen Working Groups: The coordination team, the national contact points, the EWG chairment, the executives of EurOpen and all those who have participated at the numerous meetings, E-mail lists and discussions which led to the creation of the Working Groups within EurOpen.

So long and long life to EWGs!

Minutes of the EWG-Security Meeting, 24 November 1992

The EurOpen Working Group on Security, has 28 members, from 16 different National Unix Users Groups. the group discuss in the ewg-secur mailing list, and held its first meeting on Tuesday morning, 7 April, at the Hotel de France, during the EurOpen Jersey conference.

The EurOpen Working Group on Security decided to have a meeting on each EurOpen event, twice a year, and to have a one day meeting in Utrecht, on Tuesday 24 November (EurOpen Working Group Day).

The main problem of the group is the POSIX drafts availability, so the EurOpen Working Group on Security decides that the P1003.6 and the P1003.7 drafts, and any other draft of interest have to be distributed by EurOpen Working Group secretariat via postal mail to the members requesting it. We asked EurOpen chairman and EurOpen IEEE Institutional Representative (Stephen Walli) to do what is needed to achieve this distribution.

The group had a debate to decide at which level we should contribute, and it reached the following consensus position: the EWG-Security point of view on

important documents as POSIX standards related to security must become an EurOpen point of view, but output decisions, representing an EurOpen point of view outside EurOpen, are first published in EurOpen newsletter, giving each EurOpen member a chance to contact us. The output will be then submitted to the Governing Board after having gone through a formal process within the Working Group.

The EurOpen Working Group on Security decides to write a short guide on the POSIX drafts related to security, to explain the purpose of the drafts, and to find members of IEEE P1003.6 and P1003.7 balloting processes, to present their work at the next meeting of the group in Utrecht, on 24 November.

To become a member of the Working Group on Security, just send a mail to ewg-secur-request@eu.net, or send a fax to the chairman.

EWG-Security chairman:

Herve Schauer
Herve Schauer Consultants
Unix & TCP/IP Security
142 Rue de Rivoli
F-75039 Parix Cedex 01
France

Telephone: +33 1 46 38 89 90
Facsimile: +33 1 46 38 05 05

E-mail Herve.Schauer@enst.fr

Report on the EWG Internationalisation/ Localisation Meeting, 6 April 1992

The meeting started with a presentation by Keld Simonsen of his draft, "Nordic and Baltic Locales and Character Sets". this 56 pages document describes the cultural dependent items described in the POSIX standard (ISO/IEC 9945-2 2CD 1991) for the nordic countries: Denmark, Finland, Iceland, Norway, Sweden and also for the two baltic countries Estonia and Lithuania and specifies the character set in common use in these countries.

It first describes "locale", "locale nationale", "locale definition" concepts, and presents the locale definition nationale. the valuable work is available for the other national groups through the chairman and e-mail. Since the localisation concerns each national group the EWG on Intern./Local. would be too large, it was suggested to work on subgroups; Nordic countries, Arabic speakers groups .... The Algerian and Tunisian chairmen, in a previous informal meeting, agreed to create a common arabic localisation WG, and adding Morocco to the group. The first meeting of this subgroup will be held in Algiers during the AIUUG even (9-12 May 1992).

Sergei from SUUG, pointed out that such subgroups won't be easy to create in the former USSR - and moves toward this direction should be taken with caution, because of the potential situation. However, Mr Sergei will provide the chairman with the Russian locale. In fact, it was decided during this meeting that each group provides the locales of his language by June 1991, followed by successive versions through OpenForum meeting.

This locale specifications will be based on the document presented by Keld. The output will be promoted via standard bodies to International Standardisation Bodies (ISO, X/Open, CEN ...).

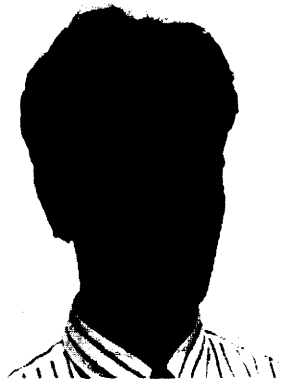
It was finally suggested to hold a workshop on localisation/internationalisation during the next EurOpen conference or on another convenient place and date.

To go further, and to develop the valuable results presented by Keld, the local groups or subgroups will need a lot of documentation from standardisation bodies and also from some major manufacturers and software houses. Keld, being active in many standardisation bodies, could provide the documentation or at least indicate to the other groups where to get it.

Before closing the meeting, the participants identified the need for a WG Coordinator, after the resignation of JM Cornu.

OSF Column

Mark Laureys
Communication Manager European Operations
Open Software Foundation
15 Avenue des Pléiades
1200 Brussels
Belgium



Mark Laureys joined OSF's European Operations based in Brussels in June 1991. He is responsible for the European Communications and his main task is to provide members of the European press and consultants with information on OSF.

He was previously working in a similar function for a network integrating company called Telindus Networks, part of the Telinfo Group.

For further information on this column, please contact Mark Laureys: telephone +32 2 772 88 88

The Open Software Foundation and Open Systems

The Open Software Foundation (OSF™) is a computer industry first: a non-profit corporation jointly established by the industry's leading vendors as a neutral supplier of open systems software technologies. OSF's mission is to be the leading supplier of a complete, innovative and equitable open systems environment to the world-wide computer industry.

