NEWSLETTER

EUUG

European UNIX® systems User Group

Volume 9, No. 3
Autumn 1989

CONTENTS

EUROPEAN UNIX® SYSTEMS USER GROUP NEWSLETTER

Volume 9, Number 3
Autumn 1989

Editorial Team:

Philip Peake
Publications Executive
CETIA
9 Avenue de Laponie
ZAC de Courtaboeuf
91940 Les Ulis
France
peake@inria.fr

Alain Williams
Editor
Parliament Hill Computers Ltd.
7 Prospect Street
Caversham
Berkshire RG4 8JB
U.K.
addw@phcomp.co.uk
Tel: +44 734 461232
Fax: +44 734 373194

Typesetting:

Laura Dekker
The Instruction Set Ltd.
City House
190 City Road
London EC1V 2QH
U.K.
laura@inset.co.uk

Printed by:
Rank Xerox (Copy Bureaux) Ltd.
68-74 Rochester Place
London NW1 9JX
U.K.

Published by:
The EUUG
Owles Hall
Buntingford
Hertfordshire SG9 9PL
U.K.
euug@inset.co.uk (...!mcvax!ukc!inset!euug)
Tel: +44 763 73039
Fax: +44 763 73255

ISSN 1011-4211

Editorial

Alain D. D. Williams
addw@phcomp.co.uk

Parliament Hill Computers Ltd

Vienna

By the time that you read this you will only have a short time to ensure your place at the Autumn EUUG conference in Vienna. You will find a list of tutorials and a provisional technical time table on pages XX.

The line up looks good, a flavour of EUUG conferences can be gained by looking at the abstracts for the papers delivered in Brussels this spring – see page xx.

I will be there, will you?

FORTRAN Benchmarks

In the days when I started programming to produce a portable program meant writing in FORTRAN – BCPL was never widespread enough and C was B. FORTRAN is still a popular and important language.

Nhuan DoDuc looks at portability from a direction that will probably surprise many a Unix and C hacker – although it shouldn't.

Strategy Meeting

In early May an EUUG strategy meeting was held in Dublin. The job of those present was to refine the direction of the EUUG, ie what is the purpose of the EUUG.

Much was discussed, Norman Hull present a viewpoint on what happened on page xx. Read it, it is important, if you have strong feelings about what is mentioned don't keep quite write and tell me or any member of the EUUG executive.

Some of you will soon receive a survey letter from the EUUG, please help us to help you by completing it and returning it.

Group Reports

In this issue you will find the regular reports from the Netherlands and UK groups, and the first from the new Portugese group.

Did you know about the Unix group in Israel? You can find out what they are up to on page yy, just before the regular report from Usenix.

Regular Columns

William Roberts continues his tour of the World of Windows and examines how text is displayed on bitmap screen, it is not as easy as it looks.

Janet Davis has some guest writers who delve a little into some aspects of the OPEN LOOK toolkit, part of Windows AT&T style. Just because it is AT&T doesn't mean that it will become standard, life is not that simple. Dominic Dunlop gives you his view on page xx as to what the confusing standards world has been up to recently; he will also be speaking at Vienna – so if you don't understand what he has written you will be able to ask him yourself.

After you have spoken to Dominic you can collect your conference tape. I don't know what Frank Kuiper will have in store for us, but you can see what he has done so far on page ff.

Most of this software will have been taken from public sources, Donal Daly (page cc) will probably have reviewed some of it for you. If he hasn't written about your favourite bit of software, knock off a a few words about it and send it to him – he does a good job, but could do with a hand.

Laura Dekker

This is Laura's last newsletter, she is off back to college (UCL) to get a M.Sc., we wish her all the best.

Compatible FORTRAN Compilers

Nhuan DoDuc
ndoduc@framentec.fr

Framentec
Paris La Defense
France

DoDuc loves ice creams, likes to discuss cosmology and cosmogony, dreams of running benchmarks, uses FORTRAN more than COBOL, believes in C more than in Lisp, lives within UNIX more than within MS/DOS. He is an active member of AFUU where he participates in many working groups (e.g., Benchmark, Station de Travail, Calculateurs Scientifiques). Since his graduation in (theoretical) physics, he works within the DP activities of the French nuclear company Framatome, and presently with one of its affiliate, Framentec, an A.I. dedicated company.

Standardisation and Compatibility are real facts not limited to hardware or operating systems. This paper will browse through the FORTRAN compiler scene, as seen from an user's point of view, with some emphasis on MS/DOS and UNIX offerings.

Once upon a time, there were specific hardware families, closed operating systems and proprietary software such as compilers, etc... and of course very strictly regulated relationships between the different partners, clients and vendors. Then a strange and long awaited event occurred:[1] the new actors were the now standard microprocessors and standard operating systems; namely the Motorola and Intel *et al.* chips and boards and systems based upon them on the MPU side, and their companions, MS/DOS and UNIX on the OS side. Furthermore their arrival has greatly changed the scene since then. The modifications are probably definite and irreversible to the behaviour of every player of the trade, users and suppliers alike.

1. but probably neither predicted nor expected

Why another FORTRAN compiler?

Certainly, there are many explanations for the appearance of compatible compilers, i.e., equivalent compilers within the same environment; some of them are economic and others are similar in nature to those prevailing in the domain of hardware compatibility

Two other explanations, however, are worth mentioning: (1) some native, or bundled, compilers are frankly so bad, in any sense of the term, that they should be considered a mockery; (2) some people with much of experience in writing compilers on mainframes are only too glad to try their entrepreneurial imagination on friendlier or more popular machines, and please don't forget that this is a really hard job, which, up to now, is somehow underestimated by most systems designers and integrators.

This paper is *not* a survey of the domain, it cannot be thorough or complete. Such a survey remains to be written. It is only a narration by a good (uh!

uh!...) FORTRAN user who wishes to limit himself to existing or locally available products, more precisely in France. Also, it will cover essentially the MS/DOS and UNIX offerings.

A Precursor

The birth of a compatible compiler is only possible if the (operating) system is widely used and if the bundled compiler has some glaring shortcomings, in functionality or in performance. One good example is the WATFOR/FIV family from the University of Waterloo. People are not exactly happy with the debugging capabilities of the existing IBM's FortG1 or Fort 4.H.extended, and so built their own compiler, the ANSI66-compliant WATFOR and its ANSI77 counterpart WATFIV. With much deserved success, although not quite widely used outside academia, it has been ported to the VAX/VMS and recently even to the MS/DOS world.

The MS/DOS World

Much like Venus, MS/DOS was born *naked*, and the first compilers were in fact hazardous things to use. Of course here Microsoft reigned supreme, and not until 1985 was version 3.31 capable of functioning correctly, especially in the support for ndp (e.g., 80x87) codes. People were in such a desperate mood that IBM was able to market its version 1, which is an early version of the Microsoft product; however the product was so lamentable that people became really desperate until the appearance of IBM Professional FORTRAN 1.0, which is none other than a licensed version of a Ryan McFarland product. This product and the version 2.11 marketed directly by Ryan Mc Farland proved to be good, robust and performant, proof of the real know-how of their creator. For some time, the version 2.11 was acclaimed as the most performant compiler for big fat applications downloaded from mainframes, and especially in the domain of numerical computation.

Around 1985, many compilers arrived on the market, most of them quite quickly and thoroughly evaluated by many serious reviewers, another proof that they were really awaited, being part of the processing departmentalisation phenomena. In this serious and hard arena, too, there is of course no clear winner[2] but there are excellent compilers and there are average ones. As with any other product, the user has to make up his/her mind regarding his/her specific needs and preference.

The MS/DOS World: the Leaders

The fastest (and of course still accurate) in execution time is still the Ryan McFarland product whose authors were busy developing IBM mainframe compilers since the sixties. However compilation time is incredibly long and probably prevents its use in the development stage where global optimisation is not required. Recently Austec, the trade name of the new merged company, released version 2.4, with some improvement in compilation time, and most notably RM/Forte which is a really impressive programming environment, with menu-driven and multi-window display, source-code manager, (UNIX)make-like capability, context-sensitive help... all the niceties that you always wished to have for your daily work...

The fastest (and of course with excellent and clearly understandable diagnostic message) in compilation time is still, and by far, the Lahey product present on the scene since 1984. One particular functionality deserves to be mentioned: the Source On Line Debugger; the compiler generates *.SLD files (along with the *.EXE and *.LST files) which store everything needed for a full debugging process: this original idea allows the simple deletion of *.SLD when the debugging is over (and if disk space runs low, which invariably *is* the case); the *.EXE files do not contain any debug instruction that may slow down execution or that need to be stripped off as usually done elsewhere. In summary, the Lahey product, with many extensions targeting the mainframe (e.g., IBM and Vax/VMS) environment as well as the Dos environment is quite probably the best choice for the early development or porting stage of big size code.

Microsoft, the front player on the scene, has never given, and still does not, any thought to FORTRAN. Logical to its business, it has put emphasis first on Pascal then on C, the two most

2. e.g., no compiler benchmark, at least not yet and quite probably never!

popular languages which successively were used to write the FORTRAN compiler, but this should not surprise us. As a matter of fact, the version 3.31 has a definite 'parental' relationship with Pascal, while the 4.xx version is simply the FORTRAN front-end to the new generation of C-based compilers. Clearly Microsoft has taken the challenge about developing a whole new compiler family (e.g., with *front end* (that analyses and parses the source language statement into an intermediate language)—*optimiser* (that does the really hard home work)—*code generator* (or back end, that generates native code for a specific processor)), using the newest and latest proven technology. Nothing needs to be said about these compilers except that, like IBM products, they bear the Microsoft trademark. They do not show any notable feature: the version 4.xx now seems to be as fast as other competitors, the CodeView debugging facility is acceptable but this is *not* specific to the FORTRAN compiler. Perhaps we should mention the availability of different numerical emulations or code generations... At the time of writing, Microsoft has just announced version 5.0 of its Optimising Compiler.

The MS/DOS World: the Others

Digital Research's latest product version is 4.1. This is a plain vanilla compiler without specific plus or minus. However, if an anecdote must be told about it, the author may mention that (but this should not be taken as a criticism against the product), that when emulating the 80x87 extended (e.g., 80 bits) precision, it is much slower than any competitor.

Prospero Software supplies two compilers: Pro FORTRAN 66 and Pro FORTRAN 77, presently respectively at version 2.15 and 1.26. Another plain vanilla family of compilers, except that the family does cover many other *non MS/DOS* environments such as CP/M and CP/M-86, Sinclair QL. A very interesting point: the version 2.13 of Pro FORTRAN 77 for GEM for the Atari ST family (and with support for the 68881) is apparently the *only working* FORTRAN compiler for it.

For the sake of completeness, Ellis Utah FORTRAN must be included. While it is the cheapest compiler, it is also the only one which *does limit* the code to 64Kbytes and the data to 32Kbytes, clearly an unacceptable constraint for real scientific programming. Lastly, the author has seen mentioned a Supersoft FORTRAN-66, version 1.07, but has no details about it and has not yet seen it. Just for the record.

The 'Extended' MS/DOS Compilers

The only too well known 640Kbytes usable memory limit has proven to be too much for users accustomed to mainframe addressing capabilities. From another point of view, it's too tempting to exploit the full functionalities and power of the i286 and i386[3] and Weitek chips. This conjunction results in a new category of compilers that can still be used within *good old* MS/DOS, but through some *DOS extenders* can tap, at run time, the whole performance of these 32-bits chips.

At the present time, the author has found four of them. The oldest and probably the fastest comes from Silicon Valley Software[4] already well known for their good products for many UNIX boxes. Microway, the specialist in numerical processing for PC, created the first MS/DOS compiler that was able to generate code for the Weitek chip, which resulted in an incredible jump of execution speed in heavy number-crunching problems. Polyhedron Software markets FTN77/386, developed by those men from Salford University who created the FTN77 compiler for Prime machines. While these three compilers specifically target i386 machines (and require i387 co-processors), Lahey offers two *protected mode* products: F77L-EM/32 is meant for i386 machines while F77L-EM/16 can already fully exploit the i286 characteristics, for instance 15Mbytes (instead of 32Mbytes) of usable memory. It's a tribute to Lahey's competence to mention that their most famous feature, namely compilation speed, is still present at least with F77L-EM/32 as tested by the author. Lahey and Salford also announce support for Weitek chips, a real proof of good democratisation in the crowded MS/DOS world (the author thinks that SVS FORTRAN does support Weitek chips but was not able to check or test this).

All these compilers deserve high marks: the products are all solid, reliable and performant. "Mainframe functionalities and performances on

3. and now the i486 too!
4. just merged with Trio Software System

a PC'', we might say.

The OS/2 Picture

The future of OS/2 might be as bright as its present is currently dark: the author is not imprudent in predicting a bleak future knowing that neither IBM nor Microsoft are dead yet. But one must recognise that: (1) there is no incentive (e.g., only promise and many advertisements, but no action) from either of them to push users to migrate toward OS/2, (2) OS/2 offers virtual memory management while any dos-extender uses real memory, (3) UNIX is real, sound and safe, and if statistics on very small data have any value, the author must state that in every case he knows of where escape from MS/DOS (or downloading from IBM mainframe) was mandatory, UNIX was chosen.

So just for the record, one can mention some 'paper tigers': IBM FORTRAN/2 and Microsoft 4.1, as well as some *tigers-to-be*: Austec Ryan McFarland, Prospero, Lahey...

IBM FORTRAN/2, up to now version 1.0, like IBM Professional FORTRAN, is a licensed product from Austec (Ryan McFarland) while Microsoft is the same version 4.1 that can be used in any of the two OS. These two compilers are unsensational which should cause you not to regret not going the OS/2 way. At least not yet.

The UNIX World

UNIX originally blossomed in the world of the mini-computers, i.e., between the mainframe world where the compilers are usually supplied with the hardware without much choice left to the user, and the micro world, where none is provided to the casual user. Another specific point is the existence, within any UNIX software source, of a f77 compiler.[5] However this compiler and BSD's are particularly ineffective or inefficient on many systems, therefore their existence means next to none.

5. ATT recognises that f77 is part of any SystemV-compliant release, however not every supplier acknowledges this fact, perhaps wishing to push their own third party compiler(s)? Is Sun OS 4.x SystemV-compliant?

The UNIX World: the Main Players

A commonality exists among users of Sun, DEC, HP, Apollo... machines, which traditionally remain linked to the bundled compilers, as they did when they were with the old mainframe. Most of them dare not explore anything new or exotic alternatives, which of course may be risky. Credit must however be given in that these manufacturers have tried their best to supply a correct or acceptable product. However system integrating and compiler writing are clearly two distinct businesses, and so fortunately it may happen that users will get some choice.

If HP and Sun compilers are good and sound products, there is no excitement in using them. The HP compiler seems to have reached its 'cruise' performance long ago and so no big improvement, in functionality or in performance, is expected (except towards Vax/VMS extensions). The Sun compiler is currently at a critical stage in its life. F77 Revision 3.2, BSD derived, is the first and foremost compiler from Sun still widely used on many Sun 2 and most Sun 3; a very honest compiler, usually accepted because it is bundled with the Sun OS, without question by almost all users who don't care, or don't dare to care, about better alternative. With the Sun 4 and then the Sun 386, a major effort was set up to write f77 1.1 which has now evolved, with the necessity to support different combinations of MPU (68K,i386,Sparc...) and FPU (Sky board, Weitek, Texas...), to f77 1.2. Clearly the imperative of compatibility across too wide a range of chip results in a heavy burden for the manufacturer for whom FORTRAN has never been a high priority.

The ULTRIX user has at the present time 3 compilers. If his machine is a DEC Station 3100, or a DEC System[6] then his only compiler is none other than the version 1.31 from Mips. If he works on a Vax architecture machine, then he has two compilers available, one originating from BSD that he *can* put aside in his private museum or vault, another which *is* the well known compiler from VMS, that he *will* use for daily work as he would do on VMS machines. At last, it is not too early for ULTRIX users to get permission from

6. 3100 too!, the *twenty* is long dead gone five years ago.

DEC to work seriously: *les vacances sont finies!*.

The Apollo compiler is a class of its own, a living process with continuous performance improvement at each major release. The compiler is closely bound to the Motorola instruction set and exploits every advantage of the Motorola chips. A definite proof is contained in the way it uses the FPA chip (which is a Weitek 3164). Any other compiler, asked to generate code for the Weitek chip, will faithfully and blindly give the execution of transcendental functions to the Weitek math library, whereas these functions being wired in the 6888X (and also in the 80X87) should be executed there. Perhaps the rationale for not doing so is the necessity and the ability for such an executable file to run even in the absence of the 6888X chip? This deed, with an old and dead exception, to the best of the author's knowledge, is unique among FORTRAN compilers. This and other similar facts auger well for the overall performance[7] of the pair compiler + Prism processor in the future (alas, if any).

In the realm of RISC processing and out of any market-driven and biased debate, the compiler is much, much more than a system utility, and many RISC-processor-based systems suddenly appear much less advantageous simply because of the bad behaviour or poor performance of the compiler. Unlike many 'CISC' compilers which are usually written once their system is up, 'RISC' compilers could not be treated as an added-value gadget; it's clear that when you just translate '*(existing)* machine-independent intermediate representation code' into specific code for a *new* processor, you are *not* and you *cannot* completely exploit and harness the full potential of this new processor. But to 'parallelise' the design of the processor and of the compiler is a very risky process and, in having done so, Mips, which is now releasing version 2.0 of its f77, is harvesting fully deserved compliments. Now this does not mean that Mips' compiler is bug-free: give me a compiler and I'll show you a list of bugs (sorry, unbelievable constructs of code that is sure to fool any honest compiler) but then this is one of the two compilers (e.g., with Apollo Prism's) for RISC processor that I would recommend for your evaluation.

7. just compare the two releases 10.5(15) and 10.7(21)!

The UNIX World: the Third Parties

For most of the other manufacturers, concern about a FORTRAN compiler is less than nil: after all, FORTRAN may not be their main business. They just port or license third-party compilers. Some of these are enumerated below.

Greenhills compiler family, also marketed under the trade name Oasys, and now just merged with it, is certainly the most well known if only because it is widely used in many environments. Since the BSD and the ATT f77s are too bad products, any integrator who doesn't bother about creating his own compiler would just look for a compatible compiler, and Greenhills' gf77, now generally at version 1.8.x, is clearly the most frequently selected, a tribute to its quality and performance level. To name a few systems that have chosen gf77: Apple, Edge, Bull, Intergraph, Ridge, AMD... and a few compilers that have grown out from gf77: Unisoft, Microway, Mercury... Perhaps the main reason for this success is, beside the fact that the family grows around the triplet (front end - optimiser - code generator), that gf77 already works on more platforms than any other competitor: 68K, i386, NS32000, Weitek, Ridge, Clipper... and soon on the i860 too.

SVS compiler, just as good as Greenhills's, is somehow less widely seen, probably because of its way of selling the product (e.g., no direct sale); we may encounter it on 68K machines like NCR Tower, Definicon boards... on i386 machines like Prime EXL... Nkr compiler is a recent competitor and up to now seems to stay only within 68K machines: Sun 3, Mac (A/UX), Sony News, Integrated Solutions... LPI although also a newcomer is also offering a multi language family that seems quite promising; the product for the Sun 3 is at version 3.00 and still does not support the FPA (Weitek chip); the product for i386 machines running under XENIX is frequently bundled in many offerings, e.g., SCO, Data General Dasher... Philon is the last player with offerings for 68K machines (NCR Tower, Plexus... and of course Sun 3), but also for CCI and ATT's 3B series (DEC ULTRIX too?).[8]

8. There are rumours about possible merging between Philon with another well known compiler company.

TopExpress is also a late comer for which not much is known about.

So You Want MS/DOS and UNIX Altogether?

Some compilers may have versions for MS/DOS *and* UNIX. The best known is Ryan McFarland version 2.11 probably because it is old and runs under XENIX; so does Intel's FORTRAN-286. The third compiler for XENIX-286 is version 3.31 from Microsoft: at first glance it looks like version 3.31 for MS/DOS but on using it, it seems to be less sophisticated than the MS/DOS version. Microway's mf77, now at version 1.4 as for its MS/DOS cousin, runs only under System V release 3.x on i386-based machines but does so rather well, supporting of course Weitek code. LPI claims that it also runs on MS/DOS: however the MS/DOS version is not available in France. It has been already stated above that SVS runs on MS/DOS.

The Mac World

Practically nobody is working in FORTRAN within Mac/OS; nevertheless there are many compilers that are good and solid but not very performant. Version 2.20 from Microsoft is as good as the MS/DOS version 3.x except for co-processor support: there is none to support!. Absoft's product is another plain vanilla one. DCM FORTRAN from Dataproducts is the best choice, having most of the features usually encountered elsewhere: library manager, symbolic debugger, unlimited program and data size, interface to C and Assembly... even support for numeric co-processor (e.g., M68881) if present; the genesis of this product deserves an explanation: it was initially developed on specification and for private use by a big industrial company who has much invested in Mac hardware, and then subsequently marketed, apparently with some success. Recently Language Systems Corp. announced FORTRAN in MPW, where the most interesting part is MPW, the acronym for MacIntosh Programmer's Workshop; this product is not yet available in France.

A/UX is not even two years old yet that we are already being supplied with good and solid compilers. Version 1.8 of gf77 is as good as the equivalent products for other UNIX boxes. The Optimising Compiler from Unisoft should be even better. Nkr's version 2.0 is presently the best choice. If Absoft's FORTRAN, not yet released, is simply the UNIX port of the Mac OS version, then we shall have yet another more *odourless and colourless* plain vanilla FORTRAN.... This situation seems to prove that (1) A/UX and the FORTRAN compilers give the impression that the breakthrough by Mac into scientific and engineering processing is possible and nearing, and (2) portability, standardisation, and compatibility in UNIX are not empty concepts.

The DEC/Vax World

Most certainly a notable share of the popularity and credibility of Vax/VMS amongst university and scientific users is due to the quality of the VMS FORTRAN compiler. Some ardent partisans even use it for reference results in numerical accuracy comparison. There is no reason for this entry, about the Vax/VMS compiler, in this paper, if not as an illustration for a rule and its exception: the author knows for sure that there is *no*[9] compatible compiler here, and is willing to bet *une bouteille de vin mousseux* on this. As a matter of fact, there is another reason to mention this Vax/VMS compiler here: being one of the first ANSI 77 compliant (1978-1979), it does attract users and being widely accepted, it can allow *Vax/VMS extensions*. These extensions are so widely used that, now, for most f77, they become an unavoidable must: in some sense, all these f77 are Vax/VMS FORTRAN compatible compilers!

The IBM Mainframe World

FORTRAN 4 H developed for the IBM System/360 between 1963 and 1967 and then *extended* in 1973 to include new features available in the System/370 account for most of the scientific processing on IBM (and PCM) mainframes throughout the sixties and seventies. There even exists an Extended Optimisation Enhancement version of Fort 4.H released in 1978 under pressure from some big and prestigious users who not only had the need for ever increasing performance and/or functionality, but also care to voice firmly for it. VSFORTRAN, ANSI 77 compliant, which appeared at the beginning of the eighties, did not meet the promises of the supplier and the expectations of

9. other than the Waterloo product mentioned earlier.

the users,[10] at least not until version 2 well into the mid-eighties. The mature VSFORTRAN now de facto became the FORTRAN element of IBM's SAA; it is one of the two compilers available for the PC/RT and the only one for the PS/2 if you stay with AIX. Most curiously, this PS/2 AIX vs compiler, common to C and Pascal, strongly suggests that this is a new offspring and hence a question: how or how much 'vs' is this common compiler with respect to the mainframe homonym?. However, with the perceptible (*but is this a mirage?*) thrust from IBM into super-computing, especially the parallel processing domain, things may change quickly, yet it remains that, up to now, there is no excitement to use FORTRAN compilers from IBM.

In conjunction with the pseudo-compatible phenomena engineered by Fujitsu (and to a lesser extent by Hitachi) there is an ANSI 77 FORTRAN fully compatible for the MVS environment, whose most impressive particularity is the Fortune (to help interactively fine- and hand-tuning the code) and Vectune (to help squeeze the last vectorisable part of the algorithm) utility tools.

For completeness, it should be remembered that the Waterloo family does have a MVS member too.

The Value-Adders

Using supercomputers such as Cray is much like driving a sports car: every plain street-walker can settle for a casual trip, but to squeeze out the last percentage of performance required tons of experience or lots of help. The notorious Pacific Sierra Research Corp. with its Forge and Vast has set out to help people with this. In fact, its Vast, a vectorising precompiler is now bundled with the FORTRAN compiler on many machines: Sperry ISP, Cyber 205, ETA, Nas VPF, Convex, Stellar... and soon on the i860 ('a Cray on a chip'). Forge is meant to interactively help users in restructuring and optimising code for Cray computers; it is 'UNIX-based' (a must nowadays!) and is available on many 68K based machines, notably Apollo and Sun 3. Lastly, PSR also offers a kind of Virtual Array Memory Manager "to provide very large arrays in environment of limited resources" (!).[11]

Still many more companies strive to write new compilers and/or translators or produce new tools to improve existing investments: Digital Research, Tartan Laboratories... In the chapter of programming utilities, almost every CS Dept. of any university and many DP Dept. of most big FORTRAN industrial shops have developed some kind of FORTRAN syntax checker, beautifier, profiler or cross-reference analyser... for example, Forcheck, "a f77 verifier and programming aid" or Spag, the "spaghetti unscrambler" (also a pretty-printer) from Salford University, or Sabah, a listing analyser... In order to improve an existing compiler, HCR sells a kind of *portable* global optimiser that can be inserted into any compiler chain that does have an intermediate language generation. The company claims that such a product is used in many compilers on many machines: IBM PC/RT, CDC Cyber 180, DEC/Vax, Gould PN and NPL... Rapitech and Intrinsic sell translators from FORTRAN to C...

The Idealist

Kenneth Wilson, the Nobel-prize physicist, once declared imprudently that "FORTRAN should not be taught any more".[12] Fortunately not everyone has heard about his assertion, (and myself a simple-minded partisan of FORTRAN, am wondering if I should or should not be punished for reviving it?) and it might be that we will have great joy in welcoming soon a *gnu_f77*. Considering the past, the spirit, the accomplishment of the Free Software Foundation and the performance of *gnu_cc*, much can be expected and why not a nomination for FSF as the man of the year if such gnu_f77 exists? (what year? uh!).

10. how many of you were authorised and able to compile under TSO?

11. by the way, what did Seymour Cray say about real versus virtual memory?

12. he has reiterated recently in claiming that "A scientist who writes in FORTRAN is not a thinking scientist". What do these scientists think about this?

Further readings

Benchmarking FORTRAN Compilers, Avram Tetewsky, Byte, February 1984.
Serious FORTRAN, Chris Wolf, PC Magazine, Vol 4 # 26, December 1985.
FORTRAN Perspectives, John Voglewede, PC Tech Journal, June 1987.
Three FORTRAN 77 Compilers, David Burleigh, Byte, November 1987.
MS/DOS FORTRAN Roundup, Scott Robert Ladd, Computer Language, Vol 5 # 11, November 1988.
FORTRAN meets OS/2, John Voglewede, PC Tech Journal, December 1988.
Off the shelf, (a review by) Richard Morin, UNIX Review, Vol 6 # 8.
Tested mettle, (a review by) David Wilson, UNIX Review, Vol 7 # 5.
Ftn77/386, Mike Gunn and Arul Britto, .EXE Magazine, Vol 3 # 11, May 1989.

Some addresses in France

Greenhills/Oasys products by SCT Electronique, Antony.
Microway products by Elsa Software, Velizy or Softway, Paris.
LPI products by Top Log (Metrologie), Suresnes.
Austec Ryan Mac Farland products by AENI, Courbevoie.
Lahey products by Aquitaine Systemes, Paris La Defense.
Prospero products by Imaco, Paris.

and in the UK

Prospero, London.
Polyhedron Software, Standlake Witney or University of Salford, Manchester.
Austec, London.

C Traps and Pitfalls—Book Review

C Traps and Pitfalls
Andrew Koenig
Addison-Wesley ISBN 0-201-17928-8 (UK) Price £13.45, Paperback, 147 pp.
Reviewed by Lindsay F. Marshall, Computing Laboratory, University of Newcastle upon Tyne, UK.
email: Lindsay.Marshall@newcastle.ac.uk

Do not read this review. Go *now* and buy the book. You will not regret it and it isn't expensive. Everyone who programs in C will learn something new from reading it and will experience great relief when they see that other people are just as stupid and make the same mistakes as they do.

The book is really well written and a joy to read. There are some wonderful examples (I particularly like the ones for finding out if your C compiler nests comments or not) distilled from Bell Labs' years of experience of C programming. The only disadvantage I could see is that some of the horrors described could give you nightmares and put Stephen King out of business. Why are you still reading this? Go! Buy it!

(P.S. I didn't even get a review copy for this—I just wanted to tell people what a great book it is.)

EUUG Executive Report

Helen M W Gibbons
euug@inset

European UNIX system Users Group (EUUG)
Owles Hall
Buntingford
England

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

The EUUG Executive Committee met again on 3rd July, 1989 in London, and spent some time in discussing the overall policies and finances of the Group.

The financial situation is at present stable, with a healthy but not excessive reserve, and the possibility of setting up ECU accounts and European Bank Accounts to make payments easier throughout Europe is being looked at.

Now that membership is approaching 4,000 members affiliated through 16 national groups, investigations are in hand to see if it will be possible to create a more meaningful corporate image to the outside world. The Governing Board agreed, at its meeting following the Dublin Workshop in May, that the logos of the National Groups should be aligned to the EUUG. A graphics company has been briefed to come up with an exciting new logo design and this will be discussed in more detail at the next Governing Board meeting which will be held on 23rd September, 1989 in Vienna.

To get a view of what this corporate identity should be and also to seriously assess what are the needs of members and how they can best be served, a survey is being conducted through a professional survey company. The survey is being sent to a 20% sample of members and the results should greatly help in directing the management of the Group.

Slide presentations on the Group are already in existence and are available for National Groups to use at their meetings from the Secretariat at Owles Hall.

Three new members have joined the Executive Committee in order to share the workload. These are Frances Brazier (Dutch Group), Norman Hull (Irish Group) and Johan Helsinguis (Finnish Group). Norman Hull will take on Internal Relations with the National Groups and Johan Helsinguis will cover External Groups.

Conferences continue to be a major part of the EUUG calendar and the next Conference in Vienna is already organised for 18th-22nd September, 1989. Bookings booklets containing the full provisional programme have already been sent out to all members. The Keynote Speaker will be Professor Ahmed Elmagarmid from

Purdue University in the States. Professor Elmagarmid is a specialist in the field of concurrency control for multi database systems. Full details for anyone who has not received their personal copy of the Bookings Booklet are available from the Secretariat at Owles Hall.

For the first time there will be a mail service at the Vienna Conference.

Future conferences are being organised in:

Munich, West Germany 23 - 27 April, 1990

Nice, France 22 - 26 October, 1990

Tromso, Norway 20 - 24 May, 1991

Budapest, Hungary 16 - 20 September, 1991

Mr. Dominic Dunlop has now taken up his role as Standards Officer and attended the ISO JTC1 SC15 WG 22 (POSIX) meeting in Ottawa. He will attend the next meeting in Brussels in October.

Co-operation with EARN is progressing.

Congratulations to Sunil Das on his recent promotion to the post of Senior Lecturer at the City University, London.

Dublin Workshop 5th.-7th. May

Norman Hull
Cute Hoor <norman@q2rs.uucp>

Q2RS
Court Place
Carlow
Ireland

Norman Hull is about to celebrate (in Vienna) 21 years in computing. During this time, for most of which he has been self-employed, he has worked on all major makes of mainframe (VERY alliterative), a few minis and many micros. His speciality is systems design, particularly in Office Automation related applications. It was this which brought him, belatedly, into the UNIX field a little under 4 years ago. Since then he has refused to work on anything less. Having joined the IUUG when working on his first Unix contract, he is currently secretary of the Irish group and a co-opted member of the European executive - which still leaves him a couple of days a week for his Software company Q2RS.

I am not really the best person to write about this as I was too involved in the local end of things, however I shall try to be reasonably objective.

First of all, as this was an EUUG event, and as such there were representatives from most European countries very much in keeping with the spirit of 1992. As is only to be expected in Ireland, we had a last-minute crisis. The Minister for Science and Technology, who was to welcome people and officially open the event had to stand in for a senior minister who had fallen ill, and all other ministers were occupied. However, in typical Irish fashion, the local member of the European Parliament re-arranged his schedule (with less than 24 hours notice) and performed the honours.

After the opening, and dinner, we had an introductory session. This was chaired by Michel Gien, who had put tremendous effort into arranging the subjects for discussion despite receiving very little input from the National Groups. This laid down the ground rules and prepared people for the main events on the Saturday. Having been at the Paris Workshop, I knew that the Saturday would be hard work and would quite likely not finish until quite late so, in common with a few others I had a very early night - about 3. am.

The first session on Saturday covered EUUG objectives and overall strategy, following which we broke into a number of working groups. These groups, some of which ran in parallel, covered Conferences, Public Relations, Publications and Software Distribution, Membership Structure, EUnet, and EUUG Administration.

Each working group produced a report of its deliberations, and these were all typed up and printed by Mairead O'Gorman who worked selflessly to get copies available in the shortest possible time, and was much nicer looking than most of the delegates. This meant that copies of each paper were available to each delegate when we resumed in plenary session to discuss the findings of each sub-group. Saturday's sessions extended until late evening, as expected, and were followed by an informal meeting of some of the delegates from smaller countries at which a number of issues were discussed. It was agreed that Friedrich Kofler from Austria would engage in various discussions on our behalf, with certain larger groups as we had a strong consensus on our European identity. This meeting broke up after 1. am.

The Sunday morning plenary session had the task of filtering the sub-group papers and putting forward voting points for the brief Governing Board meeting which followed.

The Governing Board meeting, which was over in record time, dealt with and agreed all of the voting points put to it by the body of the Workshop. This left us just a little late for lunch.

One unfortunate omission from the weekend was the Italian team, but I suppose that they can be forgiven as Joy Marino experienced the true meaning of his name when he became the father of Nicolo' on Thursday 4th. May. This, of course, left no time for him to book his ticket to Dublin ;-)

Summary of Discussions

In the summary below I have high-lighted the main points of discussion and the main conclusions that were reached. These have been grouped by the session/working-group where the items were discussed.

Remember this was a strategy meeting, we were in Dublin to decide on major goals and overall policy; the burden of implementation of what we decided is the job of the Executive Committee and the National groups.

EUUG OBJECTIVES

The first session was to decide upon the overall EUUG objectives. What is the EUUG about, what should the EUUG be doing ? All other discussions and decisions depended on this.

After much debate the following were agreed upon:

- UNIX is of prime interest
  UNIX is the operating system/environment which is of prime interest to the EUUG. Our definition of Unix should not be too precise, but should include such things as Unix being a culture, a platform for developing portable applications, and a vendor hardware independent system.
- EXCHANGES of technical information
  The EUUG should act as a co-ordinator in the exchange of technical information. In doing this it should enhance the services provided by the National groups. This technical information will mainly be concerned with the technology surrounding Unix rather than with its use.
- Exchanges of commercial information
  The EUUG should also act as a forum for the exchange of services that the National groups provide. This could be implemented through commercial exhibitions. The EUUG should, when presenting technical or commercial information, present only facts, not opinions; we didn't want to be seen to have a particular commercial bias.
- Increased european awareness
  The EUUG should stimulate European awareness both inside and outside of the EUUG. This should be done in a very positive way. Various opinions/views/concerns of the National groups should be expressed by the EUUG.

  There was a strong feeling that the national groups should work together, and be seen to be working together.
- EUUG as a co-ordinator
  The EUUG should act as means of co-ordination between National groups and National groups' activities. In this role of co-ordinator, when conflicts arise, these conflicts should be brought to the attention of the Governing Board, where votes on the actions resulting from the conflicts can be taken.

  It was recommended that the Governing board decisions once ratified be binding on National groups. This is to ensure that the European infrastructure which we have built up, is maintained and grows.

EUNET

This group discussed EUnet and the EUUG's relationship to it. We found that due to a combination of historical and PTT licensing reasons EUnet is a EUUG service for EUUG members, and for EUUG members only. However national groups may disguise EUUG membership costs to non-members who wish to use EUnet services; they must be informed that they have become members of the EUUG.

It was considered that archive facilities are highly desirable. These services must be reliable and consistent and it was agreed that financial support be provided.

As the usefulness of EUnet depends on being able to reach those that you want a decision was made to encourage more EUUG members to subscribe

to EUnet.

We felt strongly that EUnet was not an end in itself and so if it was found that someone else can provide a better and cheaper network, EUNET should be wound up.

To help public awareness of EUUG activities, an EUUG newsgroup is to be created. This would be used to advertise EUUG and National group activities.

EXECUTIVE DIRECTOR

There are many things that the EUUG could or should be doing. One of the problems is the mainly voluntary nature of the members of the Executive and Governing boards; this means that they cannot often devote the time that some issues really need and progress into new fields can take a back seat to dealing with their current EUUG work – and earning a living. Owles Hall is a secretariat and runs existing activities, it's job is not that of starting new ideas.

It was noted that both Usenix and the AFUU had benefited greatly by taking on paid staff as initiators. Both organisations agreed that, after some initial trepidation (eg about the financial risks), the decision had been well justified.

It was agreed that the EUUG should look into the possibility of hiring an "Executive Director" whose role would be to:

- Relieve the Executive committee from day-to-day management of the EUUG.
- Initiate and operate actions in the following areas: Co-Ordination (national groups with each other and with groups outside Europe); improving the external visibility of the EUUG and of the National groups; look at new areas in which EUUG might be involved; and in managing EUUG projects.

CONFERENCES

Neil Todd (the Conference Executive) had prepared substantial reading material for this session.

The first observation was that the Conference Executive noticed that there is a need to plan 2 to 3 years ahead which is considerably more than previously. Because of the financial stability, advance risky bookings can now be afforded.

The major observation was that the EUUG conferences should benefit members. They are organised under responsibility of the EUUG - with local support.

First point of discussion was the target audience of the conferences. It is forseen that, certainly in the future, there will be a split in the areas of interest which then will be covered in parallel sessions. For instance the AFUU recognises research papers, user papers and product papers to which we will probably add tutorials. The AFUU even has three different Program Committees to select papers.

A suggestion was made to host a specific session for Manufacturer Sales talks, perhaps associated with the exhibition.

The second major point of discussion was the conference language. It was eventually decided that the EUUG member survey should ask if the official conference language should be English, or that there be two official conference languages: English and the local language. Members views on simultaneous translation should also be sought.

A next point of discussion was tutorials. There is a strong tendency to hold tutorials before the conference. It was generally felt that the tutorials are "good value for money".

We were pleased that quality of the proceedings has gone up. They are repeatedly produced by the same team of people. Printing is done in the local country. Tutorial help to prospective authors may be investigated.

It was felt Calls for Papers for conferences did not receive a wide enough circulation, and that this be improved to maintain the quality of papers being received.

There was no firm conclusion on the point, but the feeling was that we could try one Conference per year with additional technical work shop(s) throughout the year. Warning was issued that one conference per year does not necessarily imply a doubling of the figures for attendances.

There was a suggestion to contact other User groups such as the Sun User group, Decus and the HP User group.

There is a general feeling that conferences should be cheaper but it is realised that it is difficult to achieve.

PUBLICATIONS and SOFTWARE DISTRIBUTION

It was felt that publications are needed but not exclusively on paper, some should be available in electronic form - email and software tapes.

No value was seen in the EUUG being involved with publications in areas that are already being covered sufficiently elsewhere. Although the EUUG wasn't in the business of acting as a book shop, there were a few instances where bulk purchase may be of benefit to members; however the market for items such as particular manuals had to assertained before getting a lot of them.

One perennial problem is how to find information; this is an area where a catalogues of catalogues would be useful. The EUUG should produce one.

Several National groups had begun to duplicate EUUG conference tapes and distribute them locally, it was decided that this should be allowed and encouraged. This would result a smaller number of tapes being sold and so a rise in price. Therefor the production costs should be funded centrally by the EUUG.

It should be noted, however, that the tapes remain with copyrighted by the EUUG and that there may be further software licensing restrictions.

Similarly tapes produced by National groups should be made available at cost price to other EUUG members. This is subject to copyright and the price may include production and research costs.

National groups should make EUUG aware of the existence of such tapes, and there should be a central service from EUUG to process the requests of items which are sourced from separate National groups.

The EUUG Newsletter should be published in English. Original language copies of individual articles are to be available through the EUUG (and from the National group); there should be a notice in the Newsletter about this when appropriate.

PUBLIC RELATIONS

It was felt that at present the relationship of one national group to another is not obvious in the public eye, if this could be improved all should benefit. Various means were discussed and it was decided that initially all UUG logos and other "visibility" material should be aligned to exhibit a "common corporate" identity.

Further more there should be a PR budget for press conferences and all National groups should assign someone to deal with public relations.

COORDINATION

It was felt that there was not enough coordination between the National groups. An example of an area of improvement is that presently National groups occasionally hold conferences at the same time; members might be interested in going to both of them – especially if they are geographically close and speak the same language.

To help in this the EUUG should organise a centralised register of events. This would be sent monthly to Governing Board. A five year plan should be drawn up.

National groups should have more than one contact point as a single contact is some times not available.

National group fax numbers should be on the back of the Newsletter.

STANDARDS

It was felt that the EUUG should have an active participation in standards organisations, the purpose is to present a European point of view, and to ensure that our special needs are not left out.

The organisations to be represented on are those that are: non political, further the aims of the EUUG, don't promote products.

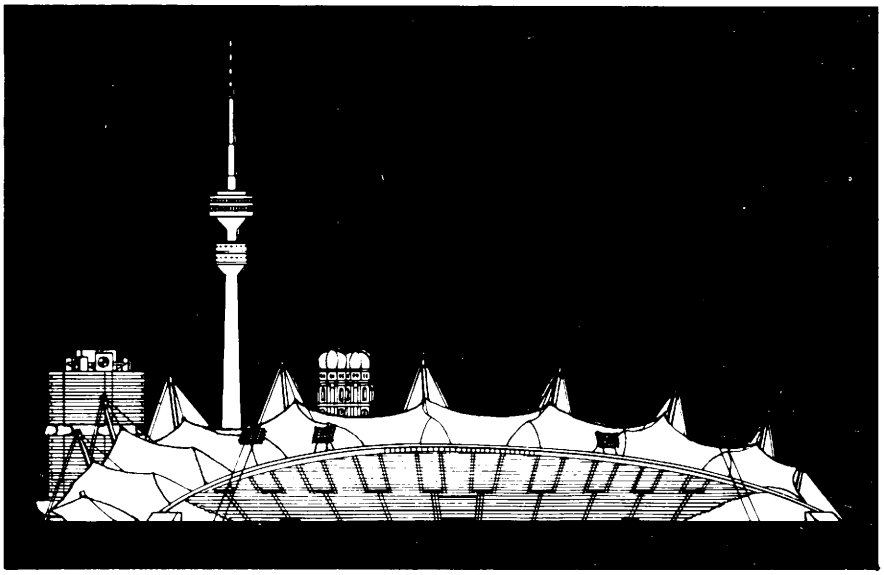
The EUUG should seek passive observation of other standards organisations.

EUUG

European UNIX® systems User Group

PRELIMINARY ANNOUNCEMENT and CALL FOR PAPERS

EUUG Spring '90 Conference and Exhibition



Preliminary Announcement

The GUUG will host the Spring '90 European UNIX® systems User Group Technical Conference at the Sheraton Hotel in Munich, West Germany, Europe.

Technical Tutorials on UNIX and closely related subjects will be held on Monday 23rd and Tuesday 24th April, followed by the three day **Conference** with commercial **Exhibition** finishing on Friday, 27th April.

A pre-conference registration pack containing detailed information will be issued in January 1990.

Call for Papers

The EUUG invites papers from those wishing to present their work. Full papers or extended abstracts must be submitted. All submitted papers will be referred to be judged with respect to their quality, originality and relevance.

Suggested subject areas include, but are not limited to:

Standards for UNIX Systems
Internationalisation
Object Oriented Development Tools
Object Oriented Graphical Toolkits
Object Oriented Languages
Program Generators for Commercial Applications
Network Administration
Security Issues and Authentication Techniques
Document Context Architectures

Important Dates

Submission deadline	15th November, 1989
Acceptance notification	15th December, 1989
Final paper	10th February, 1990
Closing date for Student Grant Application	1st March, 1990

Method of Submission

Full papers or extended abstracts must be submitted by post to the EUUG Secretariat and, if possible, in electronic form to euug-munic@cwi.nl. All submissions will be acknowledged by return of post.

Tutorial Solicitation

Tutorials are an important part of the EUUG's biannual events providing detailed coverage of a number of topics. Past tutorials have been taught be leading experts.

Those interested in offering a tutorial should contact the EUUG Tutorial Executive as soon as possible.

Additional Information

We will be pleased to provide advice to potential speakers. We can be contacted at the addresses below.

If you wish to receive a personal copy of any further information about this, and future EUUG events, please write, or send electronic mail, to the Secretariat.

Useful Addresses

Secretariat
EUUG
Owles Hall
Owles Lane
Buntingford
Herts
SG9 9PL
UK

Phone: (+44) 763 73039
Fax: (+44) 763 73255
Email: euug@inset.uucp

Conference Executive
Dr. Ernst Janich
Nixdorf Computer AG
Söflinger Str. 70
D-7900 Ulm
W. Germany

(+49) 731 1526 464
(+49) 731 1526 105
janich@nixulm.uucp

Programme Chair
Hans W. Strack-Zimmermann
iXOS Software GmbH
Technopark
Bretonischer Ring 12
D-8011 Grasbrunn
W. Germany

(+49) 89 46 10 05 0
(+49) 89 46 10 05 99
unido! ixos! hsz

Local Organisation
GUUG
Vereinigung Deutscher
UNIX® -Benutzer e.V.
Dr. Anton Gerold
GUUG-Vorstand
Elsenheimerstr. 43
D-8000 München 21
W. Germany
Phone: (+49) 89 570 7697
Fax: (+49) 89 570 7607
Email: gerold@lan.informatik.tu-muenchen.dbp.de

Tutorial Executive
Neil Todd
GID Ltd
1 Captains Gorse
Upper Basildon
Reading
Berkshire RG8 8SZ
UK

(+44) 491 671 964

neil@gid.co.uk

Programme Committee
Daniel Karrenberg
CWI, CST
POB 4079
NL-1009 AP Amsterdam
The Netherlands

(+31) 20 5924112
(+31) 20 5924199
dfk@cwi.nl

Marc Nyssen
Department of Medical Informatics
VUB
Laarbeeklaan 103
B-1090 Brussels
Belgium

For further details, contact:
The Secretariat
European Unix® systems User Group
Owles Hall, Buntingford, Herts SG9 9PL, UK
Phone: (+44) 763 73039 Fax: (+44) 763 73255 Network Address: euug@ inset. uucp

ORDER FORM

CONFERENCE & WORKSHOP PROCEEDINGS

Qty	Proceeding		Add'l Unit Price	Total	Foreign Postage	Total
___	Baltimore Conference	June '89	$20	$_____	$15	$_____
___	UNIX Transaction Processing Workshop	May '89	12	$_____	8	$_____
___	Software Management Workshop	Apr. '89	20	$_____	15	$_____
___	San Diego Conference	Feb. '89	30	$_____	20	$_____
___	C++ Conference	Oct. '88	30	$_____	20	$_____
___	UNIX and Supercomputers Workshop	Sept. '88	20	$_____	15	$_____
___	C++ Workshop	Nov. '87	30	$_____	20	$_____
___	Graphics Workshop IV	Oct. '87	10	$_____	15	$_____
___	Washington DC Conference	Jan. '87	10	$_____	20	$_____
___	Graphics Workshop III	Dec. '86	10	$_____	15	$_____

Total price of Proceedings _____
Calif. residents only add applicable sales tax _____
Total foreign postage _____
Total enclosed $_____

* Discounts are available for bulk orders. Please inquire.

PAYMENT OPTIONS

☐ Check enclosed payable to USENIX Association. ☐ Purchase order enclosed.
☐ Please charge my: ☐ Visa ☐ MasterCard

Account # _____ Exp.Date _____

Signature _____

Overseas? Please make your payment in U.S. currency by one of the following:
* Charge (Visa, MasterCard, or foreign equivalent)
* International postal money order
* Check - issued by a local branch of a U.S. Bank

5/89

Shipping Information

Orders to U.S. and Canada are shipped via printed matter. Please allow 2 weeks for delivery. Foreign orders are shipped via air printed matter. Please allow 10-14 days for delivery. Shipment by UPS, first class, or airmail is available upon request.

Ship to: _____

_____

_____

_____

Please mail this order form to:

USENIX Association
2560 Ninth Street
Suite 215
Berkeley, CA 94710

Portuguese UNIX systems Users Group Report

PUUG Executive
puug@inesc.pt

Portuguese UNIX systems Users Group
Av. 24 de Julho,
1200 LISBOA
PORTUGAL

How it Happened

The Portuguese UNIX systems Users Group was founded recently (last June).

The idea of creating a local group of UNIX fans in Portugal had been around 4 to 5 years but it had not been possible to gather the critical mass to get things working. Taking advantage that the EUUG 1988 Autumn Conference was held in Estoril (around 20 Km from Lisbon, as many of you still remember) we finally organised a start-up meeting. After this start-up meeting we began the preparation of the necessary bureaucracy and in the 2nd June we had our first General Assembly.

Who is doing What

The General Assembly has elected the following people to run the PUUG:

Executive Committee

Jose Legatheaux Martins	Chairman
Manuel Simoes	UNIX Convention
Paulo Amaral	Treasurer
Pedro Veiga	Network
Paulo Vilela	Secretary

Fiscal Council

Antonio Ferrari
Antonio Mendes dos Santos
Jose Armando Silva

The list of activities for the first mandate is to try to make the PUUG widely known and useful to the UNIX Community. As such two major topics will be dealt during the first year:

- Organisation of a Portuguese UNIX convention, to be held early next year.
- Promote the expansion of the Portuguese branch of EUnet. The network has already been running for 3 years, but with a limited number of nodes, although these are very active.

Membership

After the General Assembly of the 2nd of June we already have the following membership:

- Institutional members: 15
- Individual members: 14

We expect that the membership will increase during the Institution's first year, especially after the organisation of the first PUUG convention.

News from the Netherlands

Frances Brazier
frances@psy.vu.nl

Department of Cognitive Psychology
Vrije Universiteit
Amsterdam
The Netherlands

Frances has a master's degree in Mathematics and Computer Science, and has been doing research at the Department of Cognitive Psychology for the past 7 years. Human-machine interfaces and information retrieval are her major fields of interest.

Conferences

As was mentioned in the last Newsletter the NLUUG's last conference, held on May 9th, was on human-machine interfaces. A short description of each of the talks should provide an impression of the programme's contents. As the quality of the slides was above expectation conference notes will be published of copies of the slides presented.

A. Jameson and W. Claassen (Nijmegen Institute for Cognitive Research and Information Technology, KU Nijmegen), discussed the promises and prospects of natural language communication with computers, using UNIX consultants as an example of the nonlinguistic issues involved in the design of such systems and the resolution of ambiguities as an example of linguistic issues. Three UNIX consultants which use a form of natural language communication, were compared on the basis of the knowledge on which inferences made during interaction, are based. The major problem identified concerned the fact that such systems have much less and simpler knowledge than do human experts. The promising aspect concerned the fact that such systems use the same sort of knowledge that humans use in handling the same types of situation.

René Collard, (Océ Nederland BV, Research and Development), described the process of developing a graphical user interface for a publishing and printing system. The four phases of product development he distinguished: analysis, design, realisation and testing, were described. The phase of design, the phase in which the user interface was developed, was described in slightly more detail. The principles behind the design of the interface were those of simplicity, consistency and flexibility. The three layered approach, in which concept, dialogue and presentation are considered separately, was illustrated for menu design. A number of design decisions were highlighted, the resulting screens were shown.

Scott Ritchie's (SUN Microsystems Inc, USA) session on the OPEN LOOK graphical user interface, provided an overview of the goals, implementation, and implications of X/Open. The basic design goals (1) balance of simplicity, consistency and efficiency, (2) good visual design, (3) interoperability and (4) device independence,

were described, illustrated and clarified when necessary. The legal status, availability and application were also addressed.

Kieron Drake, (Department of Computer Science, Queen Mary College, London) held a revised version of the paper presented at the EUUG conference in Lissabon: *NeWS and X, Beauty and the Beast?*, in which the strengths and weaknesses of the two systems are compared.

S. Doaitse Swierstra, (Department of Computer Science, Rijks Universiteit Utrecht) presented the design of a system in which a programming environment based on the incremental evaluation of attribute grammars is combined with a graphical interface. The use of higher order attribute grammars, the interpretation of user interactions, the interface with the graphical subsystem and incremental evaluation were topics included in this presentation.

Charles van der Mast, (Information Systems, Technical University of Delft) discussed an object oriented approach to user-interface design of information systems. The types of expertise required for the design of information systems were identified and assigned a role within the design process. The complexity of the design process of object oriented systems was stressed.

The rôles of the three components distinguished: static component, dynamic component and the visual component, were discussed. The necessity to consider help and error messages was mentioned.

A layered model of the components involved in the user interface formed the basis for Bouwe van der Eems' (Apollo Computers bv), presentation on a standard user interface management system. The role of the X-Window System in the model, namely for the data stream encoding layer, base window system interface and for the toolkit intrinsic layer, was described in detail. The toolkit layer and the presentation layer were delegated to OSF/Motif, the Dialog Layer to Open Dialogue. The functions of both systems in this model were described to a fair level of detail.

Our next conference

On November 10th the NLUUG will be holding its next conference, with an exhibition, on *UNIX & Connectivity*. Anyone interested in presenting a paper is invited to contact the programme chair (Emile van Dantzig, emile@ace.nl), before September 8th. For further information on the conference itself please contact the NLUUG buro (address on the cover of this Newsletter).

Call for Papers

UNIX & Connectivity

Autumn Meeting of the NLUUG:

9 November 1989

EDE (De Reehorst)

The Netherlands

Papers

The program committee for the biennial NLUUG meeting invites you to submit a paper for the upcoming meeting, which will have as key subject: "UNIX and Connectivity".

The aim of this meeting is twofold:

- informing participants of the latest developments in the areas of networks and network applications,
- offering instruction, through the tutorials, in the areas of network use and UNIX-related matters.