OSF is a non-profit company supported by its membership - hardware and software vendors, end users, and university and research organisations. Unlike the bodies that set standards and specifications, OSF is a one-of-a-kind organisation that provides three-pronged offerings, made up of a specification, an implementation, and validation code.

OSF provides software that addresses three key user needs:

- portability -

which allows application software to run on multiple vendors' systems;

- interoperability -

which allows systems from different vendors to work together transparently;

- scalability -

which allows a software environment to be hosted on the entire range of available hardware platforms, from desktop devices to super computers.

History and Membership

OSF was formed in May 1988 with the support of seven original sponsors: Apollo Computer Inc., Digital Equipment Corporation, Groupe Bull, Hewlett-Packard, International Business Machines Corporation, Nixdorf Computer AG, N.V., and Siemens AG. Subsequently, Hitachi Ltd. of Japan became a sponsor.

The number of OSF members has surpassed 300. Members include systems manufacturers, hardware systems vendors, independent software vendors, semiconductor manufacturers, service companies, industry consortia, educational institutions, public and quasi public entities, and consulting and research firms. Membership is open, and all members may participate in decision-making and take advantage of early access to technologies under development.

The Open Process

OSF uses an innovative, open process to solicit, acquire, and develop technology. Through this process, the world-wide computing community works to promote open systems.

Central to the open process is OSF's Request for Technology (RFT). Through the RFT process, OSF solicits and evaluates technology from the world-wide computer industry as well as educational institutions, government agencies, and end users. All OSF members,

submitters of technology, and other interested parties may contribute ideas on technological and market needs. Taking into consideration the recommendations of its membership as well as those of non member submitters, industry consultants, and standards groups, OSF selects technology to incorporate in its open computing environment.

OSF's RFT and development processes also provide members timely access to open systems technologies. Copies of code under development, called snapshots, help members evaluate OSF software, develop applications in parallel with the efforts of OSF, and port the software to their systems. In this way, OSF's Snapshot Program ensures rapid transfer of technology to the industry.

The open process extends beyond the development of new technologies to include research and address technical support, education, and market needs. For example, the OSF Research Institute works closely with universities, industry research laboratories, and government research institutions to benefit form their experience working with emerging technology. It then recommends development directions for OSF as well as opportunities for future RFT's.

The OSF Portability Lab is a state-of-the-art facility that provides technical support and guidance, assisting OSF members with the timely transfer of OSF technology to their product lines.

Other mechanisms that support the open process include membership meetings and special-interest groups. OSF regularly meets with its membership to exchange ideas on open systems technology. Member special-interest groups, or SIGs, have a powerful voice in the open process. made up of experts form member organisations, they help define the scope and requirements for OSF's RFT's.

Through all of its efforts, OSF adheres to seven guiding principles:

- Offerings based on relevant industry standards
- Open process to actively solicit input and technology
- Timely, vendor-neutral decision process
- Early and equal access to specifications and software
- Hardware-independent implementations
- Reasonable, stable terms for licensing
- Technical innovation through universities and research

OSF Offerings

OSF provides an open computing environment, a collection of open systems technologies that enable users to mix and match software and hardware from several suppliers in a virtually seamless environment.

OSF's software portfolio breaks down the barriers between diverse systems, giving users the freedom to choose the systems and technologies that best meet their business needs.

This vendor-neutral software environment consists of several offerings, all based on relevant industry standards. Components are the OSF/1 operating system, the OSF/Motif graphical user interface, and the OSF Distributed Computing Environment. DCE and Motif can be layered on OSF/1 as well as other operating systems. In addition, OSF is developing technology for its Architecture-Neutral Distribution Format (see below).

The OSF open computing environment is a rich one that will continue to evolve to meet industry needs. Working with its membership, OSF will consider both proven and emerging technologies for inclusion in the environment.

The OSF/1 operating system delivers a new foundation for computing systems, taking full advantage of the revolution underway in multiprocessing and networked computing. It integrates the best open systems technologies available and conforms with X/Open and POSIX specifications. OSF/1 features a modular, streamlined kernel that OSF has engineered to perform as a stable, powerful platform for current and future applications.

OSF's first RFT, issued in July 1988, solicited technologies for the graphical user interface component of OSF's open environment. It resulted in 39 submissions, 23 of which met OSF's criteria. OSF selected several technologies to integrate into its OSF/motif offering, a graphical user interface providing user-oriented PC-style behaviour and screen appearance for applications. OSF/Motif gives applications a common appearance and behaviour on all classes of systems, from desktop devices to mainframes. Today it runs on more than 145 hardware platforms and 47 operating system platforms available world-wide.

Using its RFT process, OSF also selected several technologies for inclusion in the OSF Distributed Computing Environment (DCE), which addresses the industry wide problem of computing in environments made up of hardware, software, and networks from multiple vendors. The DCE offers a comprehensive, integrated set of essential services that support the development, use, and maintenance of distributed applications. It lets users access diverse network resources from the desktop.

OSF also has selected technology for an Architecture-Neutral Distribution Format (ANDF), which enables developers to create and distribute applications in a format that can be installed and run on diverse open systems architectures. With this technology, OSF is delivering the first hardware-independent software distribution format, providing a consistent development and distribution environment for multiple platforms. ANDF is under development.