The papers to be presented will be technically and/or research oriented. The program committee suggest the following subjects:

A. Media & protocols (UUCP, TCP/IP, DECNET, SNA, FDDI, Token Ring, Ethernet, Sockets, XTI, USENET/NSFnet/Internet/X.25),

B. ISO/OSI/Standards (ODA/ODIF, X400/X500, ISODE),

C. Management (security, configuration, maintenance, performance),

D. Distributed environments (file transfer, electronic mail, remote login, NFS, RFS, X-11, distributed operating systems).

Tutorial sessions

The program committee suggest the following tutorial subjects:

A. Network configuration and performance,

B. NFS configuration,

C. Security and management,

D. UNIX for new users,

E. UUCP configuration,

F. Connectivity with UNIX (PC, Mac, OS/2 etc),

G. Standardisation.

Correspondence

Please send abstracts (maximum 1200 words) BEFORE 8 SEPTEMBER 1989, to one of the members of the program committee:

Emile van Dantzig (chair) <emile@ace.nl>, or
Henk Hesselink <henk@ace.nl>
ACE Associated Computer Experts bv
Van Eeghenstraat 100
1071 GL Amsterdam
+31 20 6646416

Marten van Gelderen <marten@xirion>
Xirion bv
World Trade Center
Strawinskylaan 1135
1077 XX Amsterdam
+31 20 6649411

Willem de Vries <wdv@sunnl>
SUN Microsystems Nederland bv
Postbus 1270
3800 BG Amersfoort
+31 33 501234

Jos Alsters <josal@wn2.sci.kun.nl>
Katholieke Universiteit Nijmegen
Afdeling CCZ
Toernooiveld 1
6525 ED Nijmegen
+31 80 613174

Organisation

Times

The meeting will start at 9.30 am. and finish at 17.00 pm., on Thursday 9 November 1989, in EDE in the "De Reehorst" conference centre.

Sessions

There will be two technical sessions of six presentations in parallel with four tutorials.

Duration

The presentations will last 45 minutes each, the tutorials will last 1 hour and 15 minutes each.

Proceedings

Proceedings will be made available.

Exhibition

An exhibition hall is available for hardware and software companies. The exhibition will be open from 9.00 am.

Questions

General organisatorial questions should be directed to:

Patricia Otter <patricia@xirion>
Xirion bv
World Trade Center
Strawinskylaan 1135
1077 XX Amsterdam
+31 20 6649411

Organisatorial questions regarding the exhibition should be directed to:

Hans Linschooten <hans@hpuamsa>
HP Nederland bv
Startbaan 16
1187 XR Amsterdam
+31 20 5476911

UKUUG Report

Mick Farmer
mick@cs.bbk.ac.uk

Birkbeck College
University of London
England

Mick is a lecturer at Birkbeck College (University of London) and the Secretary of the UKUUG. His interest is in all aspects of Distance Learning and he is the Senior Consultant (Software) for LIVE-NET, an interactive video network connecting London's colleges. He is also a member of the University's VLSI Consortium, mainly because the design tools draw such pretty pictures.

UKUUG Summer '89 Meeting

We held a successful Technical Meeting at the University of Strathclyde (Scotland) on 27-28th June. Twelve varied papers (synopsis in the last issue) were enthusiastically received by over 80 participants who had braved a strike on British Rail to be there.[1] Jim Crammond's tutorial on Sendmail was extremely popular. 46 people turned up and copies of his notes ran out.

Your Average Conference Attendee

At Strathclyde Technical Meeting the participants were asked to complete a questionnaire based on that given out at EUUG conferences. Just over 60% did so. We've analysed the returns using SPSS and, with this information, I've been able to reconstruct *Strathclyde Person* who, for reasons that will become obvious, I've nick-named Lindsay.

Lindsay is 31 years of age and lives in the United Kingdom, probably England.[2] Lindsay works primarily in an academic/research environment at the senior/professional level and has worked with computers for ten years, five of those involving UNIX. Naturally, Lindsay is a member of the UKUUG and possibly a member of the Sun UK User Group. Lindsay probably subscribes to and reads the EUUG Newsletter (through the UKUUG subscription) and also reads UNIX Review, UNIX World, and IX Magazine.

Lindsay travelled by scheduled flight to get to the meeting and, like most people, stayed in the Student Accommodation provided by the University of Strathclyde. The cost of attending this conference was paid by Lindsay's employer. As for previous meetings/conferences, Lindsay probably attended the UKUUG Winter '87 Meeting in London, the UKUUG Winter '88 Meeting in Canterbury, and possibly the EUUG Spring '88 Conference in London.[3] Lindsay heard about this meeting in the EUUG Newsletter and through the UKUUG mail shot, and probably saw the announcement on UKnet. Lindsay thinks that accommodation, coffee, lunches, and the conference dinner should be included in the conference fee but is less interested in the other social events or tapes. Lindsay feels that there should be a discount for UKUUG members but not for block booking or EUUG membership!

Lindsay felt that the technical sessions and the proceedings were of good quality and excellent importance.[4] Overall, Lindsay thought that the food was average, with the conference dinner

1. Copies of the proceedings may be purchased from the UKUUG Secretariat at £10 including post and packing.
2. I'm not offering many surprises!
3. Obviously not one for foreign travel.
4. What's that mean? - Ed.

good. In addition, Lindsay felt that there should be more product demonstrations, exhibitions, and book stalls with no commercial sessions, sales talks, or spouse programmes. For future meetings/conferences Lindsay would like to see more research speakers and UNIX super stars, but is not impressed by industry big shots or commercial speakers. Overall, Lindsay felt that the meeting was good value for money.

Finally, Lindsay thought that networking tutorials or those that are end-user oriented would be useful. Paying by Barclaycard/VISA would also be useful.

UKUUG Winter '89 Meeting

This meeting will be held at the University of Wales College of Cardiff on 11-13th December, 1989. Details are yet to be finalised but the main theme of the meeting will be networking. In a spirit of collaboration we expect the meeting to be sponsored jointly by the UKUUG, UKnet, JNT (Joint Network Team), and IUNC (Inter-University Networking Committee). The three days will include up to four sessions of talks, and we hope to run a couple of tutorials.

One of the sessions will be an introduction to UKnet and will be especially useful to new or potential UKnet members (both Commercial and Academic) as well as being of more general interest. Another session will be devoted to topics from the IUNC and JNT. We expect there to be a small exhibition.

Suggestions for tutorials and speakers can be made to the UKUUG (ukuug@ukc.ac.uk) or to the local organiser:

Robert Evans
Dept of Computing Mathematics
U.W.C. Cardiff
PO Box 916
Cardiff CF2 4YN
United Kingdom
Phone: +44 222 874000 x 5518
Fax: +44 222 371921
Email: robert@cm.cf.ac.uk

UKUUG Summer '90 Meeting

We plan to hold this meeting in London, England, on 11-13th July, 1990. Further details will follow when they are available.

UKUUG Winter '90 Meeting

We plan to hold this meeting in Cambridge, England. Further details will follow when they are available.

London UNIX User Group (LUUG)

Three talks have been scheduled for the last quarter of the year. These are

- September 28—Keith Brazington is talking about Amateur Packet Radio.
- October 26—Mick Farmer is talking about LIVE-NET, London University's interactive video network. This meeting will take place over LIVE-NET. November 30—Lori Grob is talking about Parallel Computing and UNIX.
- December 28—No meeting.

The LUUG lectures are organised by Andrew Findlay (Andrew.Findlay@brunel.ac.uk).

Glasgow Local UNIX Group (GLUG)

This group is currently resting over the summer. Suggestions for speakers and venues to Jim Reid, the local organiser (jim@cs.strath.ac.uk).

UNIX Security Workshop

A three hour VHS video of the proceedings is available for £50.00, including post & packing (excluding VAT in the UK) from the address below. You will be sent an invoice.

Birkbeck College Video Services
Birkbeck College
Malet Street
London WC1E 7HX
ENGLAND
Phone: (+44) 1 631 6351

UKUUG System Administration Meeting

Feb '90

The UKUUG intends to hold a workshop on UNIX Systems Admin during February 1990 in London.

Current topics of interest are:

- Smart shell and menu driven systems
- Administration in a Distributed Environment
- System Audit Procedures
- Software Integration and Test Procedures
- Data Protection Compliance
- Yellow Pages—friend or foe ?

Further suggestions for the programme are welcomed.

If you wish to offer a talk, please contact the Programme Chair as soon as possible—you will not be committed (at this stage).

Prog Chair

Neil Todd
GiD Ltd
1 Captains Gorse
Upper Basildon
READING
Berks
RG8 8SZ

neil@gid.co.uk

+44 491 671964

Secretariat

UKUUG
Owles Hall
Owles Lane
BUNTINGFORD
Herts

ukuug@ukc.ac.uk

+44 763 73039
+44 763 73255 (Fax)

AMIX report—Israel UNIX Users Group

Ariel J. Frank
ariel@bimacs.biu.ac.il

Deputy Chairperson
Dept. of Mathematics and Computer Science
Bar Ilan University
Ramat Gan, Israel

Dr Ariel Frank received his PhD in Computer Science from State University of New York at Stony Brook. He discovered computers during his studies at Bar-Ilan University and the Weizmann institute of Science in Israel.

Dr Frank is Deputy Chairperson of the Mathematics & Computer Science of Bar-Ilan University. His research interests are in Distributed Operating Systems and AI Environments. He has a keen interest in UNIX and serves for the last 3 years as the elected AMIX (Israeli UNIX user group) Chairperson.

Even though 1989 is not yet over, there has been no lack of activity by AMIX members in Israel. On 25-26 of January we had a series of tutorials and workshops on UNIX user interfaces. The most recent AMIX event was the 4th annual AMIX conference that took place on 14-15 of May in Ramat-Gan, Israel.

The UNIX interfaces tutorials were given by AMIX board members Dr Maurice Bach (author of the *The Design of the UNIX Operating System*) and UNIX guru Dr Joel Issacson. The tutorials covered UNIX system calls and library calls, terminal interfaces, and an introduction to the X system. The following day a workshop on Graphical User Interfaces (GUI) reviewed all the important UNIX GUIs (including DEC XUI, HP New Wave, Apollo Open Dialogue, and Sun/AT&T Open Look). The opening talk by Dr Issacson was titled *The window is open—beware not to fall*. About 70 participants attended both events.

The intensive first day of the 4th UNIX conference was comprised of two half-day tutorials. The first one, given by AMIX chairperson Dr Ariel Frank, was on the *Structure of Operating Systems and UNIX*. The second tutorial was given by UNIX hacker Eli Marmur on *Advanced Concurrent C Programming in a Commercial UNIX Environment*. About 50 participants attended both tutorials.

The second conference day turned out to be extremely interesting. The opening remarks were provided by Dr Micha Chanani, IPA (Israel Data Processing Association) chairperson. He wished for continued UNIX success but as a way of warning, he traced the fate of the successful Algol-60 language that turned into the failed Algol-68 system because of over ambition.

The opening talk, on the computing policy of the Israeli Government, was given by Naor Tal of the Finance Ministry. His main message was that UNIX is great and he is just waiting to be offered working and reliable UNIX products. It seems that the Israeli market is ready for UNIX if only all the concerned, especially system and software companies, organise and get more serious about the UNIX business.

Ron Lachman, president of the USA Lachman Associates (now part of Interactive), delivered an

excellent keynote talk. He traced and remarked on all the recent major events of the open systems and UNIX world. His theme was that *Yes, UNIX exists!* and he 'proved' it in many ways and cuts.

The afternoon was dedicated to a UNIX users session and a panel. The UNIX users session consisted of short presentations given by representatives of four high-tech Israeli companies who use UNIX heavily (Ready Systems, Optrotech, RobCad and Tovna). They described their successful use of various UNIX systems and tools for design and development of their advanced products. The concluding panel theme was *UNIX directions and developments in the 90s*. The participating experts elaborated on the UNIX success story in various parts of the world, the lessons to be learned, and on how to advance the state of UNIX in Israel, especially in the commercial market. The conference was attended by over 100 people.

In mid June we had a two-day commercial UNIX Show and Workshop organised by People & Computers (a computer weekely) with support of AMIX. The show was very successful with 20 major exhibitors. A rather nice crowd (for Israel) of about 300 people attended the show. The workshops included opening talks by Dr Ariel Frank of AMIX on *Trends in the UNIX Market* and by Richard Kolodynski of Europe SCO on *Developments in the UNIX/XENIX World*. A dozen additional talks where given by representatives of the companies exhibiting in the show.

USENIX Association News for EUUG Members

Donnalyn Frey
donnalyn@frey.com

Frey Communications

Ms Frey is the USENIX Association Press Liaison. She provides members of the press, USENIX Association members, and EUUG members with information on the activities of the USENIX Association.

1990 Winter USENIX Association Conference

The 1990 Winter USENIX Conference will be held in sunny Washington, DC on January 20 – 26, 1990. The first two days will be devoted to tutorials, with the next three days for technical sessions. Topics which may be presented include UNIX and Artificial intelligence, Ada and UNIX, Software Release Systems and Servers, Architectures and Compilers, File Systems and Servers, Distributed Systems and Servers, Distributed Systems and Services, User Interfaces, and Novel Applications. For further information on the conference, contact the USENIX conference office.

The 1990 Summer USENIX Association Conference

The 1990 Summer conference will be held on June 11-15, 1990 at the Marriott Hotel in Anaheim, California, home of Disneyland. The call for papers will be announced in November 1989.

Distributed Processing Workshop and Graphics Workshop

The USENIX Association will be holding a Distributed Processing Workshop in Fort Lauderdale, Florida October 5 – 6, 1989. The fifth Graphics Workshop will be held November 16 – 17, 1989 at the Doubletree Hotel in Monterey, California.

Further Information on Conferences and Workshops

If you need further information on registering for upcoming USENIX Association conferences or workshops, contact the USENIX conference office at Conference Office at 22672 Lambert Street, Suite 613, El Toro, CA 92630, USA. Email to {uunet,ucbvax}!usenix!judy or judy@usenix.org. The conference office can provide you with information on the annual Computer Graphics, Large Installation Systems Administration, UNIX Security, and UNIX and Supercomputers workshops. The office can also provide information on the 1990 C++ conference and the semi-annual technical conferences. A schedule of upcoming events is included in the calendar in this issue.

Preliminary Call for Participation

USENIX C++ '90

Tentatively in late-April 1990 in California

C++ continues to show explosive growth as the object oriented implementation language of choice for production level work. The nearly-annual C++ conference is a haven for those who use the language, those who develop the language, and those who are interested in the language. The conference enables them to take a look at where C++ has been, where is it now, and where future developments should take it.

The conference will consist of a day of tutorials and classes and two days of technical sessions. Papers are invited on all aspects of C++, from the development of compilers and preprocessors to case studies of projects which have used the language. Proposals for tutorials or classes on systems which make use of C++ or on the uses of C++ are also invited.

Paper abstracts and tutorial proposals are due **January 12, 1990**. Abstracts should be no more than two pages, and should describe the work in sufficient detail to allow the referees to judge the merit of the work. Tutorial proposals should be no more than four pages in length, and should describe the content, purpose, and intended audience. Abstracts and tutorial proposals should be submitted either electronically (preferred) or in hard copy; electronic submissions should be either plain text, *n/troff*, or PostScript. Notification of acceptance will be made by **February 2, 1990**; final papers in camera ready form must be received by **March 9, 1990**. Accepted papers which meet this deadline will be published in a conference proceedings.

Abstracts and proposals should be sent to:

Jim Waldo
Apollo Computer
330 Billerica Road
Chelmsform, MA 01826

waldo@apollo.com
decvax!apollo!waldo

Conference Announcement

Vienna, Austria
18th - 22nd September, 1989.

Professor Elmagarmid to speak at the EUUG Conference.

The European UNIX systems User Group is pleased to announce that Professor Ahmed Elmagarmid from Purdue University, USA has accepted an invitation to be the Keynote Speaker at the Conference. Professor Elmagarmid is a specialist in the field of concurrency control for multi-database systems.

As well as the keynote speech a full technical programme has been organised for 20th, 21st and 22nd September which will cover such subjects as Standards, Multiprocessor systems, User Interfaces Management, Graphical User Interfaces, Transaction Processing and Object-Oriented Systems.

The two days prior to the technical Conference, Monday 18th September and Tuesday 19th September will be devoted to Tutorials. These will cover UNIX Network Programming, RISC, System V IPC, MAKE, Beyond 4.3 BSD, The Andrew Toolkit and OSI.

The Lord Mayor of the city of Vienna has invited all delegates to a Conference reception at the City Hall on the opening day.

The Conference will be held in the Vienna Wirtschaftsuniversität and will be accompanied by a table top exhibition.

Registration Booklets giving full details are available from:

The EUUG Secretariat,
Owles Hall,
Buntingford,
Herts SG9 9PL,
England.

Tel: + 44 763 73039
Fax: + 44 763 73255.

The tutorial timetable and provisional programme appear on the following pages:

PROGRAMME OF EVENTS

TUTORIALS

Only EUUG National Group or Direct Members are permitted to attend Tutorials.

Each Tutorial lasts for one whole day and will start at 09.30 except M3 & M4 which are half-day only.

Monday 18th September

Registration from 08.00

Tutorial M1 – UNIX Network Programming

Tutor: Richard Stevens, Health Systems International

Little information exists on the actual development and programming of software to run in a network environment. In the UNIX world this means understanding sockets or TLI, and also the basic primitives provided for concurrent programming (fork, exec, signals, IPC).

Some knowledge of communication protocols is also needed, although at a different level from that described in most current networking texts. The goal of the tutorial is to provide the programmer with the basics requried to write network programs, and to develop and examine actual examples. The orientation is towards sockets and TCP/IP, since these are the most widely used today.

Specific topics covered include: overview of UNIX process handling, daemons, UNIX IPC, a networking primer, a user's view of the TCP/IP protocols, Berkeley sockets, comparison of sockets and System 5 TLI, remote command execution, pseudo-terminals and remote login, other client-server examples, and remote procedure calls.

Intended Audience: UNIX/C programmers interested in learning how to write programs that communicate across a network. No networking knowledge is assumed, since the basics that are required are presented in the tutorial.

Richard Stevens received his Ph.D. in Systems Engineering from the University of Arizona in 1982. From 1975 to 1982 he was with Kitt Peak National Observatory in Tucson where he started using UNIX on PDP-11s in 1978. Since 1982 he has been with Health Systems International in New Haven, Connecticut, where he is Vice-President of Research and Development.

Tutorial M2 – RISC

Tutor: Jean Wood and Ashis Khan, MIPS Inc.

The RISC architecture revolutionized the methodology of designing of the Central Processing Unit. During the pre-RISC era, the commercial processors used to be designed by semiconductor houses that had tremendous expertise in hardware design and integration but lacked compiler and O/S knowledge. This resulted in hardware that was very sophisticated but useless as far as software is concerned.

RISC technology introduced the idea of looking at HLL compiler output and O/S requirements before designing the Processor and then implementing the minimum hardware required to satisfy these requirements. Also, if some requirements by software are very infrequently used, they are left out from hardware implementation and relegated to software. This approach however put a great deal of burden on architects who must judge which feature to put on the chip without sacrificing any performances.

This seminar will introduce various architectural choices and their relative merits, such as optimal number of registers, cache matrices, etc. and discuss the software factors such as optimizing compilers and UNIX Kernel implementations for RISC chips. A comparative study of contemporary RISC architectures will be presented.

The seminar will be conducted by Ashis Khan and Jean Wood of the Technology Products Group at MIPS computer Systems in Sunnyvale, California. Ashis provides consultation on system designing with MIPS RISC architecture and conducts seminars worldwide on contemporary RISC designs. He was a Senior Design Engineer in the 80386 Team at Intel Corporation, where he also worked on peripheral controller, graphics and other microprocessor products. At Chips and Technology, he defined and marketed LCD controllers and other chip sets for Lap Top controllers. Jean was an original member of the UNIX Engineering group at Digital Equipment Corporation. She was part of the ULTRIX and VAX C development teams and was 'loaned' to the University of California's Computer Systems Research Group. She set up DEC support efforts for ULTRIX and other UNIX based products in Europe. She joined MIPS in November of 1988, where she provides consultation on software component products.

Jean Wood has many years' experience of working for American companies, including DEC and MIPS Computer systems, both in Europe and the States. She has also a long history of involvement with EUUG and EUUG Conferences in particular. She and her colleague Ashis Khan are well qualified to present this tutorial on RISC.

TUTORIALS

Monday 18th September

Registration from 08.00

Tutorial M3 – System V IPC (Morning only)

Tutor: George Philippot, NCR Norway

In principle, processes under UNIX run in their own independent virtual machines. Originally, the only way for them to communicate was through the file system or by sending signals to each other.

To overcome the performance penalty of these crude methods, System V introduced three mechanisms for IPC: Shared memory, message queues, and semaphores. This tutorial covers these and a few related methods as follows:

- Binding mechanism
- Shared memory
- Message queues
- FIFO special files
- Semaphore resources and programming
- Critical and restricted regions
- Mailboxes
- File locking

This tutorial is aimed at programmers with some experience in C, though not necessarily under UNIX. UNIX concepts involved will be explained as necessary.

Tutorial M4 – MAKE (Afternoon only)

Tutor: George Philippot, NCR Norway

The MAKE command is an excellent tool for automating the generation and testing of programs. Programming in MAKE is done on a higher level than that of ordinary Shell, in that you do not explicitly specify a sequence of actions to be taken. Instead, you give the rules, and MAKE figures out which ones to execute and in what sequence.

The following topics are covered:

- Simple invocation of MAKE
- Programming in MAKE
- Macros
- User defined rules
- Recursion
- Including other files in a makefile
- Automatic detection of C #include
- Testing
- MAKE's default definitions

This tutorial is aimed at programmers who are comfortable with Shell programming. No knowledge of any other particular programming language is required, since MAKE is a general tool that works equally well with programs, documents, etc.

George Philippot is a senior consultant at NCR Norway. He has a Siv.ing. (Master of Science) degree in computer science and has worked with UNIX at NCR for four years. He develops and teaches advanced C and UNIX classes, gives customer and internal support, and develops software.

EUUG

European UNIX® systems User Group

PROGRAMME OF EVENTS

TUTORIALS

Tuesday 19th September

Registration from 08.00

Tutorial T1 – Beyond 4.3 BSD

Tutors: Mike Karels and Kirk McKusick

This tutorial will present a detailed discussion of several topics that have not been covered in depth in recent 4.3 internals tutorials. In particular, it will emphasize changes to the system that are available in the 4.3BSD Tahoe release or that are being developed for a future BSD release. The tutorial will cover four major topic areas. First will be an overview of the revised internal structuring of the networking and the ongoing development of OSI protocol support. Second will be a description of the four major changes in the 4.3BSD Tahoe release: a kernel memory allocator, disk labels, a kernel debugging facility, and filesystem modifications to support Gigabyte disks. Third will be an overview of the changes being made to support the POSIX P1003.1 interface including job control and a new terminal line discipline. Fourth will be other system changes in progress including the vnode-like file system interface, the filesystem reorganization, and the internal kernel reorganization. Presentations will emphasize system organization, data structure navigation, and algorithms.

This tutorial is directed to systems programmers who have taken a course on 4.3 internals or who have had at least a year of experience working on the 4.3 kernel. No licence is required for this tutorial.

Mike Karels received his B.S. in Microbiology at the University of Notre Dame. While a graduate student at the University of California, he was the major contributor to the 2.9BSD release of the Berkeley Software Distribution for PDP-11's. He currently is the Principal Programmer at the Berkeley Computer Systems Research Group, continuing the development of future versions of Berkeley UNIX.

Kirk McKusick got his undergraduate degree in Electrical Engineering from Cornell University. His graduate work was done at the University of California, where he received Masters degrees in Computer Science and Business Administration, and a Ph.D. in the area of programming languages. While at Berkeley he implemented the 4.2BSD fast file system and was involved in implementing the Berkeley Pascal system. He currently is the Research Computer Scientist at the Berkeley Computer Systems Research Group, continuing the development of future versions of Berkeley UNIX. He is a director of the Usenix Association, a member of the editorial board of UNIX Review Magazine, and a member of ACM and IEEE.

Tutorial T2 – The Andrew Toolkit

Tutor: Nathaniel Borenstein, ITC CMU

The Andrew Toolkit (ATK) is a powerful set of tools for building complex user interface programs in a window-system independent manner. Besides providing many basic building blocks for such interfaces, the ATK also provides substantial facilities for object-oriented programming, which greatly simplify the process of constructing new interfaces. However, these facilities require the programmer to approach his programs in ways that may not be familiar to most C programmers. Just as it is possible for a LISP programmer to write bad C programs because he still thinks in LISP, so too the ATK programmer must free himself from certain preconceptions.

In this tutorial, we will learn how to use the Andrew Toolkit. The primary emphasis will be on thinking like an ATK programmer. Within this conceptual framework, we will then study the various components, or building blocks, that the ATK offers, and how they may be put together to form complex applications.

This tutorial is intended for experienced C programmers who know nothing about programming with the Andrew Toolkit. Some familiarity with the concepts of object-oriented programming would be helpful, but is not required. The tutorial will explain enough about the ATK to allow such people to begin serious application development using the ATK, if they are so inclined.

Those with prior experience with the ATK are welcome to attend, but should not expect extended discussion of advanced or csoteric ATK features at this tutorial.

Nathaniel S. Borenstein is currently Manager of Applications Development at the Information Technology Center and a Lecturer in the Computer Science Department at Carnegie Mellon University. (However, by the time of the conference he will be a Member of Technical Staff at Bell Communications Research in Morristown, New Jersey.) He received his Ph.D. in computer Science from Carnegie Mellon with a dissertation entitled "The Design and Evaluation of On-line Help Systems" in 1985. He is the author of the forthcoming books.

TUTORIALS

Tuesday 19th September

Registration from 08.00

Tutorial T3 – OSI

Tutor: Colin I'Anson, Hewlett Packard

This tutorial covers the general principles of OSI with a special emphasis on the upper layers. Aimed at providing a practical user introduction it will look at the general system infrastructure and the services provided. Therefore it will cover the basics of OSI, management, security, network environment, transport, session, presentation, common application service elements, before considering some of the specific applications in more detail. The user applications will include message handling (X.400), directory services (X.500) and file transfer access and manipulation (FTAM).

A basic knowledge of common networking protocols like X.25 will be assumed although this is not an essential prerequisite.

Session will be structured as follows:

Session 1:
- Introduction to OSI
- Reference model
- Naming and addressing
- Management
- Security
- Network

Session 2:
- Transport
- Session
- Presentation
- ASN.1

Session 3:
- Application
- ROS
- ACSE
- CCR
- MHS

Session 4:
- Directories
- FTAM

Colin I'Anson is a project manager in Hewlett Packard Laboratories Europe. He joined Hewlett Packard in Scotland in 1980, immdeiately after completing his PhD studies at the University College of North Wales, and moved to his current position in Bristol in 1987. He has participated in a number of OSI standards committees in the CCITT, ISO and BSI and in the last two years has managed a team that has successfully prototyped a number of the X.400 1988 extensions (Message Store, X.500 integration and secure messaging). He is a member of the Institution of Electrical Engineers and a Chartered Engineer.