The objective of OSF's fourth RFT is to simplify both the development and use of system management applications across multiple hardware platforms. OSF identified a set of managed objects, a management framework, and a core set of relevant applications for OSF's Distributed Management Environment. DME will provide a uniform framework for efficient, cost-effective management of open systems. It will unify system, network and application management, as well as provide tools and utilities that address basic management tasks associated with the installation and operation of stand-alone and distributed systems.

OSF and Standards

OSF's offerings are consistent with the X/Open Common Applications Environment (CAE) and Portability Guide, Issue 3 (XPG3), The National Institute of Standards and Technology Application Portability Profile, IEEE POSIX, and equivalent European and international standards.

OSF Research Institute

More than one-third of OSF members are universities, industry research laboratories, or government research institutions. The OSF Research Institute establishes strong links with these research organisations to benefit from their experience and technology. Its mission is three-fold:

- To involve the research community world-wide in the evolution of open systems
- To analyse emerging technologies and identify opportunities for future Requests for Technology or OSF development
- To build a long-term shared vision and architecture for the future of open systems

The OSF Research Institute serves as the link between the foundation and both university and commercial laboratories around the world. To date, the OSF Research Institute has awarded several grants to research organisations, including MIT Project Athena; Trusted Information Systems; Carnegie Mellon University, USA; the University of Guelph, Ontario, Canada; the University of Lowell, Massachusetts, USA; Vrije University, The Netherlands, and the University of Calgary, Alberta, USA.

Book Review

Managing Projects with Make
Andrew Oram and Steve Talbott, O'Reilly (Nutshell), 1991,
ISBN 0-937175-90-0
Price $12.95, Soft Back, 140 pp,

Reviewed by Rob Henley, Siemens-Nixdorf,Bracknell, United Kingdom.

E-mail rob@siesoft.co.uk

I use *make* very frequently in my day to day work and thought I knew everything that I needed to know about it. After reading this book I realised that I was wrong!

This book is very readable and well presented. For someone already familiar with the UNIX environment, it will take you a couple of evenings to browse through. You are almost certain to find something of interest, and probably several features which will be immediately useful.

If you are a UNIX novice, the book provides a gentle but thorough introduction. Its well worth reading just to give you an idea of what *make* can do for you: when you need to put ideas into practice, the book provides a good index and Quick Reference section. There is also a wonderful picture of a *potto* on the front cover. If you would like to know about the potto, you'll have to read the book!

The main text starts with a chapter on writing a simple makefile. The book is organised around practical examples and it would be a good idea for newcomers to *make* to work through them at a terminal. There is even an FTP site from which you can obtain the main example files used in the text.

There follow chapters on Macros, Suffix Rules and Commands which fill in all the details about *make* syntax and semantics. The Project Management chapter then deals with strategies for organising large projects and for overcoming some of the limitations of *make*. The last chapters cover the command-line flags and the interpretation of error messages.

Finally there are appendices containing a useful Quick Reference, a summary of common extensions to make (e.g. *nmake* and *shape*) and a list of features which may be system dependent. The features in this last section are also dealt with thoroughly in the main text, where they are clearly flagged as being system dependent. A nice feature of this summary is a set of tests for checking which features your own system supports.

In summary, I would recommend this book to all but the most hardened guru. The latter has probably written his own *make* front end anyway!

USLE Column

Julian Lomberg
Principal Consultant
USLE
London
United Kingdom

Julian Lomberg is Principal Consultant at UNIX System Laboratories Europe. He has extensive UNIX experience, and was a key developer on the UNIX System V Release 4 SPARC port.

UNIX System Laboratories announced the standard multiprocessing release of UNIX System V Release 4 - SVR4 MP - in October 1991. Julian Lomberg joins us in this edition to talk about the product.

For further information on this column, please contact Gill Smith on gill@uel.co.uk Gill is Marketing Manager at USLE.

UNIX® System V Release 4 Multi-Processor

Introduction

One characteristic of the computer market in the 1990s is the increasing use of multi-processor technology to boost performance in computer systems. Multi-processing is not new, it has been used in supercomputers since the 1960s. The trend emerging in the 1990s has been for this technology to be used in smaller systems, indeed now in 1992 a number of personal computer manufacturers whose desktop machines sell for only a few thousand dollars have now added multi-processor systems to their range. Today, if you have about $10,000 to spend, you can buy a multi-processor system.

The more widespread use of multi-processor technology in an increasingly competitive and cost-conscious computer market gives rise to some issues that did not exist ten or twenty years ago. Now for products to be accepted into the mass computer market they must adhere to standards, and this is where the multi-processing release of UNIX System V Release 4 (SVR4 MP) comes in. SVR4 MP is designed to enable users of the new generation of symmetric multi-processor systems, a term that I shall explain shortly, to benefit from the increases in throughput available from such systems without loosing compatibility with their existing applications.

The Target Architecture

There are many different types of multi-processor architectures and the one currently being exploited in low and medium cost systems is symmetric multi-processing. In this architecture, as shown in Figure 1, each processor shares the same system memory. Each processor has complete access to the whole of system memory without regard for other processors running in the system. If two processors attempt to access the same memory location simultaneously, the hardware will delay one of the processor's memory access until the other has completed it's access. This phenomenon is known as bus contention and is the reason why most symmetric multi-processor systems provide some cache memory local to each processor. Local cache memory reduces bus contention because data being used frequently by a particular processor will remain in that processor's cache, thus reducing the number of accesses that are needed to the shared memory.

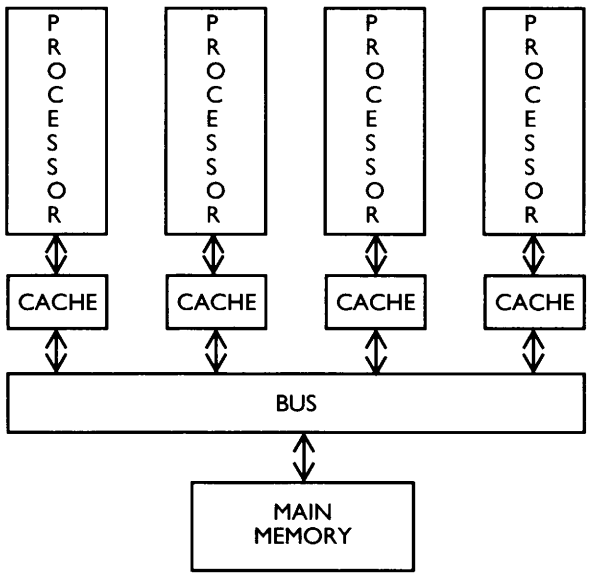


Figure 1: Symmetric Multi-processor Architecture

Local processor caches give rise to a problem referred to as cache consistency. If multiple processors are operating on the same piece of data from main memory they will each have a copy of that data in their local cache. If one processor modifies the data in its cache, then the corresponding copies of the data in the other processor's caches become invalid unless they are all updated to reflect the modification. As with bus contention, cache consistency is typically handled by the hardware; the SVR4 MP implementation assumes this to be the case.

Although we have discussed the symmetry of the target architecture with respect to memory access, the symmetry of I/O and interrupt delivery must also be considered. If all I/O devices are accessible from all processors then the architecture is said to have symmetric I/O. The alternative is asymmetric I/O where certain devices are only accessible from certain processors. An example would be a system where the main memory was accessed by the processor via a dedicated special purpose high-speed bus, but all the I/O devices are attached to an EISA bus that is only accessible from one of the processors. SVR4 MP can support either symmetric or asymmetric I/O. SVR4 MP can also support symmetric or asymmetric device-interrupt distribution. That is, interrupts can always go to one processor, different interrupts can go to different processors, or any interrupt can go to any processor.

Finally, the hardware must support a mechanism for one processor to raise an interrupt on another processor, and the processors used must support an atomic test-and-set operation for implementation of data structure locking (see later).

Design Goals

Before looking at some of the changes that were made to SVR4 in adapting it to a symmetric multi-processor architecture it is important to look at the design goals set for the system as these have influenced the design. Three major design goals have shaped the implementation of SVR4 MP:

Complete compatibility with uniprocessor SVR4. Not only does this mean that SVR4 MP complies to the same standards as SVR4, such as IEEE Posix, X/Open Portability Guide and USL's System V Interface Definition, but also that the SVR4 Application Binary Interface, and in the most recent SVR4 MP version for the 386 the Intel Binary Compatibility Specification extension 2 (iBCS2), are supported. This means that the user doesn't have to throw away his or her SVR4 applications when moving from a uniprocessor to a multi-processor implementation.

Minimal changes to uniprocessor SVR4 source code. This makes it easier to ensure the compatibility goals above are met, it also provides an easier implementation path for computer manufacturers migrating their operating system from a uniprocessor to a multi-processor base, and eases the movement of product enhancements between uniprocessor and multi-processor bases.

Ease of hardware portability. The code specific to a particular processor architecture is isolated, as is the code that is specific to a particular implementation of symmetric multi-processing such as starting and stopping individual processors and sending interrupts between them.

In addition to the above design goals, some performance criteria were also defined:

When running on a single processor, no more than 5% performance degradation compared to uniprocessor SVR4. There will always be some overhead introduced by the locking mechanism, explained later in this article, that need to be introduced for multi-processor systems. To run SVR4 MP on a uniprocessor system would not be sensible, given that the uniprocessor SVR4 implementation would be suitable, but it does provide a good way to measure the overhead introduced in making the operating system multi-processor capable.

An increase of 85% of available processor power for every processor added to the system. The ideal situation would be to achieve an increase of 100% of available processor power for every processor added, issues such as bus contention and locking overhead make this impossible to achieve in practice.

The Technology

This section examines some of the key changes that were made to the SVR4 kernel in order to adapt it for a multi-processor system. Most of these changes are completely transparent to the user and the application developer. The only changes that might effect these people are described in the final subsection, *New and Old Code*.

Scheduling

The UNIX System has always been a multi-processing (not to be confused with multi-processor) system. Multi-processing means that the system can have multiple processes active at any one time, for instance there may be two users on the system, one running an editor process and the other running a compiler. On a uniprocessor system the kernel uses a scheduling algorithm to share the one available processor between the multiple processes that are waiting to run. An available process is picked from the queue of available processes and allowed to run until the kernel decides to schedule another process from the queue. This scheduling algorithm is easily extended to a multi-processor environment. Instead of only having one processor available on which to run processes, there are now multiple processors. Each processor schedules its time independently, however they all select the processes that they are going to run next from the same queue. The scheduling algorithm on each processor will always pick the highest priority process that is on the queue, and since each processor is picking processes off the same queue the load will be evenly spread across the processors.