Wednesday 20th September

Registration from 08.00 AUDITORIUM MAXIMUM

Provisional Technical Programme

Wednesday
20th September

9.30	Keynote Professor Ahmed Elmagarmid
coffee-break	
	STANDARDS
11.00-11.30	Dominic Dunlop Are Standards the Answer?
	MULTI-PROCESSOR SYSTEMS
11.30-12.00	Andrew R. Huber, Michael H. Kelley Engineering a (Multiprocessor) UNIX Kernel
12.00-12.30	Ph. Bernadat An implementation of STREAMS for a Symmetric Multiprocessor UNIX Kernel
lunch	
	USER INTERFACES
14.00-14.30	Martin D. Beer, Steven M. George, Roy Rada Developing Writing Tools for UNIX Workstations
14.30-15.00	Hans-Joachim Brede, Nicolai Josuttis, Achim Loerke Implementation of a Window Manager under X11R3
15.00-15.30	Michael Haberler, Martin Ertl Offloading A Mainframe or Teaching A Spreadsheet How To Access a Big Database
coffee-break	
	SECURITY
16.00-16.30	Dale A. Moir System Security Administration through Automation
16.30-17.00	Ernst Piller Lettermatrix for the selection of passwords through the user
17.00-17.30	Peder Chr. Norgaard On user experience with large-scale IP network
17.30-18.00	Alicia Dunbar Gronke How to Protect Your Software Through International Copyright Laws: Step-by-Step Instructions
	UNIX International BOF
18:30-19:30	Maurice Schwartz and Ed Chatlos Introduction to UNIX International, aims and objectives, and discussion on SV.4, open panel

Thursday 21st September

Registration from 08.00 AUDITORIUM MAXIMUM

Provisional Technical Programme

Thursday
21st September

	UNIX IN GERMAN SPEAKING COUNTRIES
9.00-9.30	Wolfgang Kabelka UNIX in German speaking countries
	UNIX MODELLING
9.30-10.00	Gail Anderson, Paul Chung, Anita Lunderberg Using an Object-Oriented Model of UNIX for Fault Diagnosis
10.00-10.30	Florian Dupre-Blusseau Modelling the NFS service on an Ethernet local area network
coffee-break	
	RISC ARCHITECTURES
11.00-11.30	Daniel V. Klein A Comparison of Compiler Utilisation of Instruction Set Architectures
11.30 -12.00	Kurt Judmann On the evaluation of the performance of RISC systems
12.00-12.30	Martin Lippert SPARC A Scalable Processor Architecture
lunch	
	TOOLS
14.00-14.30	Sunil K. Das, Ali Harlow An Interpreter for C
14.30-15.00	Ralph Zainlinger, Thomas Hadek A SQL Programming Interface for the Relational Database System db++
15.00-15.30	Jaap Akkerhuis, Ann Marks, Jonathan Rosenberg, Mark S. Sherman Processable Multi-media Document Interchange
coffee-break	
	NETWORK MANAGEMENT
16.00-16.30	Josef Matulka TCP/UDP Performance as Experienced by User-Level Processes
16.30-17.00	Philippe Blusseau Interconnection of LANs, using ISDN, in a 'TCP/IP' architecture
17.00-17.30	Georg-Michael Raabe, Carol Rosenstock System Administration of UNIX Networks: Two Approaches to Supporting the Management of Large, Distributed, Multi-Vendor Networks
17.30-18.00	D. Schenk, G. Reichelt, A. Pikhard Resource Management System for Networks

Friday 22nd September

Registration from 08.00 AUDITORIUM MAXIMUM

Provisional Technical Programme

Friday
22nd September

	GRAPHICAL USER INTERFACES
9.00-9.30	Nannette Simpson Porting Applications to XVIEW and OPEN LOOK (tm)
9.30-10.00	Timothy L. Ehrhart X Display Servers: Comparing Their Performance and Architectural Differences to Diskless Workstations
10.00-10.30	Robin Faichney A System for the Redirection of Graphical User-Interaction
coffee-break	
	TRANSACTION PROCESSING
11.00-11.30	C. Boccalini, J. Marino, M. Paolucci Performance Analysis for Shared ORACLE Database in UNIX Environment
11.30-12.00	Heike von Luetzau-Hohlbein Transaction Monitor for SINIX
12.00-12.30	Giandomenico Spezzano, Domenico Talia An Approach to Reliability in Distributed Programming
lunch	
	OBJECT-ORIENTED SYSTEMS
14.00-14.30	Donal Daly, Chris Horn, Vinny Cahill Object-Oriented Distributed Systems
14.30-15.00	Marco Menichetti XEiffel: an object-oriented graphical library and Open Look toolkit
15.00-15.30	D. Decouchant, E. Paire, M. Riveill Efficient implementation of low-level synchronisation primitives in the UNIX-based Guide kernel
	EUUG AND USENIX
15.30-16.00	John S. Quarterman Social Aspects of EUUG and USENIX

Calendar of UNIX Events

This is a combined calendar of planned conferences, workshops, or standards meetings related to the UNIX operating system. Most of this information came from the various conference organizers, although some was taken from ;login: (USENIX), 13, 1, Jan/Feb 1988, CommUNIXations (/usr/group), VII, 6, Nov/Dec 1987, and the /usr/group UNIX Resources Guide.

If you have a UNIX related event that you wish to publicise then contact either John Quarterman at *jsq@longway.tic.com* or Alain Williams at *addw@phcomp.co.uk* giving brief details as you see below.

Abbreviations:

C	Conference
G, MD	Gaithersburg, Maryland
S	Symposium
T	Tradeshow
U	UNIX
W	Workshop
UG	User Group

year mon days	conference	(sponsor,) (hotel,) location
1989 Sep 5-6	Usenix W Distributed Systems	Ft.Lauderdale, FL
1989 Sep 7-8	Sun UK UG	C Manchester, UK
1989 Sep 7-9	USENIX Sys Admin	W Austin, TX, USA
1989 Sep 18-22	EUUG	WU Vien, Vienna, Austria
1989 Sep 19-22	ACM SIGCOMM	Austin, TX
1989 Sep 26-28	GUUG	CT Wiesbaden, Germany
1989 Oct 16-20	IEEE 1003	Brussels, Belgium
1989 Oct	UNIX Expo	New York, NY
1989 Oct 5-6	USENIX -Distrib Proc	Ft. Lauderdale, FL, USA
1989 Oct 31-Nov	2 IETF	IAB, U. Hawaii, Honolulu, HI
1989 Nov 1-3	UNIX EXPO	Javits Conv. C, New York, NY
1989 Nov 6-10	DECUS S	Anaheim, California
1989 Nov 9	NLUUG C	The Netherlands
1989 Nov 9-10	14th JUS UNIX Symposium	Osaka, Japan
1989 Nov 16-17	USENIX Graphics	Monterey, CA, USA
1989 Nov 24	AFUU C	Paris, France
1989 Dec 5-6	JUS UNIX Fair 89	Tokyo, Japan
1989 Dec 8-9	Sinix C	Singapore
1989 Dec 11-13	UKUUG C	Cardiff, Wales, UK
1990 Jan	U in Gov. C&T	Ottawa, ON
1990 Jan 22-26	USENIX	Washington, DC
1990 Jan 23-26	UniForum	Washington Hilton, Washington, DC
1990 Jan 29	IEEE 1003	New Orleans, LA
1990 Feb 6-8	IETF	IAB, (FSU, Talahassee, FL)
1990 Mar 27-30	AFUU	Paris, France
1990 Apr	USENIX C++ Conference	California – tentative
1990 Apr	IEEE 1003	Montreal, Quebec

1990 Apr 23-27	EUUG	Munich, Germany
1990 May 2-4	IETF	IAB, (U. Washington, Seattle, WA)
1990 May 7-11	DECUS S	New Orleans, Louisiana
1990 May	U 8x/etc C&T	/usr/group/cdn, Toronto, ON
1990 Jun 11-15	USENIX	Marriott, Anaheim, CA
1990 Jul 11-13	UKUUG C London	
1990 Jul 31-Aug	2 IETF	IAB, ?, not in North America
1990 Sept 11-14	AUUG Conference	Southern Cross, Melbourne, Australia
1990 Oct 22-26	EUUG	Nice, France
1991 Jan 21-25	USENIX	Dallas, TX
1991 Jan 22-25	UniForum	Infomart, Dallas, TX
1991 Feb	U in Gov. C&T	Ottawa, ON
1991 May	U 8x/etc C&T	Toronto, ON
1991 May 20-24	EUUG	Tromso, Norway
1991 Jun 10-14	USENIX	Opryland, Nashville, TN
1991 Sept 16-20	EUUG	Budapest, Hungary
1992 Jan 20-24	USENIX	Hilton Square, San Francisco, CA
1992 Jan 21-24	UniForum	Moscone Center, San Francisco, CA
1992 Spring	EUUG	Jersey, UK
1992 Jun 8-12	USENIX	Marriott, San Antonio, TX
1993 Jan	USENIX	Town & Country, San Diego, CA
1993 Mar 2-4	UniForum	Washington, D.C.
1993 Jun 21-25	USENIX	Cincinnati, OH

Organising Bodies

NIST/NBS/POSIX
Roger Martin
National Institute of Standards
and Technology
Technology Building, Room B266
Gaithersburg, MD 20899
+1-301-975-3295
+1-301-975-3295
rmartin@swe.icst.nbs.gov

IEEE Computer Society
P.O. Box 80452
Worldway Postal Center
Los Angeles, Ca. 90080

/usr/group
4655 Old Ironsides Drive, Suite 200
Santa Clara, California 95054
U.S.A.
+1-408-986-8840
+1-408-986-1645 fax

/usr/group/cdn
241 Gamma St.
Etobicoke, Ontario M8W 4G7
Canada
+1-416-259-8122

Tracy MacIntyre
Exhibition Manager
EMAP International Exhibitions Ltd.
Abbot's Court
34 Farringdon Lane
London EC1R 3AU
United Kingdom
+44-1-404-4844

AUUG
P.O. Box 366
Kensington
N.S.W. 2033
Australia
uunet!munnari!auug
auug@munnari.oz.au
+61 3 344 5225

AMIX, c/o IPA
P.O. Box 919
Ramat-Gan
Israel, 52109
+972-3-715770
+972-3-715772
amix@bimacs.bitnet
amix@bimacs.biu.ac.il

Japan UNIX Society (JUS)
#505 Towa-Hanzomon Corp. Bldg.
2-12 Hayabusa-cho
Chiyoda-ku, Tokyo 102
Japan
bod%jus.junet@uunet.uu.net
+81-3-234-5058

UNIX Fair '88 Association
1-1-1 Hirakawa-chu,
Chiyoda-ku, Tokyo 102
Japan

Singapore UNIX Association - Sinix
20 Bideford Road #11-05
Wellington Building
Singapore 0922
+65 734 3256

DECUS U.S. Chapter
219 Boston Post Road, BP02
Marlboro, Massachusetts 01752-1850
U.S.A.
+1-617-480-3418

USENIX Association Office
2560 Ninth St., Suite 215
Berkeley, CA 94710
USA
+1 415 528 8649
office@usenix.uucp

Sun UK User Group
Sue Crozier
Sun Microsystems, UK
+44 276 62111

EUUG National group addresses can be found on the back cover of this newsletter.

EUUG

European UNIX® systems User Group

4.3BSD MANUALS

The USENIX Association now kindly offers all members of the EUUG the opportunity to purchase 4.3BSD Manuals.

The 4.3BSD manual sets are significantly different from the 4.2BSD edition. Changes include many additional documents, better quality of reproduction, as well as a new and extensive index. All manuals are printed in a photo-reduced 6" x 9" format with individually coloured and labelled plastic 'GBC' bindings. All documents and manual pages have been freshly typeset and all manuals have 'bleed tabs' and page headers and numbers to aid in the location of individual documents and manual sections.

A new Master Index has been created. It contains cross-references to all documents within the other six volumes. The index was prepared with the aid of an 'intelligent' automated indexing program from Thinking Machines Corp. along with considerable human intervention from Mark Seiden. Key words, phrases and concepts are referenced by abbreviated document name and page number.

While two of the manual sets contain three volumes, you may order complete sets only.

The manuals are available now. To order return a completed '4.3BSD Manual Reproduction Authorisation and Order Form' to the EUUG secretariat along with your remittance. You must be an EUUG member.

The EUUG has bulk shipped these manuals from the USA thereby saving you 5 kg transatlantic postage.

Manual	Cost
User's Manual Set (3 Volumes) User's Reference Manual User's Supplementary Documents Master Index	£25.00/set
Programmer's Manual Set (3 Volumes) Programmer's Reference Manual Programmer's Supplementary Documents, Volume 1 Programmer's Supplementary Documents, Volume 2	£25.00/set
System Manager's Manual (1 Volume)	£10.00

4.2BSD Manuals are No Longer Available

4.3BSD Manual Reproduction Authorisation

Date _____

As an EUUG member† in good standing, and pursuant to the copyright notice as found on the rear of the cover page of the Unix ®/32V Programmer's Manual stating that:

"Holders of a Unix ®/32V software licence are permitted to copy this document, or any portion of it, as necessary for licenced use of the software, provided this copyright notice and statement of permission are included."

I hereby appoint the USENIX Association/EUUG as my agents, to act on my behalf to duplicate and provide me with such copies of the Berkeley 4.3BSD Manuals as I may request.

Signed _____

Institution (if Institutional Member) _____

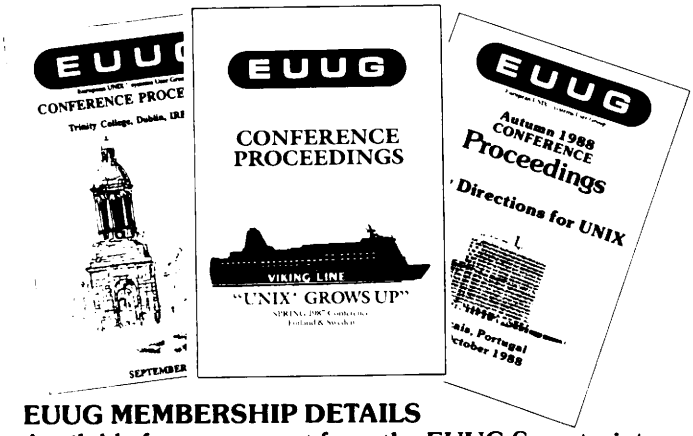
† Members of EUUG National groups are automatically members of the EUUG.

EUUG

European UNIX® systems User Group

ADDITIONAL PUBLICATIONS AVAILABLE FROM EUUG

Title		Cost
EUUG—CONFERENCE—PROCEEDINGS		
Spring 1983	— Bonn	£2.00
Autumn 1983	— Dublin	£2.00
Spring 1984	— Nijmegen	£5.00
Autumn 1984	— Cambridge	£5.00
Spring 1985	— Paris	£5.00
Autumn 1985	— Copenhagen	£10.00
Spring 1986	— Florence	£20.00
Spring 1987	— Finland/Sweden	£20.00
Autumn 1987	— Dublin	£20.00
Spring 1988	— London	£20.00
Spring 1989	— Brussels	£20.00
European E-Mail Directory		£18.00
4.3BSD MANUAL (Members only)	— User's Manual Set	£25.00
	Programmer's Set	£25.00
	System Manager's Manual	£10.00
Language C—Standard Proposal		£8.00
USENIX Conference Proceedings C++ —Santa Fe, 1987		£30.00



EUUG MEMBERSHIP DETAILS
Available free on request from the EUUG Secretariat.

UKUUG PROCEEDINGS
Winter 1988 –UKNet Meeting, Canterbury **£10.00**

If you wish to order any of the above publications, or receive details of EUUG Membership, please use the order form below and return it to the EUUG Secretariat, Owles Hall, Owles Lane, Buntingford, Herts. SG9 9PL, U.K. If you are ordering publications then your remittance should be enclosed by cheque, bank draft etc. made payable to EUUG.

ORDER FORM

Name _____

Address _____

_____

_____

_____

Phone _____

Network Address _____

The prices shown include surface postal charges. All payments must be in Sterling by means of a cheque drawn on a UK bank, or a Eurocheque, or instruction to charge your VISA card (quoting the number, the date of expiry, the name of the card holder, and the address which VISA use when corresponding with you).

Please send me _____ no. copy/ies of the following:

_____

_____

I would like to receive membership details of the EUUG YES/NO

User's Manual Set (3 Volumes) at **£25.00/set** = £ _____
Programmer's Manual Set (3 volumes) at **£25.00/set** = £ _____
System Manager's Manual (1 volume) at **£10.00/set** = £ _____
Total £ _____

We accept
Visa, Access, Euro Card, Master Card, UK cheques, and Euro cheques.

Please complete name and address of credit card holder if different from name and address on order form.

NAME _____

ADDRESS _____

_____

_____

Expiry Date: _____

Credit Card No:

I enclose my cheque, bank draft etc., in the sum of £ _____ and understand a receipt will be sent.

Make your cheque payable to 'EUUG' and post it with this order form

To: EUUG Secretariat, Owles Hall, Owles Lane, Buntingford, Herts. SG9 9PL, United Kingdom.
Tel: Royston (0763) 73039 +44 763 73039
Fax: Royston (0763) 73255 +44 763 73255

For BSD manuals please sign Manual Reproduction Authorisation Notice overleaf.

This page may be photocopied for use.

USENIX Online Index and Library

Deborah Scherrer
scherrer@mtxinu.com

mt Xinu
Berkeley
California

Deborah Scherrer is currently the USENIX Vice President and has been serving on its Board of Directors for many years. She is also a Computer Scientist for mt Xinu, Inc., where she works on their Mach-distribution project. She also serves on the mt Xinu Board of Directors, is a member of the *UNIX Review* Editorial Board and the *Computing Systems* Editorial Panel, has been a Contributing Editor for *UNIX/World*, was founder of the Software Tools Users Group, and always loves to come to EUUG meetings.

As the quantity of USENIX, EUUG, and related publications rapidly expands, and the quality becomes increasingly attractive, the need arises for some sort of online access, at least to an index, or list, of published articles. USENIX has recently funded and implemented an electronically accessible index to USENIX, EUUG, and related groups published articles. Versions of abstracts and/or full papers will be made available in cases where authors are willing to donate them.

Currently we have indexed all available issues of the following:

USENIX:
- Conference proceedings
- Workshop proceedings
- Computing Systems Journal
- Newsletters (;login:)

European UNIX User Group:
- Conference proceedings
- Newsletters

Software Tools User Group:
- Conference proceedings

Australian UNIX User Group:
- Newsletters

The *UNIX Review* periodical is currently being indexed and will soon be available. Other sources (JUS, AFUU, GUUG, NZUSUGI, etc.) are being collected and evaluated and will be included as deemed suitable.

The index is kept as a simple ASCII file, in refer/bib format, sorted by author. It contains information about title, authors, publications, page numbers, dates, and the usual bibliographic information. In some cases, electronically readable versions of full papers or abstracts are also available. If a paper is available online, this is indicated in its index entry. USENIX is currently soliciting authors of publications to donate their full paper, or abstract only, for inclusion and distribution through the online library. (When the paper retrieval capability is fully functional, we will announce the procedures.)

How to Get the Index:
The online index is currently available online from uunet, either via a mail server or anonymous ftp. The index is about 200K, and available only in entirety. To get it via electronic mail:

```
echo send bibliography | mail library@uunet.uu.com
```

A (non-human) server will automatically break

the index up into mailable chunks (if necessary), and return it to the sender of the mail.

Or, the index can be retrieved via anonymous ftp to uunet.uu.net:

```
ftp>  get library/bibliography
```

To get a help file:

```
echo help | mail library@uunet.uu.com
```

Or, to pick up the date the index was last changed:

```
echo send date | mail library@uunet.uu.com
```

Although the library and index will be most effectively used electronically, the index will also be printed periodically and made available through the USENIX office, for those without email access.

If You Have Ever Had a Paper Published by EUUG or USENIX:
Authors of articles in EUUG or USENIX publications are being given the opportunity to donate their full paper, or abstract only, for inclusion and distribution through the online library. Donation of either a paper or an abstract to the library is purely voluntary. Papers can be donated and distributed either with a copyright notice, or a copyright release, at the copyright holder's discretion. An appropriate release form, festooned with the appropriate signatures, is absolutely required before any paper or abstract will be made available electronically.

Authors wishing to donate papers or abstracts need to

1. Assure themselves (and us) that they own the copyright.
2. Sign the release form (attached).
3. Send an electronic version of the paper.

Note that the issue of copyright ownership should be investigated thoroughly by the donator—the original publisher may own the copyright, or multiple authors may share in copyright, or the author's institutions or funders may hold them. USENIX does not have the resources to research copyrights and will rely on the information you provide us on the signed release.

The electronic papers and abstracts will be made available for retrieval in essentially the same format they were delivered—no attempt will be made to translate into a common text formatting language, etc. (Understand that some illustrations will not be available electronically and that some people may not be able to properly print papers that they request.)

Authors may choose to donate abstracts only, even in cases where a full paper has been published. Authors donating abstracts only also need to sign the form and email a copy of the abstract, if they have one. If the abstract is not available electronically, and the author(s) gives permission for publication, USENIX will key in a copy from the publication itself (assuming the abstract had been published). USENIX does *not* have sufficient resources to key in full papers.

Please check your memories for any papers you may have had published by the EUUG, USENIX, or related organisations, and sign the attached form giving permission for us to make available at least the abstract, and the paper if you wish. If you allow us to publish an abstract, send us either an online or a hard copy version. However, if you do give us permission to make available an entire article, you will have to send us an electronic copy of that article.

More Information:
For additional information about the online index and library, and/or specific instructions for donating papers, contact:

```
usenix!index (index@usenix.org)
```

or write to the USENIX Association, 2560 Ninth Street, Berkeley, CA 94710, USA.

USENIX Online Library

Author Release Form

☐ I hereby give the USENIX Association permission to republish the following **articles**, as well as previously published abstacts of these articles, bearing the title and containing the copyright notice provided below. I also give my permission for the USENIX Association to prepare excerpts of the article. I authorise the USENIX Association to make these materials available electronically to subscribers of the UUNET Communications Services. I am the owner/authorised representative of the owner of all copyrights in the listed articles and I have full authority to grant the USENIX Association permission to republish it.

☐ I hereby give the USENIX Association permission to republish the **abstracts only** from the following articles, containing the copyright notice provided below. I authorise the USENIX Association to make these materials available electronically to subscribers of the UUNET Communications Services. I am the owner/authorised representative of the owner of all copyrights in the listed abstracts and I have full authority to grant the USENIX Association permission to republish it.

☐ The articles and/or abstracts named below are in the public domain and you do not need my consent to publish them.

☐ I explicitly do **NOT** give my permission for the USENIX Association to republish my article or the abstract or to publish excerpts thereof.

Please tick appropriate box.

Signature: _____ Date: _____

Printed Name: _____

Address: _____

_____

_____

Phone number: _____ Electronic mail address: _____

Paper Title: _____

Author(s): _____

Where/when published: _____

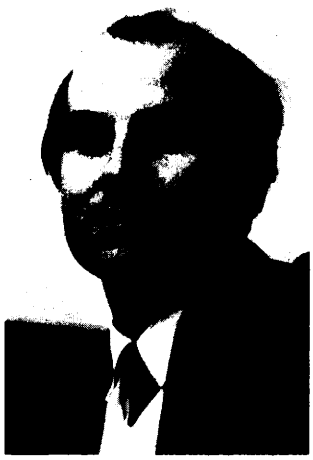
Copyright Notice: _____

(Please photocopy this page for use)

ISO JTC1 SC22 WG 15 (POSIX) Meeting, Ottawa

Dominic Dunlop

The Standard Answer Ltd.



Equipped with an undergraduate degree in Electrical Engineering from the University of Bradford in England, Dominic sidled into the world of mini- and micro-computers. From there, he managed to effect an entry into the hallowed temples of UNIX, and has hung around there ever since, writing the odd paper, contributing to the odd standard, and starting the odd company. He became an independent consultant in January, 1989, and hopes to have the money to buy his latest company, The Standard Answer Ltd., its own UNIX computer Real Soon Now.

ISO JTC1 SC22 WG 15 (POSIX) Meeting, Ottawa
1st-3rd May, 1989
'Snitch Report' to EUUG and USENIX

Red Flag Items

1. The Comité Européen pour la Normalisation (CEN—European Committee for Standardisation) is in the process of voting on a proposal from West Germany that the whole of the *X/Open Portability Guide, Third Edition, 1988* (XPG3) should become a 'draft European Prestandard'—one step away from being a European standard. (Conformance to European standards is almost mandatory for purchases made by European Community government organisations, and is strongly recommended in European Free Trade Association member governments.) This idea seems half-baked, not least because XPG3 covers a lot of ground, overlapping and conflicting with several existing European standards or prestandards. Since X/Open is committed to alignment with international standards as they appear, to have CEN, an international body, aligning with X/Open would introduce an unmanageable circularity. Consequently, the ISO POSIX working group has, in effect asked CEN to drop consideration of XPG3 in favour of the draft POSIX standard.
2. The International Organisation for Standardisation (ISO) POSIX working group has recommended that ISO should adopt draft IEEE standard 1003.2, *Shell and Application Utility Interface for Computer Operating System Environments* as a 'draft proposal' in September. Effectively, this means that the shell and tools have started on their journey to becoming an international standard.
3. The working group has decided *not* to recommend that ISO make an early start towards standardisation of 'an object-orientated language based on C'. No agreement could be reached on whether such a language should be
   - — C++ or something else (such as Objective C): and
   - — Constrained to be a true superset of ANSI C or not so constrained.

4. While, for reasons of verifiability, the working group wants to work towards the specification of POSIX in a Formal Definition Language, rather than in a less formal language, or in any particular computer language, it recognises that this can only be a long-term goal. Consequently, a message of comfort has been sent to the IEEE's 1003.1 group, encouraging it to continue in its work on a language-independent—but not strictly formal—definition. This should allow the IEEE to produce the first edition of the 1003.4 Real-Time standard in a language-independent form.

5. ISO appears to be setting up a new sub-committee concerned with all aspects of computer security (including both operating systems and communications). The POSIX group is working to ensure that the work of the new group does not conflict with the security requirements of POSIX, as developed by IEEE 1003.6.

6. Following the formation of two new IEEE working groups—1003.10, *Supercomputing Application Environment Profile*, and 1003.11 *Transaction Processing Application Environment Profile*, the ISO working group has been asked to consider its attitude to such profiles—definitions of application-specific variants or enhancements of an underlying POSIX-compliant operating system.

Introduction

This is the first of a series of reports which I shall be making on the activities of (pause for deep breath) Working Group 15 of Sub-Committee 22 of Joint Technical Committee 1 of the International Organisation Standardisation and the International Electrotechnical Commission (ISO/IEC JTC1/SC22/WG15). It is this group which is taking the work of the Institute of Electrical and Electronic Engineers (IEEE) on POSIX, a portable operating system interface, from its current official status as an American national standard to its final goal as an international standard. I have been sponsored by the European UNIX systems User Group (EUUG) and USENIX to attend the meetings of the working group on your behalf, representing your views and reporting back on developments which affect your interests. In these reports, I shall be asking for feed-back from you. As I write, there is no formal mechanism in place to handle this feed-back, so send mail to me directly for the moment. My address is *domo@sphinx.co.uk*.

Meeting Report

Hosted in Ottawa by the Standards Council of Canada, May's three-day meeting of ISO/IEC JTC1/SC22/WG15 was attended by five 'technical experts' (representatives) from the USA, three from the UK, two from Denmark, and one each from Canada, France, Japan and the Netherlands. There were three 'invited experts': myself, invited by the UK delegation to represent the EUUG and USENIX; Shane McCarron, invited by the USA on behalf of UNIX International; and Mike Lambert of X/Open Company Ltd.