Data Locking

At any given time all processors will be running simultaneously. Each processor will have selected a process to run, either a user process, a system process, or an idle task. Each processor could either be executing the process itself, or else executing within the kernel. It is the possibility of multiple processors simultaneously executing the kernel code that causes problems. Changes must be made to the kernel that allow it to continue to function reliably when it is being executed simultaneously by multiple processors. The resulting kernel is said to be *multi-threaded.*

Imagine two processors attempting to increment the same variable in the kernel. They could both read the current value *n* simultaneously, add 1 to it, and then write the resulting value *n+1* back when the real value should have been *n+2*. The solution to this problem is to implement a locking scheme to prevent concurrent access to data structures where such concurrent access could cause corruption of kernel data structures. Before accessing a data structure a process must first acquire the lock associated with it, and, once it has finished its access, it releases the lock. Another process will be prevented from acquiring the lock until the first processes has released it, and hence will not be permitted concurrent access. These locks are called *mutex* (mutual exclusion) locks. The major task in multi-threading the SVR4 kernel was to protect kernel data structures using mutex locks.

Adding locking to a kernel must be done with care if optimal performance is to be achieved. A balance must be struck between coarse-grained and fine-grained locks. Coarse grain locks cover a large number of data structures. Creating locks that are too coarsely grained results in excessive contention on the locks. Processes may have to wait for a lock in order to access data that could be totally unrelated to the data currently being operated on by the process holding the lock. Setting the locks at too fine a grain, for instance on each individual variable, can mean that the kernel spends more time acquiring and releasing the locks than it does operating on the data. The granularity of the SVR4 MP locks were determined using debug and analysis tools that provide data on lock contention and other issues related to performance and possible deadlock conditions.

The mutex locks in SVR4 MP are of two types, spin locks and sleep locks. Spin locks are used for data items that are only locked briefly before being released. When a process tries to acquire a spin lock and fails it continually retries until it succeeds. Sleeps locks are used for data items that are going to remain locked for a longer period of time. When a process fails to acquire a sleep lock it is put to sleep and woken when the lock is released by the current holder. The behaviour of each lock in the system is set up at initialisation time, this means that a given lock can be switched between spin and sleep by changing the one place in the code where it is initialised.

Two other unusual features of SVR4 MP locks are also worth mentioning, both are motivated by the design goal of minimising the number of changes to the uniprocessor SVR4 source code.

It is possible, due to the recursive nature of the SVR4 kernel, for a process to try to acquire a lock that it already holds. In many multi-processor systems this would cause a deadlock, however this is permitted in SVR4 MP. The system keeps a count of how many times a process acquires the same lock so that when it is subsequently released the correct number of releases can be performed.

The second unusual aspect of SVR4 MP locks is their behaviour when the holding process goes to sleep. A process that goes to sleep whilst still holding a lock could have a disastrous effect on system performance. Many multi-processor systems solve this problem by ensuring that processes do not go to sleep whilst holding locks that other processes are likely to need. To implement this solution in SVR4 would require many changes to the code and hence an alternative solution was required. In SVR4 MP all locks held by a process are automatically released when that process is put to sleep and then automatically re-acquired when the process is woken up. The reason that this solution works in SVR4 is because, even for the uniprocessor SVR4 implementation, sleeps are significant events because they will cause a context switch and hence another process to take control of the processor. This means that the uniprocessor SVR4 code already ensures that kernel data structures are in a consistent state before a process is put to sleep.

Processor Private Data

There are certain variables in SVR4 that are processor specific such as the interrupt mask and the details of the process currently running on that processor. In SVR4 MP these variables are grouped into processor-private regions, one per processor in the system. Each processor's virtual memory map is set up so that it's processor-private area, at the same virtual address on each processor, is mapped to a different physical address where the processor variables for that processor are actually stored. The numerous references to processor specific variables scattered throughout the SVR4 code did therefore not have to be changed.

New and Old Code

Locking interfaces have been added to the new DDI/DKI specification to enable multi-threaded device drivers and streams modules to be written. Device drivers and streams modules are linked into the kernel address space and may be running simultaneously on more than one processor. This means that new code should use the new DDI/DKI locking interfaces to protect critical data structures.

It might not be possible to modify all device drivers to be multi-threaded. Perhaps you have an SVR4 driver for a peripheral that was only supplied in binary form and hence cannot be modified to include the necessary lock

protection. SVR4 MP supplies support for single-threaded device drivers that conform to the original DDI/DKI specification. It is possible to bind a device driver to a single processor. This ensures that the specified device driver will only ever be allowed to run on the single specified processor. This means that the device driver can run secure in the knowledge that no other processor will be executing the same code and hence there will be no concurrent access to the data structures it is using.

Summary

SVR4 MP provides a standard operating system environment for multi-processor systems. Complete compatibility with uniprocessor SVR4 makes the move to a multi-processor environment transparent for the users and preserves their investment in application software. Due to some innovative new locking mechanisms, the internal changes to the SVR4 kernel have also been minimised allowing OEMs to adopt this new technology with a minimum of effort.

Abstracts

Here are the abstracts from the EurOpen Spring conference in Jersey on 6-9 April 1992. Copies of the proceedings may be obtained from the secretariat at £25.

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

Software Pre-Porting

Marc Poinot
SEXTANT Avionique
Vélizy-Villacoublay
France

This paper describes the three levels we have identified in a pre-porting process. The Palas-X SDE has been developed, ported and installed taking into account these levels: the source code portability, the development / porting platform portability and then the installation / execution platform portability. Our idea is that there always are software services for which we have no or partial information, so we have to identify these parts, encapsulate them and keep in mind that there are potential problems in there.

Porting Under UNIX – Problem Areas and a Proposed Strategy

Kenneth J. ChanDavid Jackson
University of Liverpool
Liverpool, England

E-mail {kjc | dave}@compsci.liverpool.ac.uk

Much has been written on software portability and on guidelines for writing highly portable programs. In a large number of UNIX environments – educational establishments being good examples – the members of software support teams know only too well that much of the software that they have to port either does not follow such guidelines, and, even when good programming practices have been adhered to, packages often (by necessity) contain a substantial amount of system-specific code. In such circumstances, the accumulated wisdom derived from the experience of porting many items of software over many years is drawn upon to provide the insight required to deal with a particular package. To the casual observer, the porting process can appear largely unstructured and *ad-hoc*, and a generalised methodology may seem almost impossible to define.

An initial aim of this paper, then, is to impose some sort of structure on the list of problem areas that porting specialists encounter in their activities. This is achieved by examining the typical difficulties that arise due to the particular "flavour" of UNIX that is being used, and also the typical problems that are found during each of the stages of compilation, linking and execution when installing a package. This broad classification is then used as a springboard for developing a strategy for porting software on UNIX systems. Throughout this paper, we present various examples of packages that have caused problems when ported to our particular departmental computer system, but other institutions will undoubtedly have suffered analogous tribulations.

Certifying Binary Applications

Donald A. Lewine
Data General Corporation
Westboro, USA

E-mail lewine@cheshirecat.webo.dg.com

The goal of creating a binary operating system interface standard is to define a common configuration which enables the executable binaries of application programs to be moved between computer systems. This paper gives an overview of the 88open standards which allow a POSIX or SVID conforming program to be compiled into a vendor independent binary. The bulk of the paper is dedicated to certifying binary compatibility. How does one know that an application meets the standards?

This paper describes two packages for certifying binary compatibility: ITS/88 and ACT/88. ITS/88 verifies that the Operating System supports all of the features of the standard. ITS/88 uses conventional Software Quality Assurance techniques. ACT/88 verifies that the application strictly conforms to the standards. ACT/88 uses a static, lint-like, analysis to look for errors in the code and dynamic run-time analysis to look for non-portable behavior. I believe that ACT/88 is unique technology and different from any known quality assurance tools.

Applications POSIX.1 Conformance Testing

Derek Jones
Knowledge Software Ltd,
Farnborough, UK

E-mail derek@knosof.co.uk

The Standards for POSIX and C were designed to enable the portability of applications across platforms. A lot of work has gone into checking compilers and environments for conformance to these Standards, but almost nothing has been done to check applications conformance. The incorrect assumption being made

that the development compiler will warn about any construct that needs looking at. This paper discusses a tool that checks applications software for conformance to these Standards at compile, link and runtime as well as the library interface. Any application that can pass through this checker without producing any warnings is a conforming POSIX program and a strictly conforming C program.

X is the Worst Window System – Except For All the Others

Berry Kercheval
Protocol Engines, Inc.,
Mountain View, California, USA

E-mail berry@pei.com

The current state of the art in window systems is reviewed. Some survey of the field is mixed with the authorspersonalexperiences.

Against User Interface Copyright (October 20, 1991)

The League for Programming Freedom

E-mail league@prep.ai.mit.edu

This paper describes how the copyrighting of software interfaces threatens programmers freedom to write software. It also shows how interface copyright obstructs progress even as it denies users the benefit of competition. In addition, interface copyright benefits mainly those who have already been so successful as to set a de-facto standard – primarily large companies. Conclusion: interface copyright is unsound public policy.

Against Software Patents (February 28, 1991)

The League for Programming Freedom

E-mail league@prep.ai.mit.edu

This paper describes how the extension of patents to cover software techniques, algorithms and features threatens programmersfreedomto'write software. It explains why patents in the software field yield comparitively little benefit in the form of additional published techniques, algorithms and features, while greatly hindering the persuit of software development; and, in addition, how they favor the largest companies against all others. Conclusion: the application of patent law to software is unsound public policy.

Practical Problems with Porting Software

Brian ODonovan
Digital International B.V.
Galway, Republic of Ireland

E-mail odonovan@ilo.dec.com

This paper describes some of the problems that are encountered when porting software between truly heterogeneous systems. As a practical example it describes the problems that were encountered when porting a large software system from VMS to ULTRIX.

A Health Information System based on UNIX – Client-Server called PHOENIX

Dr. Reinhard Koller
Amt der O Landesregierung, Austria

PHOENIX stands for **P**roject **HO**spital **E**nvironment **UNIX** and is the logo for a project organized by the Government of Upper Austria for 16 hospitals. The project started at the beginning of 1990 and phase I will take 3 years. Phase II will start in 1993.

The project is completely based on the requirements for portability, 4GL Tools, relational databases and standards.

So the system environment consists of:

- Motorolas88100DualRISCServer'
- UNIX System V.3
- TCP/IP and NFS
- Informix-Database, Informix-4GL
- Uniplex 7

At the moment the application software includes a patient care system (evidence and administration of patients), office automation, a staff information system, a laboratory and a pathology information system.