Mike was invited by Jim Isaak, convener of the working group, to set out X/Open's mission and its position in relation to ISO's activities. It was clear that this was necessary as, in the responses to a previous ballot on the working group's work-in-progress, several respondents effectively asked "Why are we doing this? Doesn't it duplicate the work of X/Open?" What is more, CEN is voting on the adoption of XPG3 in its entirety as a 'draft European Prestandard'—see *Red Flag Items* above. (In fact, there is officially no such beast as a draft European Prestandard; there are 'Draft Standards' and 'Prestandards'. It seems that Prestandard is the intended meaning.)

X/Open's position is clear: "X/Open is not", as the preface to each XPG volume states, "a standards-setting organisation." Instead, X/Open is committed to align itself with international standards as soon as these are agreed, suggesting that its members adhere to other, less formal, national or de-facto standards only when no international standard is in place. In order that national and international standards can be arrived at in a timely manner, X/Open fully endorses the activities of organisations such as the IEEE, ANSI and ISO, and provides resources to aid in their activities, as it has done—and continues to do—in the case of the IEEE's 1003 (POSIX) developments. Consequently, the Working Group considers that it is inappropriate for an international standards body such as CEN to align itself with the XPG; the XPG is not itself intended to be a formal standard, but rather a series of moving pointers to other standards. As such, it

performs a valuable service to industry by indicating areas where more formal standardisation work should take place in the future. Each XPG pointer keeps moving until the area it addresses has become the subject of an agreed international standards. It is unlikely that CEN would tolerate such moving pointers, and would effectively freeze the XPG in its current state.

Another problem is that XPG3 specifies C, COBOL and FORTRAN—languages covered by other European Standardisation efforts. It also calls out communications protocols, media formats and a graphics interface (X) which may or may not overlap or conflict with other standards. It is not clear that these matters were considered before CEN moved to a vote.

Happily, well-defined mechanisms exist for communication between ISO and CEN, and "maximum alignment with ... ISO ... DP9945" is a requirement of the European Community's 'order form' to CEN requesting that a POSIX-based European Standard be produced. The working group is using the channels to suggest that DP9945, and, in the near future, the draft IEEE 1003.2 standard, replace XPG3 in their deliberations.

The issue of C++ standardisation was raised in the working group, as there was a (rather vague) feeling that object-oriented facilities were essential for future developments in operating systems, user interfaces, communications systems... well, most things, really. WG15's parent, subcommittee 22, has responsibility for language standardisation. A resolution was drafted recommending that work be started on standardisation of an object-orientated programming language based on C. (The bulk of any such work would probably be farmed out to ANSI, just like the work on C itself.) However, several valid objections resulted in the resolution being dropped:

— It is not clear whether the best basis for such a standard would be AT&T's C++, Stepstone's Objective C, or something else. (The issue is known to excite religious fervour.)

— It is not clear whether or not the language (whatever it is) should be constrained to be a superset of C. Such a constraint would be desirable from the point of view of compatibility, but might compromise the ideological soundness of the language. (Religion again.)

— The business of WG15 is the definition of an operating system interface. It should not concern itself with the means of implementation of an operating system which presents that interface—even if almost everything that conforms to the definition happens to be written in on particular language—C.

All this may seem to be somewhat arcane—distanced from reality. What it boils down to is that WG15 does not think that the time is yet ripe for international standardisation of an object-oriented C derivative. More work needs to be done by industry groupings and national standards bodies—and more users need to vote with their feet—before the terms of reference for an international standard become clear.

The working group discussed the path towards a language-independent definition of POSIX, an issue which took on added urgency because the working group's decision was required in order that the IEEE could determine the initial format of its 1003.4 standard (real-time extensions to 1003.1), which moves to ballot in January, 1990. Like IEEE 1003, WG15 intends that the standards it produces should *ultimately* be expressed in a form which is independent of any particular computer language. And also like 1003, WG15 is currently drafting standards in terms of the C language. Two questions arise: how independent, and how ultimate?

IEEE 1003.1 is working towards removing C-language dependencies from Std. 1003.1-1988, but is stopping some way short of using a Formal Definition Language (FDL). While this precludes the automatic generation of test procedures which would be possible, were a verifiable FDL is used, it is do-able in the short term. Soon enough, in fact, to allow 1003.4 to go to ballot in a language independent form. If 1003.1 were to drop this work in favour of a FDL, results would be postponed for some years, and 1003.4 would have to be defined in terms of the C language, much to the distress of the Ada community.

WG15 decided that use of a FDL was most appropriate to an international standard. Consequently, the group had to decide whether it wanted

a. to ignore 1003.1's work (which could result in 1003.1 dropping the activity);

b. to recommend that 1003.1 adopt a FDL (with a resultant gross delay); or

c. to use 1003.1's work as a basis for subsequent WG15 progress towards a formal description of POSIX interfaces.

The last option was chosen, resulting in a resolution which exhorts 1003.1 to keep up the good work. Expect 1003.4 to be language-independent.

For its part, WG15 is going to look into FDLs—a particularly esoteric subject—in more detail at its next meeting in Brussels in October. Ultimately, its standards will have three levels:

— Formal description (verifiable, but almost incomprehensible to mere mortals);

— Informal, but computer language-independent, commentary; and

— Series of language bindings, which may or may not implement the whole interface. (For example, a COBOL binding might well exclude the *fork* interface.)

This should keep us busy well into the 1990s.

ISO, in order that it can exercise adequate control of activities dispersed both geographically and in time, tries to compartmentalise as much as possible, making sure that the responsibilities of each sub-committee and working group are very well defined. The trouble is that there are certain topics which just cannot be pushed into a single compartment; internationalisation is certainly one, affecting as it does almost every aspect of information technology; security—an issue which currently has many people extremely worried—is probably another. Despite this, ISO JTC1, having decided that the issue needs an identifiable home, is thought to be about to convene a new working group—probably WG27—to handle all aspects of security. (There is much vagueness here: JTC1's mailing mechanism appeared to have failed, with the result that nobody was sure exactly what would be voted on at its meeting in Paris in late July.)

Of course, this has WG15 worried, both in its own right, and on behalf of other groups and sub-committees affected by issues of security. (Most notable among these is SC21, which manages the burgeoning ISO protocol stack.) Consequently, a resolution has been forwarded to JTC1 via SC22 saying, in effect "We're in this together. Let's work together." The means of working together is a rapporteur group, a mechanism which exists to allow one group to monitor the activities of another. WG15 has such groups covering verification and internationalisation as well as security.

Jim Isaak, convener of WG15, is much concerned with the issue of *functional standards for applications portability*, or *Application Environment Profiles* (AEPs). Jim chairs IEEE 1003.0, which, in effect, is stocking the shelves of a standards supermarket from which users can pick the selection (or profile) needed to allow applications of a particular type to be realised in a portable manner. (X/Open, The Open Software Foundation and more than a few governments are doing much the same sort of thing.) One example of such a profile might satisfy the needs of applications requiring distributed database services with reliable transaction processing and high security. (Continuing the supermarket analogy, these would be shopping lists, each allowing the execution of a number of recipes—applications... Never mind.)

Already, the IEEE has working groups which are defining AEPs: 1003.10 for supercomputing and 1003.11 for transaction processing, and Jim is engaged in selling the idea to ISO. Again, there are two questions: "Are you interested?" and "If so, what profiles do you want to specify?"

It is early days yet: the issue is to be raised at Technical Study Group 1's (TSG1's) meeting in Essen, Germany, in September. (TSGs are another ISO mechanism which is brought into play to handle interdisciplinary issues.) TSG1 is developing a framework for application portability, so it should consider AEPs worth adopting. In the mean time, feedback concerning useful and desirable AEPs is solicited by IEEE 1003.0.

Finally, WG15 has decided that it is time to adopt IEEE's draft 1003.2 standard, *Shell and Application Utility Interface for Computer Operating System Environments* as the basis for its recently approved movement towards a corresponding international standard. A little procedural gymnastics is involved: the first SC22 meeting that could authorise such an adoption is in September, and it is not clear which draft of 1003.2 will be current at that time: if things go

badly it could be draft 8; if to plan, draft 9. Also, draft international standard 9945, which corresponds to IEEE 1003.1, must be renamed to 9945.1, allowing 1003.2 to form the basis of 9943.2. It took three separate resolutions to put this particular show on the road!

Those, then, are the issues I consider important to members of EUUG and USENIX. Beyond them, there was much procedural stuff—more, for example, than at an IEEE meeting, even though WG15 is apparently quite informal by ISO standards (sorry).

The Concurrent C Programming Language—Book Review

The Concurrent C Programming Language
Narain Gehani and William D. Roome
Silicon Press ISBN 0-929306-007 (US) Price $29.95, Paperback, 303 pp.
Reviewed by Lindsay F. Marshall, Computing Laboratory, University of Newcastle upon Tyne, UK.
email: Lindsay.Marshall@newcastle.ac.uk

Concurrent C has been around for several years but has not yet achieved much popularity. The publication of this book, which serves as both an introduction to the language and to some of the principles of concurrent programming, may help to change this.

As its name suggests, Concurrent C is yet another extension of C and the authors have made the wise decision to assume that readers are familiar with the basic language. This means that they can describe the new features in full detail without making the book too long. Their approach is to use examples to show how Concurrent C programs are built up and in the early chapters they lead out the old warhorses of parallelism—the dining philosophers, readers and writers, etc., etc. Not that there is anything wrong with these examples, but it might be nice to see some new ones appearing in the literature (for example the vomiting philosophers which uses backtracking to break deadlocks).

Once the language has been fully described the authors work through some larger examples. This section is excellent and the examples are interesting and relevant. There is a lot of raw code presented in this chapter, but it still manages to be readable. Sections covering discrete event simulation and models of concurrent programming are also good. The book even describes programming in Concurrent C++—an extension to an extension that comes free as the Concurrent C compiler is based on the cfront program used by C++.

The only real fault with this book is that it is rather dry and lacks the humour and sparkle that enlivens the best textbooks. The look and feel of the book will be familiar to most people as it was produced with the usual UNIX text processing tools. This does tend to give a cozy familiarity to the text which is jarred only by the Concurrent C language itself whose new features are rather too ADA like for my comfort, though the language seems much more flexible and easy to use than ADA tasking.

If you want to know more about Concurrent C this is a good book, which is lucky as there are no others. You may be a little stuck if you want to use the system though, as the book has no information about how to get hold of a copy.

Call Doc Strange

Colston Sanger
doc.strange@olibc1.oliv.co.uk

Olivetti International Education Centre

Colston Sanger is a lecturer at the Olivetti International Education Centre, Haslemere, UK and a visiting lecturer in the Faculty of Engineering, Science and Mathematics at Middlesex Polytechnic. After a recent domestic deluge he is very glad he bought a VAX...

The null column

It's summer and even in England it's just too hot to be `awking` and `greping` the night away — so this is a sort of *null column*, in which I chatter on for a page and a half about nothing in particular. (**Ed**: *thinks* ... as usual, *mutter*, *mutter*...)

Anyway, I've been playing with a PC running UNIX System V/386 Release 3.2 — you know, the AT&T and MICROSOFT joint effort that provides XENIX compatibility.

I quite like it. Mind you, there's an awful lot of new stuff: enhanced security, a virtual terminal manager, the Framed Access Command Environment (FACE) and its underlying FMLI, a shell-like, but object-oriented forms and menu language interpreter, a new LP spooler, *etc*, *etc*. These are only some of the obvious things, but there's enough in that list alone to make it seem as though you practically have to re-learn the whole UNIX system after each new release.

Enhanced security

Do you remember how to add a user in the old days? It was simple: you edited the `/etc/passwd` file with your favourite text editor, did a `mkdir` and a `chgrp` and a `chown`. Took about thirty seconds.

Can't do that anymore. Because now there's a shadow password file, `/etc/shadow`, and the two files `/etc/passwd` and `/etc/shadow` have to be in step. Here's the `/etc/passwd` file:

```
root:x:0:1:0000-Admin(0000):/:
daemon:x:1:1:0000-Admin(0000):/:
bin:x:2:2:0000-Admin(0000):/bin:
sys:x:3:3:0000-Admin(0000):/usr:
adm:x:4:4:0000-Admin(0000):/usr/adm:
.
.
.
colston:x:401:400:Colston Sanger:/usr2/colston:
```

It has permissions like a traditional `/etc/passwd` file, looks like a traditional `/etc/passwd` file — except for the 'x' where the encrypted password used to be. Here is `/etc/shadow`:

```
root:FzAe93HPeFqs6:7113:14:28
daemon:NONE:7113::
bin:NONE:7113::
sys:NONE:7113::
adm:NONE:7113::
.
.
.
colston:cQ5s8xuhi/1pI:7117:14:28
```

This is where passwords are stored in SVR3.2, and the file is readable only by `root`. The other three fields are — I think — the number of days since the beginning of time (1 January 1970) that the password was last changed, the minimum number of days that must elapse before it can be changed again and the maximum duration of the current password.[1]

What else? When you login, it checks the timestamp on the file `$HOME/.lastlogin` to tell you when you last logged in. Moreover, if the dreaded hacker attempts to login, a record is written to `/usr/adm/loginlog` (if the file exists).

On a related topic, it is now possible to set the 'sticky bit' on directories that are writeable by all users (typically, the public directories `/tmp` and `/usr/tmp`) with the effect that files in those directories can only be removed by their owners.

The virtual terminal manager

`vtlmgr`, the virtual terminal manager is a great improvement on the old `shl` shell layers. It's certainly impressive to be able to have FACE (of which more below) running in one virtual terminal, `sh` in another, a `cu` in yet another, while surreptitiously working out expenses with **Autoroute** in a DOS session under **Simul-Task 386** in yet another. All it takes is an ALT-SYSREQ-FKEY to switch from one to the other. Better still is running multiple DOS applications, each in its own virtual terminal. I've had **Autoroute** running in one virtual terminal and **SuperCalc 4** in another. I've also had **Autoroute** running on a dumb terminal (a **Wyse 50** with magic cookie glitch) — and I've seen **Lotus 1-2-3** running on a **Wyse 60**, though nobody ever believes me when I say so.

FACE

FACE, the Framed Access Command Environment, is nice. It's a bit like the user interface on the old 3B1 UNIX PC. It works in colour on the console, and in monochrome on a dumb terminal. At the moment I'm only using it for system administration, but it's really a desktop environment for naive users for organising files and invoking applications. For example, a user can store information in files and folders in a personal filecabinet (the standard metaphor!) and discard information no longer needed in a wastebasket.

FACE can be invoked directly when you login or from the shell. The first screen you see is a menu with options for Office of <login>,[2] Printer Operations, Programs, System Administration (if you have system administration privileges), MS-DOS, UNIX System or Exit. Choosing Office of <login> takes you to your personal filecabinet; Printer Operations is a front-end to the new LP Spooler package; Programs is for starting up programs or third-party applications — what appears here is customisable by a system administrator; and MS-DOS or UNIX System gives you a DOS (if you have it installed) or UNIX session.

FACE is actually an application built with the Forms and Menu Language Interpreter (FMLI). I can't tell you very much about FMLI because somebody has borrowed my *FMLI Programmer's Guide*, so what I do say is based on the notes from the UNIX System V Release 4 Software Developer Conference.

FMLI syntax is a bit like shell. The language is made up of *descriptors* or variable names that define the attributes of the form, field or menu you are developing. For example, here is a simple FMLI menu:

1. Yet two more files enter the story here: `/etc/default/login` and `/etc/default/passwd` which among other things (the *ulimit*, for example) contain parameters that determine whether a password is required, the minimum number of characters that it must have and default password ageing information.

   By the way, `passwd(1)` now has a set of options.

2. Where the <login> is your login name.

```
#Start a new menu
#
menu="AT&T FACE"

# Indicate where to look if they hit the HELP key
#
help=OPEN TEXT $VMSYS/OBJECTS/Text.face

# Add menu items, along with actions upon their selection
#
name="Office of $LOGNAME"
action=OPEN MENU $VMSYS/OBJECTS/Menu.office

name=Printer Operations
action=OPEN MENU $VMSYS/OBJECTS/Menu.printer

name=Programs
action=OPEN MENU $VMSYS/OBJECTS/Menu.programs

name=System Administration
action=OPEN MENU $VMSYS/OBJECTS/Menu.system

name=UNIX System
action=unix

# Dynamically generate user's own menu items (if defined)
#
`readfile $HOME/pref/office`

name=Exit
action=exit
```

What does the code do? Using the *menu* descriptor, it creates a new menu titled 'AT&T FACE', which you'll probably recognise as the one I described earlier. `menu` is an example of a *single instance descriptor*: one that can appear only once in an object definition file.

Next is the `help` descriptor, which tells FMLI where to get help text if the user presses the HELP key. `help` is another single instance descriptor.

After `help` comes a collection of `name`/`action` pairs. `name` is a *multi-instance descriptor*: one that can appear several times in a definition file. As you can see, each occurrence of the `name` descriptor adds an item to the menu.[3]

3. I just checked the real code and it is a bit more complex than this, but the general structure holds.

(I don't think I should pirate any more or You-Know-Who isn't going to be very happy, but I hope that's enough to give you the flavour of FMLI. Perhaps the best thing is to write a whole column on FMLI, after I've had time to get to grips with it.)

Some things that can catch you

I've mentioned `/etc/passwd` and `/etc/shadow`. If by chance the two do get out of step you can use `pwconv`(1M) to set them right.

I also mentioned the *ulimit* in passing. Yes, it is a tunable parameter, but it is also set in `/etc/default/login`.

Also, watch out for the 'XENIX compatibility'. For example, say you have a colour monitor on the console. So you change the value of `TERM` in `/etc/profile` and ... it's still in monochrome. Why? Because you also need to change it in

`/etc/ttytype`.[4]

Finally, be extra-careful with `/etc/inittab`! As part of the new Installable Drivers scheme in UNIX/386, `/etc/inittab` is rebuilt every time the kernel is rebuilt. Specifically, it is replaced by `/etc/conf/cf.d/init.base` appended with any files in the `/etc/conf/init.d` directory.

As another famous columnist might say

Winding down... Obviously there's much more that's new in UNIX System V/386 Release 3.2. I just haven't got round to looking at it yet. However, I have installed AT&T XWIN, based on X.11 Release 2 and a demonstration copy of OPEN LOOK. More on that in yet another future column.

UNIX: The Minimal Manual—Book Review

UNIX: The minimal manual, Jim Moore, Computer Science Press, Inc, 1989, ISBN 0-7167-8195-6. (UK) Price £14.95, Soft Back, 238 Pages pp, Size 23 cm x 15 cm. Reviewed by Susan D. Jackson of Xi Software Ltd.

The minimal manual is an introduction to the basic commands and features of UNIX for anyone unfamiliar with computers. It describes how to make files and directories and use the basic features of electronic mail, how to create documents using the editing, formatting and printing facilities, and how to format bibliographies automatically. The minimal manual tries to do this in a way that is easy to understand.

On the whole the book succeeds in being easy to understand, and gives a reasonable introduction to UNIX. It makes a good reference guide to basic features. The chapters on editing and formatting are easy to read and introduce the beginner to the essentials.

There is, however, a tendency to oversimplify, leading to some inaccurate statements. The introduction states that the commands covered are found on almost all versions of UNIX and that the author will note where this is not the case. The commands used are, in fact, heavily orientated towards the Berkeley version of UNIX and commands specific to this version are not always noted as such.

There is very little information about the use of terminals, and how to correct mistakes made while typing in commands. The assumption that CTRL C will stop any command could be very confusing to a beginner, as is the statement that if CTRL S is used to stop scrolling on the screen, pressing the space bar will start it again, as they do not work on every system. No alternatives to these commands are given nor is there any indication that there may be differences amongst systems. There is no mention of the shell or the different versions, only that you may have a version of UNIX with the 'Berkeley enhancements' if your prompt is a %. This of course only indicates that the user's shell is the 'C'-shell, and is no indication of the presence or absence of any other Berkeley enhancements.

The chapter on editing was very clear and easy to read, apart from one mistake where the command given for 'ed' to write at the beginning of the file was wrong. I can imagine the confusion that might result from this mistake. The chapter makes no attempt to explain the concept of the 'file pointer' around which it is built. This will make it very difficult for a user to understand the behaviour of the editor if they make a mistake.

The descriptions of the formatting macros in chapter 6 were good, but spoilt by an inadequate explanation of formatting and the different formatting programs and macro packages, and the lack of examples. As the book is aimed at the complete beginner examples are essential.

Who should buy the book, i.e., individual, lecturer, beginner... This book could be useful to a complete beginner wanting an easy introduction to UNIX. Someone using a standard Berkeley-derived system should have few problems following this book, but a user with a 'vanilla' System-V implementation would soon be lost in confusion with missing commands and system behaviour different from that described.

4. This seems to me a case of the tail wagging the dog.

Software | Review

Donal Daly
daly@cs.tcd.ie

Distributed Systems Group
Department of Computer Science
Trinity College Dublin
Eire

Donal Daly works as a researcher for the Distributed Systems Group in Trinity. His current work is involved in developing UNIX on top of an object-oriented distributed operating system. Previously he had system management responsibilities for System V and Berkeley UNIX systems within Trinity. Donal is the chairperson of the Irish UUG.

Introduction

Welcome again to *Software | Review*. This column reviews software that appears on the net. This is software that is posted to the source news groups and is free. I welcome contributions from you. In fact I actively encourage it!! You can send contributions to me at the above email address. In the future, I will accept reviews of commercial software as well.

This month's column includes a review by me on a very useful utility called *thack*. It was just the utility I needed, when I had to produce some troff quality output. It is this sort of utility I would like you to send me reviews on. This is followed by a review on UK-Sendmail, which is a sendmail configuration kit written as a collection of shell scripts and awk programs. I dunno whether the authors of this package deserve a medal for bravery, or need to be locked up!!

This is followed by a short report on GNU, what it is, and what is currently available. This is followed by a review of the GNU C compiler, *gcc*. We round off this month column with a update on European archive sites.

If you are sitting comfortably now, we shall begin.....

Thack

Thack is a filter which converts troff output into PostScript. It was written by Gareth Waddell and posted by Phill Everson (*everson@cs.bris.ac.uk*) to *comp.sources.misc* (volume 7). Two patches have also been posted to the same newsgroup recently.

A while back I needed to print out some documentation written using the ms macros package. Normally I use LaTeX, which we can generate high quality output using our laser printer. Plain old nroff output was just not good enough, and this documentation contained some tbl commands which when processed came out a mess on our printers.

At the time thack appeared in comp.sources.misc, I thought, this is a utility I might need some day, so I saved a copy of it. Well it was just what I needed. The distribution contains a makefile and everything compiled OK. To use it, I just passed the output from troff into thack and the troff

output was translated into PostScript. This PostScript was then transferred down to an Apple Macintosh for printing (using a Mac utility called SendPS). The resultant output was of very good quality. Output from eqn and tbl is not a 100% compatible but it's pretty good.

This is a good utility, in the great the UNIX tradition of doing one thing and doing it well, and also of inter-working cleanly with other utilities. It has now achieved the 'dubious' honour of residing in my *~/src* directory.

UK-Sendmail

Jim Reid <jim@cs.strath.ac.uk>

UK-Sendmail is a sendmail configuration file compiler kit. It is actually a collection of shell scripts and awk programs that generate sendmail configuration files that really work! Initial work on the package was done by Jim Crammond of Imperial College London (*jac@doc.ic.ac.uk*) which led to version 1.5. Further enhancements were made, notably by Jem Taylor of Glasgow University (*jem@cs.gla.ac.uk*). These were combined to produce the definitive version: 2.1.

The package produces sendmail.cf files that can deal with most (all?) common mail configurations and addressing styles. The package has support for SMTP (over ethernet as well as the Internet), uucp, JANET, DECnet, CSnet (using pmdf), Xerox mail over ethernet as well as local mail. In addition, it can configure sendmail to pass mail into the network news system. It can also make sendmail hide local hostnames, i.e., to the outside world, all your mail appears to come from one domain. It is generally a good idea to do this so that only your site need be aware of your local mail topography. It makes life simpler for everyone.

The package is easy to use. All that is required is for the user to edit a few template files to tell the package about the local mail topography—your domain name, lists of hosts on your ethernet, local hosts with particular network links (e.g., DECnet or UUCP), UUCP site name and so on. Once these are set up, the user simply runs a shell script and *voila!* a working configuration file is created.

UK-sendmail can make sendmail perform domain-based addressing, fully qualifying partially specified domain names. As an option, it can also swap domain ordering around to deal with big-endian and little-endian addresses. This is mainly a problem for the people on JANET who use different domain ordering from the rest of the world. It is also possible to arrange that hosts that are for mail purposes identical share one configuration file. This is particularly handy for diskless workstations that might otherwise use symbolic links to point at private file space.

A number of support programs are also provided. These include a simple authorisation program to control access to particular mail channels by certain users or hosts. Another program provides a mail interface to the netnews 'inews' program. A facility to support distribution lists is also included.

The configuration files produced by UK-sendmail work in a different way from the ones that most vendors supply. The major difference is that the configuration files generated by UK-sendmail do not perform routing based on address syntax. It converts all mail addresses into a canonical form and then decides how to deal with the resulting canonical address. Before invoking the appropriate sendmail mailer, the configuration file then rewrites the address into the appropriate syntax for that mailer.

For instance, suppose sendmail is given an address like *foo!user* when domain foo is actually an ethernet host. If sendmail routed on the address syntax, this address would imply a uucp mail transfer. The mail would probably fail because domain foo could not be found by the uucp system. By first converting to a canonical form, sendmail would then look up domain foo, realise that an ethernet mail transfer by SMTP is needed, convert the address to RFC822 syntax (i.e., *user@foo*) and then perform the SMTP transfer. It is amazing to invent bizarre mail addresses and see how UK-sendmail makes sendmail 'do the right thing' every time. Jim and Jem deserve special credit for providing something that is extremely useful and takes the pain out of configuring sendmail. Most sites who have used UK-sendmail find that they never have any more mail configuration problems once the files it creates are installed. What more could a postmaster want?

The package comes with plenty of documentation—an Overview, a User Guide/Installation Manual and there are man pages for the support programs. Annotated example template files are provided for the most usual site configurations, taken directly from the

files used to generate the configuration files used at Glasgow University.

UK-sendmail has been around for some years now. The latest release has been on widespread test for several months in the UK. This was so that any remaining bugs could be found and fixed before the package was posted all over the world. The package has proved stable and reliable and, more importantly, so have the configuration files it created. A posting to comp.sources.unix and/or comp.mail.sendmail is likely. Meantime, the package is available from the Glasgow info-server (*info-server@cs.gla.ac.uk*), although this may reject requests from beyond the UK. Plans are in hand to provide copies of UK-sendmail at the national EUnet backbones.

GNU

GNU (GNU's Not UNIX) is designed to be a complete integrated system, upwardingly compatible with UNIX. Large portions of this software are available and being distributed. GNU software is not in the public domain, but is distributed under agreement known as 'copyleft'. The purpose of 'copyleft' is to insure that everybody is free to copy a program as long as the person getting the copy is free to distribute his copy further, or to modify the copy that they have received. GNU software is produced by the Free Software Foundation (FSF). The aim of the FSF is not just to produce GNU, but is dedicated to eliminating restrictions on copying, redistribution, understanding and modifications of computer programs. I will now briefly describe some of the software which is part of GNU and is available from the FSF.