In this paper I will tell you about our relevant criterions we have considered in building these complex and large (three of our systems will consist of about 200 terminals each) information systems. I will show which strategic requirements and measures are necessary so that based on this concept the information systems can be enlarged and extended for further applications and necessities and the costs can be reduced by "downsizing" and using client-server models.

This is especially considered under the aspect of the importance and influence of standards, connectivity and multivendorship.

I will finish by giving you an outlook of our plannings for phase II starting in the Spring of 1993.

The Portability of GNU Software

Joseph Arceneaux
The Free Software Foundation
San Francisco, CA, USA
E-mail jla@ai.mit.edu

Michael TiemannD. V. Henkel-Wallace
Cygnus Support
Palo Alto, CA, USA
E-mail {tiemann|gumby}@cygnus.com

In June of 1987, GNU C version 1.0 was released by the Free Software Foundation. Since that time, it has been ported to about 50 host environments and generates code for about 20 machine architectures. In many cases, GNU C was ported to new platforms by

volunteers, working with incomplete information, more rapidly than vendor-sponsored ports of AT&Ts"portableCcompiler".

This talk will describe 4 portability case studies based on well-known GNU software packages: Emacs, GDB, GCC, and BFD. The case studies will present the initial design specifications, the evolution of the design over time, factors that influenced (or mandated) changes to the package, and experience in applying the lessons learned in the design of subsequent programs.

Portability in a Research Environment

Andrew Hume
Howard Trickey
AT&T Bell Laboratories
Murray Hill, New Jersey, USA

E-mail andrew@research.att.com

Software developed within a research environment is both easier and harder to make portable than software developed within a industrial environment. Projects utilising unique aspects of the underlying environment may prove difficult or expensive to port. On the other hand, access to the system source can sometimes solve portability problems locally. This report describes some of the approaches used within the Computing Science Research Center at AT&T Bell Laboratories for writing portable software.

Inter-Fashion Portability

Barry Shein
Software Tool & Die
Brookline, Massachusetts, USA

E-mail bzs@world.std.com

Fashion is an invention of civilization to make us unhappy with everything we already own. Many fine programs and shell scripts seem dated simply because they happen to also work on a printing terminal or otherwise do not use one of the latest, fashionable window systems.

This paper will explore a set of simple primitives designed to be executed from shell scripts (or via the simple insertion of popen or system library calls within existing C programs) which manage various user input/output requirements utilizing modern window systems. This is not a research paper per se, but rather a position paper attempting to outline some views and implementations I have been developing over the past few years, and to try to raise some new questions to the community which have been puzzling me.

I will also speculate on why these programs either haven't been written before or why they are not commonly distributed. Fashion eschews cheating by retrofit: Why fix a hemline when you can buy a whole new dress?

Camera: Cooperation in Open Distributed Environments

Gert Florijn, Ernst Lippe, Atze Dijkstra
Norbert van Oosterom, Doaitse Swierstra
Software Engineering Research Centre – SERC
Utrecht, The Netherlands

University of Utrecht
Utrecht, The Netherlands
E-mail camera-info@serc.nl

The next generation of end-user computing environments will be marked by two features: they must be able to operate in a rapidly evolving distributed environment and they must provide support for cooperation among users. The Camera system, a platform for defining (end-user) workspaces, provides a particular approach to these issues which is based on visible distribution of data and explicit communication among users.

This paper describes the architecture of the Camera system in some detail. Among the topics addressed are the Object Management System, which replaces the traditional file system as a data store, the two-level architecture which introduces a distinction between development activities and the products created during these steps, and the mechanisms for communication and cooperation, which allow users of different, isolated systems to exchange progress and merge the results of different activities. The paper includes a description of a prototype implementation on UNIX, and indicates some of the work in progress.

Distributed System and Security Management with Centralized Control

Chii-Ren Tsai
VDG, Inc., Maryland USA
Virgil D. Gligor
University of Maryland, Maryland USA
E-mail {crtsai|gligor}@eng.umd.edu

We have designed and implemented a prototype of distributed system and security management for AIX Version 3 on the RISC System/6000 by using an experimental secure remote procedure call (RPC) mechanism based on Network Computing System (NCS) and Kerberos. The prototype consists of distributed SMIT (System Management Interface Tool), distributed audit and access control list (ACL) management for AIX systems. Distributed SMIT can manage user accounts, file systems, devices, networks, spoolers and system configuration. Distributed system security management, which includes distributed audit and distributed ACL management, allows the distributed system security administrator to turn on/off auditing, perform audit system management, analyze audit trails and set ACLs on a per-file, per-directory or per-application basis. Based on the experimental secure RPC mechanism and Motif widgets on the X window system, we designed and implemented a high-level, protocol-transparent, integrated interface for the prototype of distributed system and security management.

Book Reviews

UNIX The Complete Book A guide for the professional user
Jason J Manger
Sigma Press
ISBN 1-85058-219-X
Paperback, pp 453.

Reviewed by Patrick Fogarty

E-mail fogartyp@dcu.ie

Jason J Manger has made an excellent effort at an impossible task. A complete guide is something that can be approached but not achieved. The title of this book is ambitious and the content lives up to it as well as can be expected.

Previous knowledge of UNIX is not assumed but some experience using an operating system, such as DOS, would be of great advantage in reading this book. I would further suggest, before reading the chapter UNIX-The Multi-user System, the reader familiarise him/herself with the rudiments of C programming.