Program Development

There is a C compiler called 'gcc' and it supports the 1988 draft of the ANSI C standard (it is reviewed later in this column). This compiler is now fairly reliable and NeXT build their entire system, including their port of the Mach kernel and NFS, with gcc. There is a source-level debugger called 'gdb'. Gdb supports command completion, command line editing and history substitution. Gdb also supports debugging C++ code and FORTRAN code. You can also pretty print data structures. There is also a version of gdb which can run stand-alone, so it can be used for kernel debugging, and a version which supports a serial line interface for running gdb remotely. A version which will work over UDP is being worked on.

There is a C++ compiler (g++), which is written as an extension to gcc. This is the first UNIX compiler to compile C++ directly instead of preprocessing it into C. This has benefits for debugging and efficiency. There is also an assembler (gas) and a library (libg++) with utility classes for C++. There is also a C library with a nearly complete set of ANSI C library functions. There is also a version of make called 'GNU make' (what else :-)). This version of make supports parallelism. There is a rewrite of yacc called bison.

Document Preparation

The infamous EMACS editor is also a product of the FSF. This editor must have every option or feature you would require from a editor. Then there is 'ghostscript', which provides nearly all the facilities of a PostScript interpreter. Ghostscript supports X version 11. The FSF provides extensive documentation for the software it supplies. This documentation is distributed as 'texinfo' source files. These would be processed by TeX to produce a printed manual. My EMACS manual when printed is at least an inch thick. GNU documentation can also be converted into 'info' files for browsing in emacs.

Other Software

They have a variety of other software including a re-implementation of sh called 'bash' (Bourne Again SHell). It has korn shell features as well as command completion, job control and history substitution. They have other miscellaneous utilities like ld, nm, size, strip. They claim to have the world's fastest grep/egrep and the world's fastest diff. A new fast sort has also been completed. A fast lex called 'flex' has also been released.

How do I get GNU?

There are a variety of ways to get GNU software. One way is order the latest EUUG software tape (from the Brussels conference). This tape is full of GNU software. Contact Frank Kuiper (*frankk@cwi.nl*) for more information on this. Contact your local anonymous ftp site (if your lucky enough to have one!) they are sure to hold GNU software. If you have access to the Internet (an even luckier person!!) the latest version of GNU software is kept on *prep.ai.mit.edu* for anonymous ftp. You can also send to the FSF and

ask them to send you a tape. There is a charge for this ($150-$175) but I think it is worth it, and anyway you will be supporting the FSF. For more information you should email Leonard H Tower Jr (*tower@ai.mit.edu*).

Once you have obtained some GNU software, and become hooked on it as you realise that it is good quality software, be prepared to be applying patches, as this is software that is being actively changed. These changes occur as bugs are fixed or improvements are made. It can be quite time consuming working out what the latest version is, obtaining the patches, applying them, and ensuring that everything still works as it should under your environment. Remember though, it is free after all.

GNU C Compiler (GCC)

Eamonn McManus <emcmanus@cs.tcd.ie>

The GNU C compiler (gcc) is one of the most popular and useful of the GNU products. Written mostly by Richard Stallman, its purpose is to provide a compiler that runs quickly and produces small and fast code.

Gcc recognises the C language defined in the proposed ANSI standard; in fact, it was one of the first ANSI-compatible compilers available. This means that useful new features such as function prototypes and string constant concatenation are available. Prototypes, in particular, make C programming much less error-prone.

Gcc also provides a number of its own extensions to the ANSI C language. Several of these are potentially very useful, for instance a 'typeof' operation, inline functions, and the ability to use a compound statement within an expression. However programmers interested in portability will have to avoid them or use many #ifdefs, so their worth is questionable.

The compiler is written in such a way as to be portable to different machines as easily as possible. (Porting it is easy compared to other compilers; it still isn't easy, though.) The compiler 'front-end' generates code in a Register Transfer Language (RTL) that is largely machine-independent. A machine description written in a LISP-like syntax is used to generate the 'back-end', which recognises patterns in the RTL and converts them into the native assembly language. This technique is very powerful, and makes it possible to cobble together a compiler for a new machine quite quickly. Putting together a decent machine description is considerably harder, though.

The architectures to which gcc has been ported include VAX, 68000, 32000, i386 (80386), 88000, MIPS, and SPARC. This list, which includes machines at the extremes of RISC and CISC, illustrates the flexibility of the compiler setup. Basically, any reasonably orthogonal 32-bit machine could reasonably be a target for gcc. Steve Jobs's 68030-based NeXT machine uses gcc as its only compiler.

Gcc is a pretty good optimising compiler. It applies many well-known compiler techniques, such as common subexpression detection, data-flow analysis, register combination, loop invariants, strength reduction, etc. It also uses huge amounts of memory when optimising!

Recently John Gilmore has been porting BSD UNIX through the gcc compiler. The aim of merging gcc and the Berkeley distribution is to provide ANSI C compatibility, better optimisation and improved compiler maintenance. A successful merge is seen as an important test case for gcc. The current status of this work is that the Berkeley kernel has not yet been ported to gcc, but it has been syntax checked. This of course has involved various code rewriting in the kernel. The results of this work will be made available to recipients of Berkeley's next software distribution, whenever that is.

The only other UNIX compiler which is as widely available is Johnson's Portable C Compiler (pcc), which is the basis for most compilers on commercial systems. Comparing it with gcc is a bit unfair, since pcc is considerably older. But briefly, the differences are that pcc is smaller (about a quarter of the size), and uses less memory when compiling; whereas gcc is faster when not optimising, somewhat slower when optimising, and produces much better optimised code. And of course gcc is free.

In summary, gcc is an excellent compiler that is available for a wide range of machines, and it is completely free. Long live the FSF!

Archive Sites

I have received some more information about archive sites in Europe, and the details of archive sites in the Netherlands, Sweden and Spain are presented below. At present our hardworking

backbone managers are setting up a European archive service that will be available from each national backbone. I will give you more details when they are finalised.

If you have a large collection of useful software, or a number of large software packages like X windows, or GNU software and are prepared to make up tapes for fellow EUUG members, do get in contact with me. I will publish any offers in the next issue.

Netherlands

In the Netherlands, there is a small archive located on hp4nl.nluug.nl. It is intended for use by Dutch sites only. You can access this archive by UUCP for sites connected to hp4nl (almost all Dutch EUnet sites). There is FTP access for sites connected to the Amsterdam local area network. Looking through its index, it contains a wealth of useful software including isode, X11r3, elm2, TeX, rcs, nntp, bind, ka9q and sendmail. For more information contact *dfk@mcvax.uucp*.

Sweden

In Sweden, there is software archive accessible via UUCP or by anonymous FTP. A mail service is planned in the future. It is located at the Swedish backbone: sunic.sunet.se. The software archive contains some 150MB of software including all the EUUG tapes, archives of *eunet.sources*, *comp.sources{misc,games,unix}*. Also some of the popular packages like news, rn, nntp, sendmail and all RFCs. For more information about this archive service you should contact *ber@sunic.eunet.se*.

Spain

In Spain, there is an info server located at dit.upm.es. At present it is still experimental, but widely used within Spain. It stores RFCs, the latest Berkeley networking software, uupc, sendmail, elm, rcs, des, Cnews amongst other things. For more information about this server you should email to *pepe@dit.upm.es*.

Next Issue

I hope you have enjoyed this issue's offerings. As ever, I welcome your reviews of public domain software which you found useful, like how I found thack useful, in a time of need. In the next issue I hope to review the ISIS toolkit and distributed programming environment.

The AT&T OPEN LOOK ™ Toolkit

Betty Dall
ejd@attunix.uucp

Steve Humphrey
merlyn@attunix.uucp

Marcel Meth
meth@attunix.uucp

AT&T UNIX Software Operation

Ms Betty Dall—Developer
Mr Steve Humphrey—Supervisor
Mr Marcel Meth—Developer

All three work in User Interface development in UNIX(R) System V development at AT&T Bell Laboratories in New Jersey.

Introduction

The OPEN LOOK™ toolkit developed by AT&T is based on the X Window System®. The Xt intrinsics, part of the X Window System distribution, provide the basic programming environment for the OPEN LOOK toolkit. The toolkit consists of a set of widgets (graphical objects) with convenience routines that make it easier to use the intrinsics and widgets. Others have often referred to this toolkit as 'Xt+' or simply 'OPEN LOOK.' The toolkit has been designed to let application programmers easily program to the OPEN LOOK Graphical User Interface (GUI). A useful toolkit not only has to provide a simple and logical Application Programmer Interface (API), but also should require minimal system resources. A well designed toolkit should adhere completely to the conventions provided by the intrinsics.

The design of the OPEN LOOK toolkit followed three fundamental principles. First, the toolkit has to be 100% compatible with the intrinsics as distributed by MIT. Second, the toolkit is designed to remove the application programmer from the intricate details related to the OPEN LOOK GUI.

This paper gives a summary of the components of the OPEN LOOK toolkit, concentrating on aspects which make it unique. Release 1.0 of the OPEN LOOK toolkit has been generally available as of the first quarter of 1989. Work is continuing on the toolkit with the goal of improving performance. Some of the goals of the performance improvements are discussed in a later section.

Parts of this paper have been excerpted from XNextEvent Volume 2 Number 1.

Widgets

Widgets are objects that are used to provide the functionality and semantics of a user interface. The routines for creating and manipulating widgets are collectively called the Xt intrinsics, and are part of the X Window System. The intrinsics monitor events related to user interactions, such as key presses and mouse motion, and dispatch the event to the appropriate widget. The widget acts on the event and may change the visual on the display. Widgets can then call application registered routines, *callbacks*, which handle the specific application semantics of the interaction. The OPEN LOOK widgets implement the OPEN LOOK interface on an X Window System base.

A simple example of clicking a mouse button in a labelled area will clarify the interaction between

the intrinsics and a widget. The intrinsics sit in a loop waiting for an event to occur. The button click event comes from the X server. The event is dispatched to the button widget because it registered interest in button events that happen inside its window. The button widget code changes the button visual to show its highlighted state, calls the application's callback, and then displays the button visual in its normal state.

The Toolkit Components

The list of widgets provided in the first release of the OPEN LOOK toolkit is shown below. The widget class names marked with an asterisk (*) in the list are close derivatives of the original Core Components.

- **Menu Related**

```
AbbrevStack    (abbreviated button stack)
ButtonStack    (menu with quick access to default choices)
Exclusives     (choose one of many)
Menu
Nonexclusives  (choose several of many)
```

- **Composites**

```
BulletinBoard*     (simple layout)
ControlArea        (row/column layout)
Form*              (geometric constraint layout)
ScrolledWindow*    (scroll any widget)
```

- **Text entry, control labeling**

```
Caption
StaticText*    (eneditable, multi-line)
Text*          (full scrolling, user editable, multi-line)
TextField      (left/right scrolling, editable, single line)
```

- **List manager**

```
ScrollingList  (linear/hierarchical, scrolling)
```

- **Simple controls**

```
CheckBox       (toggle)
OblongButton*   (command button)
RectButton*     (toggle)
Slider*
Scrollbar*
```

- **Dialog boxes**

```
Notice          (modal box, freeze application)
PopupWindow     (command entry, property settings)
```

No Changes to Intrinsics

As stated in the introduction, one of the primary design goals was not to change the intrinsics. The OPEN LOOK toolkit runs with the standard X11R2 intrinsics, and the next release of the OPEN LOOK toolkit will run with the X11R3 or X11R4 intrinsics. The result of achieving this goal is that the time a programmer invests in learning how to use the intrinsics pays off when developing an application using the OPEN LOOK toolkit. No special widget creation routines are needed, and no different naming conventions are required. The naming conventions given in the intrinsics documentation for widget classes are

strictly followed, so that interesting names like "OblongButt" show up.

Providing a Simple API for a Complex GUI

The OPEN LOOK GUI is a sophisticated graphical user interface. Details such as pushpins, menu modes and popup's behaviour are issues that most application programmers do not care to address. So, the OPEN LOOK API was designed to free the application writer from these details. The feedback from early customers has been positive. The next five sections illustrate the simplicity of the API with several examples.

Simplified Menu Subsystem

Conceptually, a menu is a set of choices tied together in one package. In pseudo code, this can be expressed as:

```
menu = CreateMenu();
for (each menu item)
    CreateItem(child of menu);
```

This is basically how the OPEN LOOK toolkit works. An application writer creates a **Menu** widget, then populates it with other widgets representing the **Menu** choices. It is that simple.

But how does this work? In the OPEN LOOK interface, every menu pops up on the press or click of the menu button. The location and behaviour of the pop-up requires that it behave largely independently of a window manager or any other windows in the application. Thus, every menu in an application built with the OPEN LOOK toolkit is a pop-up widget, a subclass of the **Shell** class of widgets. Creating a menu is done with the *XtCreatePopUpShell()* routine from the intrinsics. The **Menu** widget provides any menu management functions needed, and automatically creates the menu pane widget that will contain menu items. This step is necessary to avoid attaching multiple widgets as children of the menu-subshell. (This is a problem for X11R2 intrinsics, but no longer an issue in X11R3). The widget ID of the menu pane is available to the application as a resource of the menu widget. The complete C fragment for creating a menu of four colours is shown below.

```
1   Widget menu, pane;
2   Arg arg[1];
3   static char *name[4] = {"Red", "Blue", "Orange", "Green"};
4   int i;
5
6   menu = XtCreatePopupShell( "menu", menuShellWidgetClass, parent, arg, 0);
7
8   XtSetArg (arg[0], XtNmenuPane, &pane);
9   XtGetValues(popup, arg, XtNumber(arg));
10
11  for (i=0; i<4; i++) {
12    XtSetArg (arg[0], XtNlabel, name[i]);
13    XtCreateWidget (name[i], oblongButtonWidgetClass, pane, arg, 1);
14  }
```

The only difference between this code and the conceptual program outlined earlier is the need to indirectly get the menu pane widget ID. This ID is obtained by the *XtGetValues()* on line 9 of the code fragment.

To make it even easier to build a menu in an application, the same widgets that are used inside a menu can be used outside a menu. Command buttons, exclusive and nonexclusive choices can be put in menus exactly as they can be put in a control area outside a menu. Furthermore, the menu pane is a **ControlArea** widget manager, which is the same widget manager used outside menus. This reduces the complexity of having different kinds of buttons, controls, and widget managers depending on where they are used.

Simplified Dialog Boxes

The same technique used in the menu subsystem is employed in the dialog boxes. The OPEN LOOK GUI interface specifies three types of dialog boxes: command windows, property windows, and notices. They differ in their intended use and operation. The API provides the command window and property window through

the **PopupWindow** class, and the notice window is supported through the **Notice** class.

A command window is a pop-up window that is used to execute application commands or set parameters. Buttons allow the user to exit when done filling in the information or to back out of the operation. When the user operates a button, the command window usually pops down. The user can keep the window from popping down by operating a "pushpin" widget in the window header; the pushpin metaphor is that the window can be "pinned" to the screen to keep it from going away.

A property window is used for getting more persistent information, typically attributes of an object in the application. The property window has several controls of arbitrary types that represent the state or description of the application. Buttons on the bottom of the window allow the user to apply the new state or reset the controls to their original states. Applying the new state usually dismisses the property window, although, as with the command window, a pushpin widget allows the user to keep it around.

The notice is used for alerting the user that an action may have unintentional consequences, like quitting without saving, overwriting a file, or initiating a long running and uninterruptible transaction. It typically has a few buttons that give the user a choice of continuing or not. The notice goes away after the user has operated a button, so there is no pushpin.

Each of the dialog boxes is implemented as a pop-up shell, like the menu. Each automatically creates the "pane" that will contain the text fields, buttons, and other controls needed by the application. Furthermore, each dialog box detects when it should popdown, thus freeing the application writer from the details of the *look and feel*.

By detecting when the user has operated a control (typically a button), the dialog box can automatically remove itself from the screen. This allows the OPEN LOOK toolkit to hide the pushpin feature from the application; the application does not have to figure out the state of the pushpin to decide if a pop-up can be popped down. However, the dialog box sometimes needs verification that the user has filled in enough information, or that the user's information is correct, or that the user has operated the correct control. The dialog boxes provide a callback that allows the application to "OK" the popdown before it happens. This division of responsibility puts the *look and feel* concerns (is the pushpin in? is there an OK button that was depressed? did the user dismiss the window with a window manager function?) into the hands of the widget, and gives the application control of the semantics of the action.

Widgets are Device Independent: Displays

The OPEN LOOK GUI trademark guide specifies exact size and shape of the visual elements using engineering drawings. The networking supported by the X Window System allows clients built with the toolkit to run on any monitor of any resolution supported by an X server. These facts motivated a unique design that achieves device independence in the OPEN LOOK widgets.

The OPEN LOOK widgets are device independent. Two related methods are used to achieve this. First, those visual elements that can be algorithmically drawn are drawn using real-world coordinates. These are converted to pixel coordinates using simple conversion routines supplied with the toolkit. Second, those visual elements that are best drawn "by hand" are made into fonts or bitmaps. We prefer using fonts, even for the glyphs, because of the drawing speed and minimal memory requirements. The second method requires packaging enough fonts and bitmaps to cover all the device resolutions that the OPEN LOOK toolkit supports. The OPEN LOOK requirements cover resolutions from as low as 50 DPI (dots per inch) to just under 100 DPI, in aspect ratios of 1:1, 5:6, and 3:4. Obviously, the toolkit does not cover all possible displays; when running on an unexpected resolution it uses the fonts and pixmaps that fit best. Adding a new resolution to the product is simple; new fonts and bitmaps are just added to the package.

Widgets are Device Independent: Keyboards and Mice

Device independence is required for more than the display device. Servers have different keyboards and pointing devices. The OPEN LOOK GUI specifies that a user may alter the key and mouse bindings to tailor them to his or her own preferences. The users' preferences must apply across all applications running on the server.

It would be inconvenient to require each section of code that is involved with input to respond

variably to mouse button one, two, or three (or four, or five, or shift-button-one, control-button-one, etc.), depending on the user's preference. Instead, the OPEN LOOK toolkit provides a layer on top of the intrinsics' translation manager that *virtualises* the mouse buttons and keyboard. A typical translation table for a widget in a non-OPEN LOOK intrinsic-based toolkit will look like the following:

```
Button1<Enter>:    highlight ()
Button1<Leave>:    normal ()
```

This example means that when the user moves the mouse pointer into the widget with button one pressed, the widget procedure *highlight* is called. The *normal* procedure is called when the mouse pointer is moved out of the widget. The widget could use a more general translation table that accounts for all possible choices the user might want to use. But then it would be duplicating much of the work of the translation manager, deciding which mouse events are associated with "highlight" and "normal". Every other widget would have to do the same thing.

The OPEN LOOK toolkit provides a routine used by each widget to convert a device independent translation table into a standard translation table. The virtual translation table for the example given above would look like this:

```
SelectBtn<Enter>:    highlight ()
SelectBtn<Leave>:    normal ()
```

The basic idea behind translations is still the same; the same syntax is used. The change is simply to use virtual names (SelectBtn) instead of device specific names (Button1). The conversion routine called by each widget looks for the predefined functional names and replaces them with the specific names that represent the mouse buttons or keys the user has decided to use. Each widget deals with a consistent name (SelectBtn for all "selection" operations, MenuBtn for menu operations, etc.) When the user changes the mouse button assignments, the changes are handled in a single place within each client.

Help for the User

The OPEN LOOK interface specifies that a help dialog box, containing a short explanatory message, pops up whenever the user presses the "help" key. Help is region sensitive, that is, the help message displayed is related to the position of the mouse pointer when the help key was pressed. A magnifying glass in the margin of the help window shows the help context by capturing a small snapshot of the screen.

The application writer does not have to provide the help information for each part of the application window. The OPEN LOOK widgets provide "default" help on how to operate them. This frees the application writer from writing help for every widget on the screen.

The "default" help can be replaced with application specific help on a widget by widget basis, or on a widget class basis. Registering help text frees the application from run-time help requests; the toolkit acts on all help key events to display the appropriate text.

Help can also be registered for a text string. The intent here is to provide a simple hook that lets the user get help about a word or phrase within the help text itself, if he or she needs more detail. This same feature also allows every help message to be identified with a word or phrase, even those help messages assigned to a widget or window. This allows later reference to help text by word only, such as when identical help text is to be assigned for other widgets or windows.

Since static help registration does not solve all applications' needs for providing timely help to users, the help registration allows the text to be supplied at the time the user requests help. This is called an indirect help registration. For ultimate control over presentation of help, a callback can be registered instead of help text. This effectively passes the help key event to the application.

There are certain applications that may need control of help before the dialog box is popped up. For example, some character-based applications within an OPEN LOOK framework require letting all keyboard operations pass through to the character application. This typically happens inside a terminal emulator client (for example, xterm.) The xterm application, for instance, uses the callback to know when the help key has been pressed. Then, it simply generates the equivalent key sequence and passes it on like any other key sequence to the character-based application running within its window.

Performance

The main motivation for developing a toolkit is to encourage application writers to use toolkit

components freely. Ideally, the toolkit objects will require so little of the system's resources that it will become noise in the application's overall system resource usage. AT&T is currently working on improving the performance of the OPEN LOOK toolkit. Preliminary data shows that the performance release requires 50 to 75 per cent less resources than the first release of the toolkit.

A thorough investigation of the toolkit has shown two areas which promise to shrink demands for system resources significantly: memory usage and protocol request reductions.

Memory Usage Reductions

Memory usage is responsible for the most severe performance problems. Once the size of the working sets surpasses the available physical memory, the system begins to thrash. Three areas have shown the most promise: shrinking the static size of the toolkit library, reducing dynamic memory allocations by the widgets, and keeping the server small.

Static Size Reductions

Converting the intrinsics and toolkit libraries to shared libraries reduces the static size significantly. The OPEN LOOK environment, itself, typically runs three executables which link these libraries. Sharing the text of these libraries saves at least two copies of the text. Further reductions can be realised by increasing code reuse. Code reviews of the toolkit have found common functionality implemented in different widgets.

Dynamic Memory Reductions

Dynamic memory savings are especially important for applications which use many widget instances. We developed a tooled version of the intrinsics that printed out sizes of memory allocations exceeding a user specified threshold. By varying the threshold, we were able to identify the areas that used the most memory. Not surprisingly, we discovered high memory consumption for widgets that are subclassed off large superclasses. Furthermore, we found that constraint widgets were expensive to use, since they allocated relatively large amounts of memory for each child that was added to them. Finally, we saw that every byte saved in frequently used widgets was worth the effort. One technique used to save data space for frequently used widgets, was to group data and cache it in a list pointed to from the widget class. In the worst case, every widget is unique and the number of items in the list of data groups equals the number of widget instances. (This would use approximately the same amount of data space as storing the data in the widget instance record.) The best possible scenario is that all widget instances use the same data group, thus requiring only one data grouping. This is often the case for buttons.

Keeping the Server Process Small

The server process is kept small by reducing the number of dynamically allocated data structures. Using a tooled version of the server we were able to identify those structures that were frequently used or excessively large. Thus, our efforts focused on reducing the number of windows, GCs, and pixmaps. Converting frequently used widgets to *Windowless Geometric Objects* (WGOs, DEC calls them gadgets), saved a large number of windows. Utilizing Xt's caching scheme for GCs in an optimal manor reduced the number of GCs allocated. Commonly used pixmaps were moved to font characters. Thus the server provided caching for these pixmaps across all applications.

Protocol Reductions

In general protocol requests are expensive, because they involve interprocess communication. The general rule is to minimise the number of requests, especially those that generate replies. A specialised tool was developed to display protocol request and replies. Looking at the data showed obvious extraneous events and areas for optimisation.

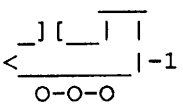
Conclusion

AT&T's toolkit provides a simple to use application programmer interface to the OPEN LOOK GUI. The OPEN LOOK toolkit is fully compatible with the MIT intrinsics and provides a host of new and innovative features.

EUUG Software Distribution

Frank Kuiper
euug-tapes@cwi.nl

Centre for Mathematics and Computer Science
Amsterdam



This article was supposed to appear in the spring issue of the EUUGN, but it failed to reach the editor before the critical time. Electronic transmission is still not entirely reliable. Oh well, let's hope it will work this time.[1]

As you read this, the Brussels conference will already be over, for those of you who were there, no more than a memory of rainy days, nice cars, and especially very good beer.
So, what news this time from the Software Distributions, you might ask? Well, quite some things actually. First the new software distributions. New are the following:

There is a new version of the ISODE distribution: version 5.0. This will replace the older version (4.0) on the EUUGD14 distribution. Those of you who ordered, but have not yet received this distribution, will automatically get the new version 5.0.

Second, those of you who missed announcements via other channels, like the EUnet News, might be excited to know that the latest release of the X Windowing system is now available as an EUUG Software Distribution. The X11R3 is available on EUUGD15. This distribution contains the entire release in compressed format. Uncompressed, this totals to some 87 Megabytes, so make some room on your (no doubt over crowded) disks before you start playing with this.

Thirdly, there is again a new conference tape: the EUUGD16, Brussels conference tape, containing all the GNU software that I could lay my hands on. See the distribution list for more details.
As the GNU material on other distributions has always been a major reason for people to purchase these distributions, I am certain that this distribution will be quite popular.
I will do my best to try and keep the latest versions of the various programs on this distribution, but can not really promise that. A lack of enough time is the main reason for this.

People who have been to the Portugal conference October last, will no doubt remember the talk about ET++ from people from the University of Zurich. I have been asked quite often lately whether this material will appear on a future EUUG distribution. For those I have good news. ET++ is available now as a new EUUG distribution: EUUGD17. I have recently received a tape from Zurich with the permission to redistribute the software for EUUG members.

Other news

As some of you may have noticed, lately I have not been quick with responses on requests. Without trying to excuse myself, I must say that there are some reasons why this is so. I hope you understand that my employer is very nice in letting me do things for the EUUG, but he prefers me keeping myself busy with 'more important' matters. This, a general lack of time and a growing number of requests for distributions, has led to a severer backlog of requests. I have been busy though, to rearrange the processing of distribution requests, and can now say that I think I have found a way to speed up the processing.

1. It had to be transmitted 3 times: it first failed due to a "." all on a line by itself, and then due to /usr/spool going full on some machine!

This should already have taken effect when you read this. We do need some time of course to clear the backlog. I still can not give any guarantee about delivery times, but in the future this surely should be a matter of weeks, and not of months (and I am keeping my fingers crossed behind my back ...).

One last remark. Sometimes people/companies send a cheque with their request. I urge you not to do this. With the current setup, you will receive an invoice with your requested software. Cheques are often difficult and expensive to cash. Thank you for your cooperation.

That's it for now. Below you'll find the list of currently available tapes and how to order them.
As always, anyone is invited to make their own tools, games, etc., available for publication on an EUUG tape. Please contact me for more details. Don't hesitate, just put the results of many nights of serious programming and hacking into the public domain, and you might even become famous!

This is a list of all the current (August 1989) EUUG software distributions. It is a short description of the available tapes. Any changes to the contents of the tapes, as well as announcements of new tapes will be placed in the EUUG Newsletter.
Prices of the tapes are in Dutch guilders (DFl), and do not include VAT-taxes. Prices include postage cost for surface mail within Europe. Any special shipment costs, like with DHL, will be billed through.
The first price listed is for reel-tapes in tar 1600 bpi format, the second one is for distributions on cartridge tapes in QIC-24 format. Prices for 800 bpi reel tapes and QIC-11 cartridges may differ from the ones listed.
Note that you have to be an EUUG member (or a member of a local UUG) to obtain tapes at list prices. Non-members will have to pay an extra DFl 300,- per tape.

EUUGD1 R6: UNIX V7 system, specially made for small DEC PDPs (11/23, 11/34, etc.). The Kernel supports the UK terminal driver. V7 source licence minimum.
Price: DFl 120,-/180,-

EUUGD2: Early Pascal compiler of the Free University of Amsterdam. V7 source licence minimum.
Price: DFl 120,-/180,-

EUUGD3 R3: Currently not available.

EUUGD4: Software tools, sampled by the Software Tools Users Group. Most of the software is written in Ratfor, for which a Fortran support tool is included. This tape is available in different formats: DEC RSX, DEC VMS, UNIVAC, IBM MVS, UNIX tar, MIT line feed format, and MIT card format (80 columns).
Price: DFl 150,-/180,-

EUUGD5: A collection of benchmark programs made up by EUUG.
Price: DFl 60,-/180,-

EUUGD6: (USENIX 83.1) USENIX tape, containing contributions from various UNIX System Group Members. This is a licence dependent distribution: V7, V32, SIII, V6 or no licence disclosure available.
Price: DFl 240,-/300,-

EUUGD7: UNIX|STAT Version 5.2. A collection of about 25 data manipulation and analysis programs written in C by Gery Perlman.
Price: DFl 60,-/180,-

EUUGD8: A collection of useful software, based on the so called Copenhagen tape (EUUG UNIX conference Autumn 1985).

EUUGD9: A collection of useful software, based on the so called Florence tape (EUUG UNIX conference Spring 1986). Price: DFl 150,-/210,-

EUUGD10: MMDFIIb. Multichanel Memo Distribution Facility (version IIb). This is a powerful, domain oriented mail system with access control and the ability to communicate over a variety of network systems including TCP/IP, JANET, UUCP, PHONENET, etc. It has been ported to a variety of UNIX's including but not limited to 4.[123]BSD, 2.9BSD, System III/V on a variety of different hardware. You should first obtain a licence agreement by sending a message to euug-tapes@mcvax. Return the signed licence with your order.
Price: DFl 90,-/180,-

EUUGD11: This is the 'Boat' tape; the Helsinki EUUG 1987 spring conference. It contains about 25 Megabytes of programs, games, etc. Including: jove, less, nag, news, rn , uEmacs, uuencode and lam.
Price: DFl 120,-/180,-

EUUGD12: This is the Dublin EUUG 1987 autumn conference tape. It contains about 26 Megabytes of programs, games, etc. Including: copytape, crc_plot, fastgrep, jove, kermit, notes, uupc, nethack, cron, sendmail, mh, Recipes, brl-gw, isode, pcip, pctelnet.
Price : DFl 120,-/180,-

EUUGD13: The latest conference tape for the London EUUG 1988 spring conference tape. It contains things like: cake, chat, config, copytape, graphedit, kermit, little-st, mcc, mstools, news, pd-diff, pdtar, perl, postscript, psfig, pshalf, shar, rpc, moria4.85, omega, arc, backup, smail, sush, watcher, and much, much more.

Price : DFl 120,-/180,-

EUUGD14: -> NOW A NEW VERSION <-
This is version 5.0 of this non-proprietary implementation of some of the OSI parallel protocols suites as defined by the International Organisation for Standardisation (ISO), the International Telegraph and Telephone Consultative Committee (CCITT), and the European Computer Manufacturer's Association (ECMA).

This release is coded entirely in C, and is known to run under the following operating system without kernel modifications:

BSD 4.2 and 4.3
Ultrix
AT&T UNIX SVR2 and SVR3
AIX
HP-UX
ROS
Pyramid OsX

Since a Berkeley UNIX system is the primary development platform for ISODE, the documentation and source are somewhat slanted towards that environment. The tape contains some 12Mb of both tools and documentation in machine readable form. EUUG will send you a tape only.
Price: DFl 120,-/150,-

If you want the complete documentation on paper (some 800 pages!) with the tape, you will have to order this distribution as follows:

Send a cheque or a purchase order for 200 Pounds Sterling to:

Department of Computer Science
Attn: Soren Sorensen
University College
Gower Street
London, WC1E 6BT
United Kingdom
Telephone: +44 1 387 7050, extension: 3680

Specify either 1600 bpi 1/2-inch reel tape, or sun 1/4-inch cartridge tape. The tape will be written with tar format and returned with a documentation set via DHL. Do not send tapes or envelopes. Documentation only is the same price.

EUUGD15: Here it is! The X11 Windowing system material, release 3: X11R3 This is one tape, containing the entire distribution from MIT in compressed format. Uncompressed this is some 87 Megabytes. This includes the core system, as well as much user contributed software.
Price : DFl 120,-/180,-

EUUGD16: This is the Brussels EUUG 1989 spring conference tape, and consists entirely of software from the GNU project from the Free Software Foundation (not to be confused with the OSF :-).
On this tape you will find: ispell, g++1.31, awk, gcc-1.33, gdb-3.1, Cscheme, emacs, lisp-manual, libg++1.32, binutils, bison, ghostscript, gas-dist, gawk2.02, gnews2.0, gnuchess, make3.27, oops-2.2, pace, ps-emacs, scheme, sed-1.01, tar-1.04 and torture.
Price : DFl 120,-/180,-

EUUGD17: This tape contains the software for ET++. >From the abstract of the "Autumn 1988 EUUG Conference Proceedings":
"ET++ is an object-oriented application framework implemented in C++ for a UNIX environment and conventional window system. The architecture of ET++ is based on MacAPP and integrates a rich collection of user interface building blocks as well as basic data structures to form a homogeneous and extensible system."
It totals some 18Mb of software that the people of the Institut fuer Informatik of the University of Zurich were so kind to let us, mere mortal souls, play with. Have fun.
Price: DFl 120,-/180,-

EUUG Software Distributions Order Form

If you want to order any tape, please write to:

EUUG Software Distributions
c/o Frank Kuiper
Centrum voor Wiskunde en Informatica
Kruislaan 413
1098 SJ Amsterdam
The Netherlands

For information only:

Tel: +31 20 5924121 (or: +31 20 5929333)
Facsimile: +31 20 5924199
Telex: 12571 mactr nl
Internet: euug-tapes

Please note that for distributions D1, D2 and D4 (and in some cases also for D8) a copy of your source licence agreement with AT&T for at least UNIX version 7 should be enclosed. Note also that you have to be an EUUG member (or a member of a national UUG) to obtain tapes at list prices. Non-members will have to pay HFl 300,- per tape extra as handling fee. Please enclose a copy of your membership or contribution payment form when ordering. Do not send any money or cheques, you will be invoiced.

All 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.

This page may be photocopied for use.

Name: .....

Address: .....

.....

.....

I would like to order the following:

.....

.....

.....

EUUG (or national UUG) membership form enclosed? Yes / No
Copy of AT&T source licence enclosed? Yes / No

"I declare to indemnify the European UNIX systems User Group for any liability concerning the rights to this software, and I accept that EUUG takes no responsibilities concerning the contents and proper function of the software."

Signature: .....

Date: .....

Puzzle Corner

Mick Farmer
mick@cs.bbk.ac.uk

Hi peeps,

Solution to Puzzle Number 4

Lack of space prevents me giving a full analysis of the answer. However, here's a summary. First, Bashful and Sleepy are not implicated in the house building exercise, so we assume they did not part. Second, the question refers to two groups the first containing Grumpy and the wearer of the blue scarf; the second containing Happy, the wearer of the white scarf, and another. From these and the other facts we deduce that the order of walking is:

Doc wearing the blue scarf carrying the saw (Grumpy).
Sneezy wearing the white scarf (Doc).
Dopey wearing the green scarf carrying the twine (Happy).
Happy wearing the red scarf carrying the hammer (Dopey).
Grumpy (Happy).

The names in parentheses are the names suggested so Happy, suggested by No. 3 and No. 5, was made to go back for the nails.

Puzzle Number 5

This puzzle has to be set in the past as today's digital clocks are too accurate!

On Friday, 1st April 1898 three new clocks were set going at the same time — twelve noon. At noon on the following day it was found that clock A had kept perfect time, clock B had gained exactly one minute, and clock C had lost exactly one minute. If all three clocks maintain the same rates of progress without stopping, when would all three pairs of hands again point at the same moment at twelve o'clock?

Puzzle Number 6

This is a fairly simple problem for those of you who understand the finer points of C and Pascal.

Write a program that will compile without errors in both C and Pascal. After the C compilation the program should output the message `"Hello Dennis!"`. After the Pascal compilation the program should output the message `"Hello Niklaus!"`. To find the solution you will need to consider how to invoke your C compiler!

Puzzle Number 7

This problem causes much hair pulling and wailing if tackled the wrong way.

Write a *self-replicating* C program, i.e. one that outputs an exact copy of itself. It should be self-contained, not reference any external files and, of course, be portable. The wrong approach is to think that you need a statement:

```
printf("int main(void)");
```

This means you also need a statement:

```
printf("printf\"int main(void)\");");
```

This obviously leads to difficulties. Needless to say, the solution revolves around the representation of string quotes!

Keep those solutions flooding in.

Mick

Recent Developments in Screen Fonts

William Roberts
liam@qmc-cs.uucp

Department of Computer Science
Queen Mary College
London, UK

William Roberts has been programming computers since age 11. After graduating from Oxford with an Honours degree in Mathematics, he worked for 2 years with microcomputers of various sorts, before returning to higher education. In 1985 he was awarded an MSc with Distinction by the Department of Computer Science at Queen Mary College, London and has remained there ever since. He is currently a systems programmer supporting the Departmental network of over 120 UNIX machines, and actively working on X11 and NeWS.

This is the third article in the series on Window Systems.

The Windows Column this month looks at the techniques for displaying text on bitmap displays, looking in particular at three recent developments by Adobe, Sun and Apple.

Introduction

This article is about fonts: the word 'font' is used very loosely by the computer industry, so I will try to be precise about what I mean. For the purpose of this article a *font* is a collection of character shapes, number shapes, punctuation shapes, etc., in a particular style. For example, Times-Roman and Times-Italic are both fonts. The character shapes in different fonts may be closely related, for example Times-Roman and Times-Bold, and a collection of related fonts is called a Font Family. Having defined my terms, consider the actual mechanisms used to make marks on paper or on the screen.

Traditional printing using moveable type involved assembling the correct sequence of pieces of metal, each marked with a character shape at a particular size. They would be in a particular font, but it is also necessary to know the size of the character in order to choose the right piece of type. Thus a printer would have large trays, each containing the pieces of type for a particular font at a particular size, e.g., Times-Roman 12 point. The size is given in a unit called *points* where 1 point is one 72nd of an inch: the '12 point' designation doesn't refer to individual characters but gives a general indication of the size with all of the character shapes scaled to fit. As a further complexity, the character shape may be changed at very small sizes to improve readability, which somewhat upsets the meaning of 'font' given above.

In more modern, photographic methods of printing, the letter shapes are made on film and can be enlarged or reduced photographically: in this case the film contains a single font such as Times-Roman and the size of the letters is controlled by adjusting the lenses used to focus the image of the letter shape onto photographic paper.

Bitmap Fonts

Most current systems with bitmap displays use *bitmap fonts*, the equivalent of moveable type. A bitmap font contains the character shapes as patterns of 1s and 0s, but only for a fixed size, e.g., 12 points, and then assuming a fixed size for

the pixels on the output device. Hence the Macintosh comes with bitmap fonts such as Times-Roman 12 point at 75 dots per inch. Bitmaps fonts are applicable to any pixel-based output device, including modern phototypesetters such as the Linotronic 300, but the amount of storage needed to hold all of the character shapes at all of the likely sizes is very large: the TEX Computer Modern Roman font for use at the 1270 dpi resolution of a Linotronic phototypesetter occupies 30 Kbytes just for the 12 point font.

The major virtue of bitmapped fonts is that they are very fast to use. Printing a character involves performing a rasterop using the character bitmap as a mask for the pen colour (which could itself be a bitmap pattern such a halftone). This kind of operation is fundamental to bitmapped display systems and tends to be highly tuned. It is also a very versatile technique because the mask can be used in different ways; as a cursor for example.

Bitmap font technology is still being advanced: on greylevel or colour screens where each pixel can display several shades of grey, the appearance of fonts can be improved by using *anti-aliasing*, essentially a way of using grey levels to fool the human eye into seeing details smaller than 1 pixel wide. The Macintosh can use antialiased fonts, and the NeXT machine from Steven Jobs uses a 2-bit graphics screen (i.e., black, white, 30% grey or 60% grey for each pixel) to achieve a marked improvement in readability.

The major limitation of bitmapped fonts is that they cannot be used satisfactorily at anything other than their intended size and orientation. Many systems will scale a bitmap, for example producing a 24 point font from a 12 point one by turning each pixel into a square of 4 pixels, but the results usually look jagged and unpleasant.

Outline Fonts

An *outline font* is the computer equivalent of the photographic font: it describes each character shape as a geometrically defined outline which can be drawn on the output device and 'coloured in' to produce the desired character shape at whatever size and orientation is required. In practice it is very common to use a letter shape more than once, so all outline font systems use a mechanism called a *font cache* which behaves in a similar fashion to the UNIX disk block cache. When a character is required at a particular size and orientation from a particular font, the font cache is searched to see it has been previously cached. If not, the outline font description is used to generate a bitmap for the character (a process known as *scan conversion*) and the result is saved in the font cache. The character is then printed from the font cache, just as though it was a character in a bitmap font. The next time that character is required, with luck it will be found in the cache and so avoid re-conversion. Font caches tend to be managed using a Least Recently Used algorithm, and with size limits to prevent a very large character from wantonly destroying a lot of small cached bitmaps. The scan conversion process is thus separated from the bitmap font mechanism and so an outline font can easily be added to an existing system with bitmap fonts.

The snag with outline fonts is that typical screens have very few dots per inch and so typical character sizes don't offer a lot of pixels to work with. A typical 10 point screen font has to construct a readable character shape in an area of about 10x10 pixels, so most screen fonts are 'hand tuned bitmaps' to make them readable. To overcome this problem, outline font character descriptions include *font hints* which are instructions about how to modify the character shape so that it fits well onto a coarse pixel grid.

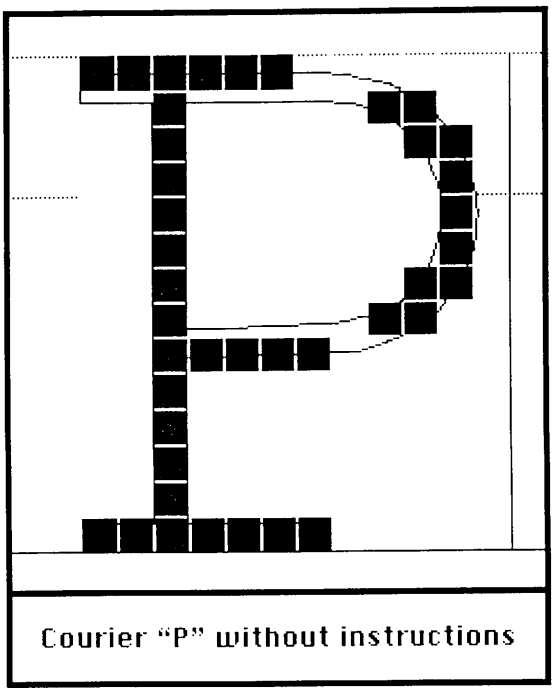


Figure 1

Figure 1 shows a naive scan conversion of a Courier font letter P at an awkward grid size, Figures 2-4 show some of the intermediate steps in adjusting the character, and Figure 5 shows the scan conversion of the hinted P.

(These figures are taken from a Hypercard Stack produced by Apple Computer Inc. to preview their Outline Font technology, described below.)

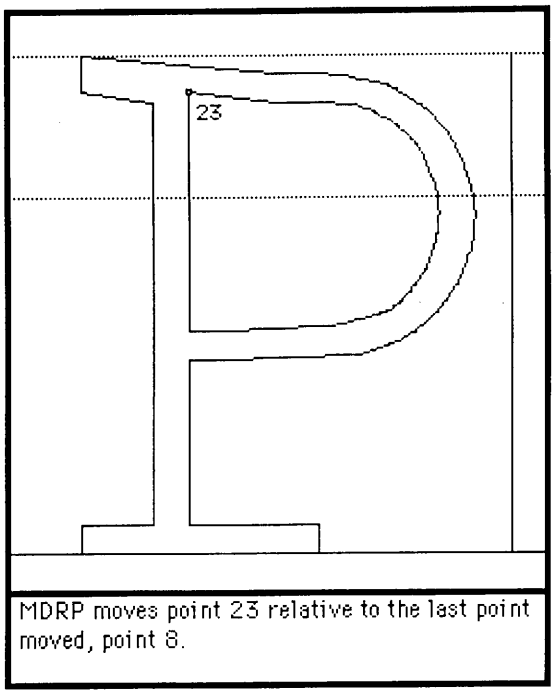


Figure 2

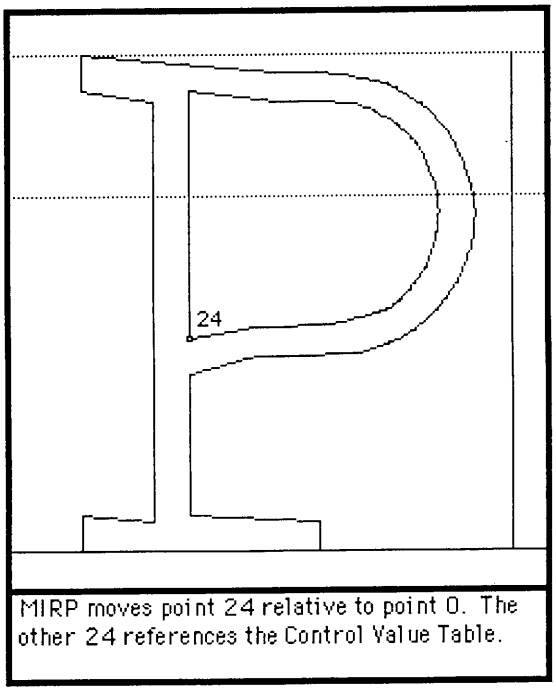


Figure 3

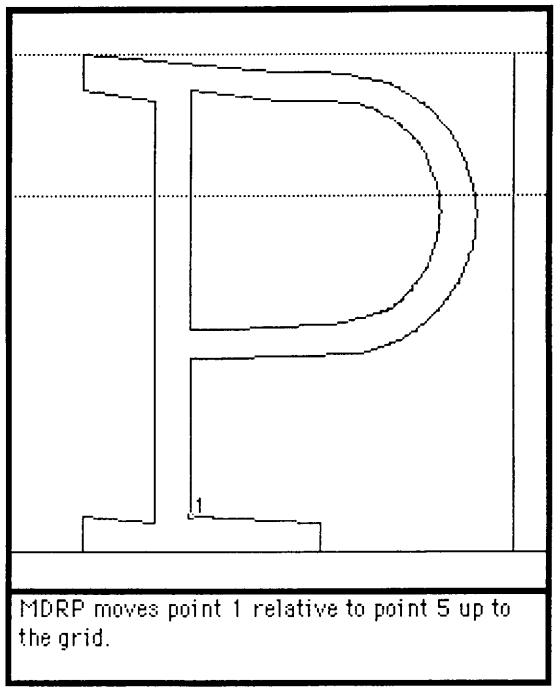


Figure 4

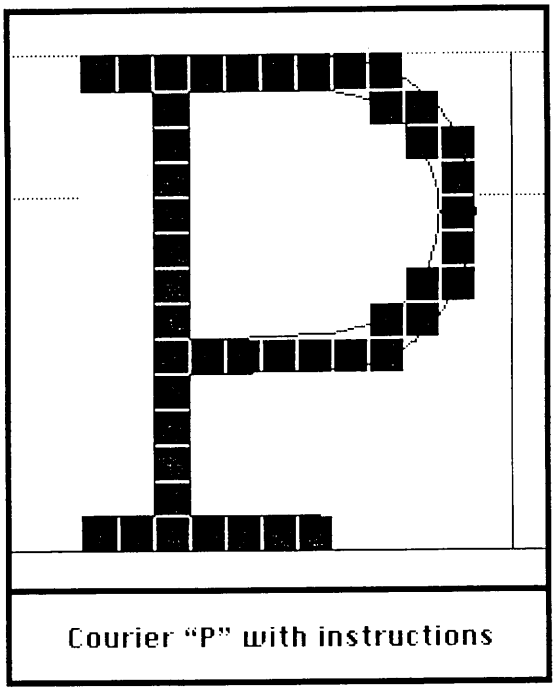


Figure 5

Part of the problem of hinting is to produce an efficient system which can give good looking results for a wide range of rotation angles and sizes, and which can be produced quickly and preferably automatically. For large numbers of dots, such as the 1270 or 2540 dots per inch on the Linotronic, outline fonts don't have this problem, but it is still significant with say 6 point text on a 300 dot per inch laser printer.

Display PostScript

Display PostScript is a product from Adobe Systems Inc. which essentially provides a PostScript interpreter for use with screen displays. The outline fonts used are the same as for a PostScript printer, which means that the screen is an accurate page proof for the printer output and that user defined fonts can be produced. The Adobe fonts are not written directly in PostScript however, but operate at a slightly lower level and include a proprietary hinting mechanism; this works well for small point sizes on a 300 dot per inch printer, and Adobe evidently feel that their mechanism is good even for small fonts at typical screen resolutions. The NeXT machine uses Display PostScript and it has also been licenced to DEC, though 'licencing' Display PostScript doesn't include access to Adobe's proprietary source code: instead, Adobe supply a core compiled for your machine, plus source code which demonstrates how to use this core in a window system.

Sun OpenFont

OpenFont is a fairly recent announcement from Sun Microsystems, consisting of three parts. The central part is a published font description language complete with hints called the F3 format and the part used in the display system to carry out the scan conversion of F3 format fonts is called the font engine. Finally there is a large expert system which accepts font outlines in a variety of formats and semi-automatically works out the necessary hints and produces the F3 format font description. In common with Sun's philosophy on such things as NFS, both the font engine and the expert system can be licenced as source code. The font engine will be part of the combined X/NeWS server currently under beta test.

Apple Outline Font

The Apple Outline font mechanism is unusual in that it uses quadratic curves rather than cubic curves, but otherwise behaves much like the others. Apple have published full details of the language used to describe character shapes, including a comprehensive list of routines for adjusting the character shape at small sizes; apparently this language has a number of operations put in there at the request of specific font manufacturers, and its performance is impressive. Apple have given away a Hypercard stack which demonstrates the hinting language in action, and includes a 'Screen recording' of the outline fonts in action.

Until the release of System 7.0 for the Macintosh, which will include the Outline Font mechanism as standard, Apple have had to rely on PostScript in the Apple LaserWriter to provide font scaling, using bitmap fonts on screen and with dot-matrix printers. Now that they have the outline fonts, the same font descriptions can be used to produce high-quality text for any pixel-based device, including cheaper laser printers, dot-matrix printers and even FAX modems. To make use of this, they have also announced a substantial revision of the way in which Macintosh printing is organised.

The outline font and printer changes would be significant by themselves, but there is a third component in the Apple announcements that will bring the two together and re-assert Apple's technical superiority in the desktop publishing and presentation markets. The *Line Layout Manager* transparently provides high quality typesetting to any Mac application at the same time as making it easier to program WYSIWYG applications. The difficulties with using an unusual point size such as 21 point is that character widths are typically not whole numbers of screen pixels. Any program which wants to do things properly has to do all of the sums associated with this and has to worry about subtleties such as kerning (where letters are moved relative to one another for aesthetic reasons) and ligatures (where letter shapes combine). These problems are even worse in non-European languages because the rules for kerning and ligatures are more complicated. The Line Layout Manager does all that; a program will simply ask for 'The office door is locked' to be displayed, and the line layout manager will deal with ligatures such as the 'offi' combination and so on. Applications can also ask which character the cursor is pointing to, ask for a range of characters to be highlighted and so on.

The Line Layout Manager uses information stored with the outline font to get closer to the original intentions of the type designer. For example, the Zapf-Chancery MediumItalic font available as a PostScript font is not a full version of Zapf's original design, but the Line Layout Manager will put in the missing subtleties: the 'e' of 'office door' should have an extending flourish on the horizontal cross-piece, and the Line Layout Manager will put that in as soon as it knows that the 'e' is followed by a space (think of it as an e-space ligature). All of this works in multiple fonts and multiple languages as well, though understanding the implications of the rules when trying to select a range of characters in a mixed Arabic-English-Chinese text is likely to be very difficult!

Conclusions

The inclusion of outline fonts in X/NeWS and the re-assertion of Apple's leadership in desktop publishing and presentation both make it very likely that we will all be using outline fonts in a few years time (except those of you still using KSR-33 teletypes...). The principal benefit will be that the screen display is just another approximation to desired image, in the same way that different PostScript printers just produce more or less accuracy in rendering of the desired image, so applications will not have to juggle two separate representations in search of the Holy Grail called WYSIWYG.

Design and Implementation of 4.3BSD—Book Review

The Design and Implementation of the 4.3BSD UNIX Operating System, Samuel J. Leffler, Marshall Kirk McKusick, Michael J. Karels, John S. Quarterman, Addison-Wesley Publishing Company, 1989, ISBN 0-201-06196-1. (UK) Price £31.95, Hard Back, 417 Pages pp.
Reviewed by Dan Debrunner of UniSoft Limited, *dan@root.co.uk*.

As the title suggests this book provides a deep insight into the UNIX operating system developed at the University of California at Berkeley. The concentration is on the 4.3BSD release running on the VAX, though this should not stop users of other architectures reading it, especially those running ported 4.3BSD code and facilities, the VAX is used more as an example, in such a way that the generic methodology is clear. With the exception of a chapter on the issues in system startup, shutdown and configuration, it concentrates on the UNIX system from the system call level downwards. Thus it is aimed at people who have some understanding of UNIX and UNIX programming. Anyone who is serious about using UNIX for programming (applications or systems) should be reading it to find the best way to utilise UNIX while others may like to peruse it just to find out exactly what is the *kernel* that everyone talks about.

The book is split into five parts, the first section provides a history of the evolution and design goals of BSD right up to the latest *4.3 BSD Tahoe* release followed by chapters on the the overall design of, and services provided by 4.3 BSD. Each of the following four parts is dedicated to a particular area of the system (Processes, I/O System, Interprocess Communication, and System Operation) in great detail. Although the level of detail is high, such as the structure of a memory management page table entry (`struct pte`), the reader is led to it logically and clearly from the overall concepts at the beginning. The text is aided by good diagrams and tables, including pseudocode and brief extracts from the source code.

Every chapter is well structured with each section following on from, and building on the last section. The reference section at the end of the chapter is preceded by the exercises, most of these are intended to be thought provoking rather than just picking answers out of the previous pages. To confidently discuss the issues raised and answer the questions requires full comprehension of the chapter rather than a quick skim.

The book is completed by a glossary with all the *buzzwords* that abound in conversations between UNIX wizards.

I found the book easy to read and very informative, a new concept is always explained before it is used, each section follows naturally from the previous one, and a diagram is usually referenced in the text at just the point where the reader thinks that one is needed. I would recommend it to anyone involved in the technical side of computing, and especially all users of UNIX systems, whatever variants.

With the imminent release of System V Release 4 this book does not lose its value to the computing community, but rather increases it, as it provides design and technical information about features now in V.4 that were taken directly from 4.3BSD. It should provide the ideal companion to M.J. Bach's *The Design of the UNIX Operating System* [System V] that you probably already have on your bookshelf.