The book was well put together - chapters being self contained as much as possible. Each chapter does provide a relatively complete reference on its subject. Topics covered in chapters include :- Logging in, File Systems & File Structures, UNIX Shell Utilities (UNIX commands) & Shell Variables. I was impressed with the treatment of the UNIX editors, which is liberally sprinkled with examples. There are also chapters on Communication, Systems Administration, Security and Shell Programming. One criticism is that chapter 3, which deals with UNIX as a Multi-user system and X-windows protocols, is out of place and could be left to a later stage of the book. As it stands it is intimidating for the UNIX beginner. Chapter 3 also requires some knowledge of C programming to be appreciated fully. Printing and presentation is good with the exception of several typographical errors which, while irritating, were generally of little consequence.

The addition of a further section on the relationship between the UNIX Kernel and Application layers would improve future editions. Also I would like more detail on device drivers and sockets.

A good glossary is a must and this book supplies an excellent glossary. There are also questions and answers which allow the book to be used as a text for UNIX courses.

Overall there is lots of information in this book and it will definitely act as a good reference text, on my shelves, for the future.

P.S. I had to fight to keep my colleagues hands away from this book !!

THE C BOOK Featuring the ANSI C Standard (Second Edition)
Mike Banahan, Declan Brady, Mark Doran
Addison-Wesley Publishing Company, Inc.
ISBN 0-201-54433-4

Reviewed by Mick Farmer, Birbeck College

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

This book is an upgrade of Mike Banahan's original C book which came out in 1988, before the official ANSI standard for C was published in 1989. The authors have taken the opportunity of a second edition to ensure that the book conforms to the new standard (where there may have been some earlier ambiguity) and to include additional material based on the courses taught by the Instruction Set.

The book is not an introduction to programming. It is designed for programmers with some experience of modern high-level procedural languages and concentrates on those things which are special to C. There are ten chapters covering the usual topics such as functions, arrays and pointers, the preprocessor, etc. It is good to see one chapter devoted to specialised areas of C. Included here are discussions on declarations and definitions, the new keywords `const` and `volatile`, and sequence points - an important topic in the standard, though rarely mentioned in other C books. Each chapter concludes with a summary and, in some cases, a few exercises.

As mentioned in the preface, many people who know C are interested in the new standard and how it affects existing C programs. This book highlights where the old and new features of the language differ significantly, because the authors realise that programmers will be working in an environment where the old and new language mix for a few years to come. This is emphasised in the chapter on libraries where the new topic of localisation (an ugly word) is covered in detail.

In conclusion, this is a good book for a reasonably experienced programmer who wants to know about ANSI C. This second edition ensures that the standard is followed closely and faithfully.

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 Peter Russin	chairman@EurOpen.se
FUUG Kai Keinanen	Kai.Keinanen@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	sec@puug.pt
SUUG Alexanter Reznitsky	air@hantarex.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	Public Relations
Cormac Callanan	Cormac Callanan
eunet-hls-wg@eu.net	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	Postal
EUnet	EUnet
NIKHEF	c/o NIKHEF
Kruislaan 409	Postbus 41882
1098 SJ Amsterdam	1009 DB Amsterdam
The Netherlands	The Netherlands
Telephone +31 20 5925124	
Facsimile +31 20 5925155	

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

Addresses of national backbones

Austria	EUnet EDV Dienstleistungs Gmbh A-1010 Vienna Schottenring 33
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 Datalogisk Institut (DIKU) Universitetsparken 1 DK-2100 Koebenhavn OE
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 I-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 uuga@tuvie.can.ac.at

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 0 400 2043
E-mail fuug@fuug.fi

France - AFUU
Jean-Luc Anthoine
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 anthoine@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
Annrai O'Toole
Irish Unix Users Group
PO Box 3396
Dublin 6
IRELAND
Telephone +353 1 7021543
Facsimile +353 1 772204
E-mail iuug-exec@dec4ie.ieunet.ie

Italy - i2u
Ing Carlo Mortarino
i2u
Viale Monza 347
20126 Milano
ITALY
Telephone +39 2 2520 2530
Facsimile +39 2 2520 2508
E-mail i2u@italy.eu.net

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
NUUG Sekreteriat
PO Box 4407
0403 Oslo
NORWAY
Telephone +47 2 952092
E-mail nuug@multix.no

Portugal - PUUG
Legatheaux Martins
Avenue 24 de Julho, nº 134, 7º
Lisboa
PORTUGAL
Telephone +351 1 395 06 42
Facsimile +351 1 397 18 76
E-mail sec@puug.pt

Spain - UUES
Usarios de UNIX de España (UUES)
Dpt. Ingenieria de Sistemas Telematicos
E.T.S.I. Telecommunicacion
E-28043 Madrid
SPAIN
Facsimile +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
E-mail kuz@ivann.delta.msk.su

Yugoslavia - YUUG
Milan Palian
ICOS Ljubljana
Titova 118
YU-61000 Ljubljana
Slovenia
YUGOSLAVIA
Telephone +38 61 181282
Facsimile +38 61 183546
E-mail Milan.Palian@ninurta.fer.yu

Associate Members

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

EurOpen Secretariat

Owles Hall
Buntingford
Hertfordshire SG9 9PL
United Kingdom

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