EUUG

Glossary

There are approximately 8000 different words in this Newsletter. Here are the definitions of some of the not-so-common ones. Where a word has several meanings, the way that it is used in this issue is the one that is explained.

acclaim	acknowledge publicly the excellence of ...
accord	grant, give or bestow
acronym	EUUG is an acronym for European UNIX systems User Group
adequate	good enough
anecdote	story – to illustrate a point
ardent	expressive, passionate
browse	look though
burgeoning	growing, developing
chat	talk idly
cobble	put (together) in haphazard manner
coined	invent a new expression
convene	bring together
converge	come together, remove differences
depict	show
desire	want
elapse	pass – particularily used with time
embarked	set out
excerpts	extracts, bits of
forum	meeting place
glyphs	carvings – often on stone
haste	speed of action
homonym	words spelt the same but with different meaning
imprudent	unwise, not thought through
indemnify	protect/guarantee against ...
mutter	talk quietly or to oneself
omission	something missed out or forgotten
omnipotent	all powerful
panacea	cure for all ills
peruse	look though – browse
provoke	cause (thought or new idea)
snitch	an informer – to teacher at school
vague	not completely defined
viz	namely, used to list items. Short for videlicet

Abstracts

Here are the abstracts of the papers delivered at the EUUG Srping conference held in Brussels this year.

Copies of the proceedings are available from Owles Hall at £20 each including post and packing (there is an order form elsewhere in this newsletter).

Thanks are due to Stuart McRoberts <*sm@ic.doc.ac.uk*> who organised the typesetting.

UNIX and Load Balancing: a Survey

C. Jacqmot and E. Milgrom

Unité d'Informatique
Université Catholique de Louvain
Place Sainte-Barbe. 2
B-1348 Louvain-la-Neuve
Belgium
cj@lln-cs.uucp

W. Joosen and Y. Berbers

Departement Computerwetenschappen
Katholieke Universiteit te Leuven
Celestijnenlaan, 200 C
B-3030 Heverlee-Leuven
Belgium
wouter@kulcs.uucp

The paper presents a survey of load balancing schemes and their application or applicability to distributed UNIX systems. It aims to provide a guided tour through some important concepts and ideas in this area, together with examples of complete or partial implementations. Relevance to UNIX is stressed, even though some of the systems under consideration have little relation to UNIX proper.

Amoeba – High Performance Distributed Computing

Sape J. Mullender

Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam,
The Netherlands
sape@cwi.nl

The Amoeba Project is a distributed project on distributed operating systems. The project, which started in 1980 is now a joint project of CWI and Vrije Universiteit in Amsterdam and Cambridge University in the UK. About a dozen people are working on the project, led by Prof. Dr. A. S. Tanenbaum (VU), Prof. R. M. Needham (Cambridge) and the author (CWI).

Implementation of an Event Distribution Mechanism (EDM) on a Network of Workstations

Martin Zellweger

Institut für Informatik
University of Zurich and
ACU Systems AG, Switzerland

Peter Gloor, Rudolf Marty

Institut für Informatik
University of Zurich, Switzerland
gloor@ifi.unizh.ch
gloor@unizh.uucp

This paper describes an event handling mechanism for distributed systems. Processes, which are distributed over a network of nodes, can then communicate easily by means of events which may be sent over the network. The notion of a 'blocking event' serves as a means to synchronize the distributed processes. An implementation of the event distribution mechanism on a network of Sun workstations using Apollo's NCS is described. Finally some applications of the EDM are briefly mentioned.

Recent Changes in North American Computer Networks

John S. Quarterman
jsq@longway.tic.com

Texas Internet Consulting
701 Brazos Suite 500
Austin, TX 78701
U.S.A.
uunet!longway!jsq
+1 512 320 9031

This paper gives an overview of developments in computer networks in North America since the publication of 'Notable Computer Networks' (NCN) in October 1986 [Quarterman and Hoskins, 1986]. Much of the material was discovered during research for a book [Quarterman, 1989]. Although some of the figures are closely related to ones in the book, none of the text of this paper appears in the book.

The EDUNET Project

Manfred Wöhrl

Higher Level Secondary School
Spengergasse 20
Vienna
Austria

This paper describes the EDUNET Unix and IBM-PC network. The aim was to produce a modular network that allowed frequent changes in configuration.

Extending User Interface Toolkits for Picture Processing

Dr S. T. Jones
Mr William L. Franklin

Value Added Networks Lab
Department of Electronic Systems Engineering
University of Essex
Wivenhoe Park, Colchester
Essex CO4 3SQ
England

The name toolkit is a good description of both its purpose and what it provides. The toolkit builder provides the objects or widgets or tool-pieces to fit the types of interfaces that the toolkit builder imagines the toolkit will be used to build. Thus not all applications can be served by a given toolkit. As an analogy, a mechanic's, or electrician's toolkit provides the tools for their trades but could not reasonably be used to build a dining room table, at least not a table that would resemble that which could be made using a cabinet maker's toolkit.

User interface toolkits exist for most of the available windowing systems, e.g., X Window System, Andrew, etc. The contents of these toolkits do not always provide suitable tools for all applications. In the first part of this paper we examine the contents of the toolkits distributed with the X Window System for use in building a user interface to a Picture Editing system. In the second part of this paper we describe the development of tools for these operations and how they can be incorporated within the existing toolkits for use in future application.

The user interface requirements of a picture editing system are described, these are compared with the facilities provided by the main toolkit distributed with X/11 release 3. This provides the basic tools but does not include tools to support picture operations like region defines that require mouse input. The implementation of a track widget to provide this function for the X toolkit is described.

The Design of a UNIX Workstation Environment for Medical Image Processing

Dirk Tombeur
Frank Deconink

Experimental Medical Imaging Laboratory
Free University of Brussels
Laarbeeklaan 101,
B-1090 Brussels, Belgium.
dirk@primis.vub.ac.be
frank@primis.vub.ac.be

We are building a medical image processing environment, based on general purpose UNIX workstations. These workstations are integrated in a heterogeneous network, which allows resource sharing and communication. To promote modularity and extensibility, the environment is based on a hierarchy of modules. These building blocks are loosely modeled after Smalltalk-80 classes. At the application level an interactive image processing tool has been created, which is manipulated through a graphical user interface. C++ is used as an implementation language. To complement the object oriented constructs of C++, we implemented a garbage collector, for managing complex dynamic data structures.

Occursus Cum Novo – Realistic Movies rendered in an UNIX-Environment

Wolfgang Leister, Burkhard Neidecker,
Achim Stöer, Heinrich Müller

Department of Informatics
University of Karlsruhe
West German
leister@ira.uka.de

The goal of the project **Occursus Cum Novo** was to generate a complex photo-realistic animation of nontrivial length in reasonable time at reasonable cost. Photographic realism comprises complex geometric models as well as the implementation of several optical effects. Both can be achieved by simulation. Simulations guaranteeing high quality, as ray tracing does for rendering, are known to be very time consuming. The film has a length of 5 minutes and was done entirely on a network of about 30 UNIX work-stations of type SUN-3. The organising scheme is described. Processing is done automatically in such a manner that it does not interfere with interactive users. The results of the project are of general interest since they show a way leading to efficient high quality photo-realistic animation synthesis in the future.

Designing A Virtual Toolkit for Portability Between Window Systems

Marc J. Rochkind
Carol J. Meier

Advanced Programming Institute Ltd.
Boulder, CO
meier@boulder.colorado.edu

Today's application developers wishing to take advantage of modern graphical windowing technology are faced with the difficult decision of which platform(s) to develop their application for. While all of the popular window systems support certain basic functionality (windows, events, menus, mice, ...), the implementations vary widely.

The Extensible Virtual Toolkit (XVT) is a high-level interface that allows graphical, interactive applications to be easily ported to various window systems, such as X-11, MS-Windows, OS/2 Presentation Manager, and the Macintosh. Behind the common interface there is a separate implementation in the form of a C object library for each host system.

This paper describes the design principles behind XVT and its key programming features. It then reviews the main problems in creating an implementation for X and explains our short-term solutions. Also discussed are plans for more thorough long-term solutions using industry-standard toolkits.

Transport Interfaces in SINIX Systems

Norbert Richter

SIEMENS AG,
Munich,
Communication and Information Systems Group
mcvax!unidol!sinix!richter

Open systems and Open System Interconnection represent a major advance with mutual benefits for the software vendors, the system suppliers and the end user community. Modern design of open systems in information technology follows closely the Reference Model of Open Systems Interconnection as defined by the International Standards Organisation. In this model, the interface between layers 4 and 5 provides abstract transport service that is independent of the details of the underlying network implementation. Application programs access the transport service through an application programming interface. It is a most important aspect for writing portable higher-layer or end-user software intended to work across different networks as well as on various machines.

The X/Open transport interface XTI is defined to meet this requirement. It provides support for international and industry standards of transport services in UNIX systems. Using XTI, communication applications achieve the necessary

independence from the application environment and the underlying networks.

This paper discusses the XTI implementation in SIEMENS' SINIX system.

Implementing Internet Protocols on a small UNIX System

Danny Backx†,
Rudi Derkinderen,
Marc Snyers,
Pierre Verbaeten,

K. U. Leuven,
Dept. of Computer Science,
Leuven,
Belgium
mcvax!prlb2!sunbim!db

The authors have designed and implemented networking software implementing the TCP/IP protocol suite for a small system running XENIX. A library emulating BSD sockets is also provided, as the systems are integrated in a network of workstations and minicomputers.

The paper describes the most important design issues and their consequences. Although we chose for simplicity and a layered structure, measurements show an acceptable performance.

Striping Network Device Driver for TCP/IP

John A. Pew

GE Aerospace
NASA Ames Research Center
pew@orville.nas.nasa.gov

This document describes a project to design a UNIX network device driver that uses multiple paths across a network. The goal of this project is to increase the performance of the network between a Cray-II supercomputer and an Amdahl 5880 mainframe. Both systems run a version of UNIX with TCP/IP. Remaining interoperable with other hosts on the network is a chief concern; therefore, modifying the TCP/IP software is not a viable alternative. The network medium being used between these systems is the Network Systems Corporation (NSC) HYPERchannel.

The limiting factor in the speed of this network is the speed of the IBM-compatible channels on the Amdahl. The HYPERchannel is rated at 48 Mbits per second, however we are unable to take full advantage of this speed because the channel connections on the Amdahl are rated at only 1.5 Mbytes (12 Mbits) per second. This 1.5 Mbyte limit makes it impossible to use the full bandwidth of the HYPERchannel. To circumvent this limit, a striping driver that will simultaneously direct traffic over multiple paths to multiple HYPERchannel addresses is proposed. In order to accomplish this striping, one IP address must map to multiple HYPERchannel addresses. However, in order to adhere to IP standards, a one-to-one mapping of IP address to network interface is maintained.

† Currently at B.I.M., Kwikstraat 4, B-3078 Everberg, Belgium

More Haste, Less Speed

David S. H. Rosenthal

Sun Microsystems
2550 Garcia Ave,
Mountain View, CA 94043
dshr@sun.uucp

In the good old days, six years ago, I was teaching at the Universiteit van Amsterdam and sharing a 1MIPS, 4Meg machine with up to 60 other users. At times, it was irritatingly slow. Now, I have a 1.5MIPS, 4Meg machine all to myself, and at times it's irritatingly slow. Where did all the extra power go? Are we doomed always to be frustrated at the performance of our UNIX systems? What can we do to improve the performance of the systems we use and the code we write?

Authentication in a UNIX Network Using Smart Cards

Dr S. T. Jones†,
simon@ese.essex.ac.uk
Dr M. J. Clark,
markc@ese.essex.ac.uk

Department of Electronic Systems Engineering
University of Essex
Wivenhoe Park
Colchester
Essex
England
CO4 3SQ

The security of the UNIX login mechanism has been found to be satisfactory for individual systems. Implementations of remote logins for a network of UNIX systems have been less satisfactory. A system that improves the security for remote login has been developed based on the use of intelligent or smart cards. This paper presents an extension to the mechanism developed that allows the card to be used for authentication by application programs.

First the operation of the login authentication system is reviewed, then an analysis of the possible extensions and problems are discussed. Finally a detailed description of an experimental implementation of a card based authentication system is presented.

The system developed stores the credentials for a user within the smart card, these are accessed by an extension to the UNIX login mechanism, this is used for both local and remote access. Then a communications path is established between the card and the user's login shell. This shell verifies each command with the card and then executes the command which inherits the communications path to the card for its own use.

Into the Padded Cell

S. J. Hinde, D. Colman, A. McPherson, A. P. Standford.

Information Technology plc, UK

Information Technology plc produce secure UNIX products for the military and commercial markets. The secure development team reveal some of their experiences in securing UNIX

† British Telecom Research Laboratories, Martlesham Heath, Ipswich, Suffolk, England, IP5 7RE.

systems.

Following the principle that the highest fences of the protective measures should be placed at the outer defensive limits, the most obvious and cost effective method of reducing the threat to system security is to strengthen the UNIX login process.

Once users are accepted by the system, a strict 'need-to-know' principle is enforced. Users are only permitted access to information required for their work. Users are compartmentalised in a 'cell' of information. A user attempting to infringe the system security is isolated by the system into a 'padded cell', without privilege or access to useful information. Coupled with the requirements for enforcing a security policy is a need to monitor the action of the users – particularly as regards security sensitive events. An audit capability allows this form of surveillance to be achieved.

In standard UNIX, the system is managed by a single omnipotent superuser. Attacks using superuser privilege are potentially the biggest threat to the security of a UNIX system. This privilege can be sensibly partitioned between users of lesser privilege, greatly reducing the risk.

UNIX Standardisation: An Overview

Cornelia Boldyreff

Department of Computer Science
Brunel University
Uxbridge UB8 3PH
United Kingdom
corn@cs.brunel.ac.uk

A short history of UNIX standardisation is given here together with an assessment of the current state of affairs vis a vis UNIX standardisation. An outline of future developments of the UNIX related standards is given; and some general trends in standardisation are discussed with respect to the development of UNIX standards.

The Next Generation of UNIX System V

Bob Duncanson

AT&T UNIX Europe
+44 1 567-7711
ukcluellbob

This presentation describes some of the features of the forthcoming release of Unix System V.4.0. A brief review of the entire computer market first sets the scene for the objectives which lead to System V.4, this includes comments on standards, real-time support, internationalisation, and ABI.

Current Research by The Computer Systems Research Group of Berkeley

Marshall Kirk McKusick
Michael J Karels
Keith Sklower
Kevin Fall
Marc Teitelbaum
Keith Bostic

Computer Systems Research Group
571 Evans Hall
University of California, Berkeley
Berkeley, CA 94720
USA

kfall@okeeffe.berkeley.edu

The release of 4.3BSD in April of 1986 addressed many of the performance problems and unfinished interfaces present in 4.2BSD [Leffler84] [McKusick85]. The Computer Systems Research Group at Berkeley has now embarked on a new development phase to update other major components of the system, as well as to offer new functionality. There are five major ongoing projects. The first is to develop an OSI network protocol suite and to integrate existing ISO applications into Berkeley UNIX. The second is to develop and support an interface compliant with the P1003.1 POSIX standard recently approved by the IEEE. The third is to refine the TCP/IP networking to improve its performance and limit congestion on slow and/or lossy networks. The fourth is to provide a standard interface to file systems so that multiple local and remote file systems can be supported, much as multiple networking protocols are supported by 4.3BSD. The fifth is to evaluate alternate access control mechanisms and audit the existing security features of the system, particularly with respect to network services. Other areas of work include multi-architecture support, a general purpose kernel memory allocator, disk labels, and extensions to the 4.2BSD fast filesystem.

We are planning to finish implementation prototypes for each of the five main areas of work over the next year, and provide an informal test release sometime next year for interested developers. After incorporating feedback and refinements from the testers, they will appear in the next full Berkeley release, which is typically made about a year after the test release.

Guidelines for an Informatics Architecture

William de Backer

Commission of the European Communities
Jean Monnet Building – Room C2-84
L-2920 Luxembourg

As a user of the information technology (IT) for its own administration the Commission of the European Communities (CEC) has been – as it has to be – a forerunner and an example in applying a procurement policy based on standards. The CEC adopted this policy in 1980.

In order to share its experience with other customers, the IT industry and the standard-making bodies, the CEC is publishing this third edition of its guidelines for the implementation of a vendor-independent architecture.

These guidelines will be revised regularly to respond to changes in the market place. Since the first edition in February 1985, there has been considerable progress in the area of standardisation and a significant shift among major customers towards adopting standardised products.

This edition incorporates further developments to the architecture with a particular emphasis on simplicity, economy, timing and end-user services. Options left open for the future in the last edition have now been settled: inter-institutional cooperation (INSIS and CADDIA), ISDN, LAN, cabling, addressing, security, interactive communication, file transfer, applications architecture. Parts of the previous edition have been rewritten, without however changing the substance.

The next edition of these guidelines will concentrate on applications architecture. The Commission of the European Communities would be grateful to receive information and suggestions.

Creation of an Open Systems Market: The rôle of X/Open in the practical establishment of open system standards

John Totman

Director of European Programmes,
X/Open Company Limited.

This paper outlines X/Open's Rôle in providing an industry standard, it's current status and future directions.

The Open Software Foundation's Open Process and the End User

Henning Oldenburg

Open Software Foundation

The Open Software Foundation has passed a major test of credibility with its recent selection of user interface technologies, announced to OSF members December 30, 1988, and presented to industry, press and consultants January 11, 1989 in the US and January 13 in Frankfurt, West Germany.

The importance of the announcement of a User Environment Component, OSF/Motif, goes beyond the adoption of technologies. Of equal significance is the way in which OSF selected the technology, and the implications of OSF's 'open process' for the industry and its end users.

NFS refreshes the filesystem(s) A/UX cannot reach

Matthew Kempthorne-Ley-Edwards

William Roberts

Department of Computer Science
Queen Mary College
190 Mile End Road
London E1 4NS
UK
liam@cs.qmc.ac.uk

A/UX, the Apple Macintosh version of UNIX, coexists alongside the original MacOS operating system: separate partitions on the same disk with little real connection between the two. The potential for running MacOS applications under A/UX lead us to develop A/UX software which provides access to the MacOS files using standard UNIX file handling. This paper describes the A/UX environment and the techniques we used: a first version based on a library to simulate UNIX file system calls and the later version using the NFS protocol to mount the MacOS partition as a UNIX filestore. We offer some reflections on the problems and successes of our work and suggest a number of things which may help those providing NFS servers for other 'exotic' filestores.

A Contiguous High Performance File System

Shigetoshi Yokoyama, Shin-ichi Yamada,
Takehiko Nishiyama, and Kazuo Ito

Development Headquarters
NTT DATA COMMUNICATIONS SYSTEMS CORPORATION
2-11-6 Kita Saiwai, Nishi-ku, Yokohama-shi, Kanagawa 220, Japan
yoko@nttdpe.ntt.jp

Recently the UNIX operating system is ported to various machines, and in proportion to its expanse, UNIX's application area grows. One of its typical new applications is an operating system for multi-media workstations. In this application, UNIX must efficiently handle huge amounts of data, such as for images and video. According to the media properties, some media must be accessed in real-time.

In this paper, first, the problems and limitations of UNIX System V manipulating these huge data files are discussed. It is concluded that the best way to solve these problems is to support a contiguous file system in UNIX. Second, implementation of a contiguous high-performance file system needing no modification of file access semantics is described. Third, the following two key technologies to increase the performance of HPF are proposed.

- High speed directory search
- High speed free space search

And last, the contiguous free space search performance of this file system from the result of a simulation is shown.

Dp: a System for Inter-Program Communication

Robin Faichney

Computing Laboratory
The University
Canterbury, Kent CT2 7NF
England
rjf@ukc.ac.uk

Dp is a system designed to facilitate communication between application programs. Originally intended to prototype a partial solution to the problem of the interconnection of highly interactive software modules, dp constitutes a high level interface to the standard UNIX Inter-Process Communications (IPC) facilities. The means by which one application may address another using dp are much simpler and more flexible than standard IPC. Integers, character strings and user-defined data may be communicated, in addition to byte streams. Dp was designed for the graphical workstation environment, and is as network transparent as the windowing systems alongside which it is used. Experience in building applications with dp has lead to additions and modifications to its facilities. The design and implementation of applications in which communications play a substantial part, require considerably less effort using dp than would otherwise be the case.

Memory Management Hardware: Panacea or Pain?

Aarron Gull
Sunil K. Das

City University London
aarron@cs.city.ac.uk
sunil@cs.city.ac.uk

In recent years computer manufacturers have flooded the bottom end of the market with a diverse range of low budget hardware. Although the processing power of these machines has steadily increased, such workstations are typically devoid of the expensive memory management hardware often found on larger machines.

This paper investigates the difficulties of supporting UNIX on such machines. It draws examples from two separate re-implementations of the MINIX operating system both of which support efficient, if not secure, UNIX-like processes without the aid of memory management hardware.

Mk'ing Hardware: A Tutorial

Tom Killian

AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, New Jersey 07974, USA
+1 (201) 582-6215
tom@research.att.com

Suppose we wish to build a piece of hardware, using wire-wrap technology. At the end of the design process, we will have a set of 'object' files, most likely a list of point-to-point wiring instructions for an automatic or semi-automatic wiring machine, and one or more files for configuring programmed array logic (PAL's). These 'object' files will be generated from a plethora of 'source' files, graphic as well as textual; the details depend on what programs are available. Mk is the natural tool to tie these programs together, and ensure that the computer always has an accurate representation of the physical state of the board as it is debugged and modifications are made. In contrast to the primordial make, mk allows meta-rules to be defined with regular expressions, and transitive closure is part of its semantics. Using examples from the UNIX Circuit Design System (UCDS), we show how these properties can be applied to construct a master mk 'library' that allows rules for individual designs to be specified compactly. Such a library also facilitates tracking of changes in the design system.

A Model-Based Diagnostic System for the UNIX Operating System

Gail Anderson
Paul Chung
Robert Inder
Artificial Intelligence Applications Institute
University of Edinburgh
gail.anderson@ed.ac.uk

The use of model-based reasoning techniques allows for the development of systems which describe the underlying structure of problems better than rule-based systems can. A model of the UNIX Operating System based on the filesystem structure is developed. Two applications of the model are described. The model is used as the basis of a program to help the user run filesystem checks on UNIX machines. It is also used to perform diagnostics, in particular to diagnose problems which occur during the boot procedure.

SCOOP: a Software Environment for C++ Object-Oriented Programming

Ph. Fontaine,
Ch. Masson,
L. Stefanelli
Intecs International
Avenue Rogier, 385
B-1030 Brussels Belgium
+32 2 7355571

In this paper we describe a programming environment intended for those users of the C++ language who design programs according to a true object-oriented approach. First, we present the underlying principles of this approach which may be considered as a satisfying alternative to the conventional approaches. Next, we give a description of the main features of SCOOP, a tool developed by Intecs International. SCOOP allows the interactive design of the architecture and the coding of software built on top of the concepts of encapsulation and inheritance. It also provides automatic translation into Smalltalk and C++.

Step-by-step Transition from Dusty-deck FORTRAN to Object-Oriented Programming

Michel Bardiaux
Philippe Delhaise
Plant Genetic Systems
U.C.M.B Laboratories
U.L.B. CP160
Avenue Paul Heger
1050 Bruxelles
BELGIUM
mbardiaux@rubens.uucp

BRUGEL is a moderately large software for computer-assisted design of biological macromolecules. Its development began 10 years ago using standard FORTRAN-77 and continues to be based on that language. To overcome two major flaws of FORTRAN, viz. lack of expressive power and waste of the address space, we have developed a preprocessor and a dynamic memory management system featuring automatic memory reclamation, structured objects and error handling. The implementation of BRUGEL has evolved towards object-oriented programming by selecting from various programming languages which features were promising enough to warrant implementation as well as the recoding needed for their usage. Each such feature had to compete not only with other features but with the necessary priority accorded to implementation of new scientific methods in the software. This demonstrates, in a sense, the *inevitability* of object-oriented approaches.

Distributed Logic Programming

Andreas Krall
eva Kühn
Ulrich Neumerkel
Institut für Praktische Informatik
Technische Universität Wien
Argentinierstr. 8, A-1040 Wien, Austria
andi@vip.uucp, eva@vip.uucp, ulrich@vip.uucp

M-VIP (Multiuser Vienna Integrated Prolog) is currently under development at the Technical University of Vienna. M-VIP extends standard sequential Prolog by multi-user capabilities. It enables the development of shared knowledge base systems fully integrated in workable UNIX environments. In this paper we present the language extensions of Prolog to allow the concurrent execution of Prolog queries, initiated by different users and its implementations.

Performance Evaluation: The SSBA at AFUU

Christophe Binot
Hewlett-Packard
Philippe Dax
E.N.S.T
Nhuan Doduc
Framentec
ndoduc@framentec.fr

This is a progress report on the activities of the SSBA, the benchmarking group within the AFUU. The paper deals successively with the justifications and specifications, the alpha and beta testings, and the first customer shippings, of the

SSBA.

An Interactive UNIX Spelling Corrector

Philip M Sleat
Sunil K Das

City University London
phils@cs.city.ac.uk
sunil@cs.city.ac.uk

When designing interactive information processing applications there are two important properties of the data files that must be considered. The first of these is the size of the file. The second is speed of access. An investigation was conducted into the most appropriate structures for the storage of dictionaries. Sequential, binary and hashing techniques proved to be inferior to tree-based methods which permitted data compression.

Tree structures for storing a dictionary file were used in writing an interactive spelling corrector because:

- tree-based storage methods were the only method to introduce data compression;
- access times were found to be better in comparison with the sequential search and comparable with binary chop or hashing; and
- it was discovered that there existed inherent spelling correction within the tree.

Context-Reflecting Pictures of a Database

Agnes Hernadi
Zalan Bodo
Elod Knuth

Computer & Automation Institute
Hungarian Academy of Sciences
Budapest

An experimentally implemented unusual database interface with a new idea of context managing mechanism is introduced to make multi-contextual data dialogues intersession resident and readily responsive to appropriate changes of the database. Dialogue contexts, called pictures, reflect the real data, so their universe is a complete information model (to be mapped to the database scheme). Moreover, pictures are the only means by which database access is brought about. A picture can be regarded as an entry form for typical data input, or a transient portion of information on something, or a detailed report on something to be kept safe, or a part of the database scheme from a given point of view, or a distinctive query specification, etc. Operations on contexts are not associated with traditional database subfunctions (insert, delete, update etc.) but are all interpreted on pictures and provide all subfunctions in one pattern.

A Simple Guide to Porting the X Window System

Lee McLoughlin

Department of Computing.
Imperial College.
London.
UK.
SW7 2BZ
lmjm@doc.ic.ac.uk

The X Window System is the *de facto* graphics standard for UNIX workstations. The authors of the X Window System have gone to considerable lengths to allow it to be ported easily.

The server provided on the X distribution tape has some presumptions about the target system. The further from this model the harder it is to port X.

This paper only gives guidelines about how to port X by explaining where the major problems are and pointing to examples in the sources of code to act as a template for your own port. Only version 11 of X is discussed.

The author is a researcher in graphics interfaces. He has ported X release 10 to the Whitechapel MG-1 and HLH Orion and ported X release 11 to the HLH Orion 1/05 and has advised on other ports.

A.F.U.U News

Matthieu Caprile
Matt.Caprile@ec.bull.fr

UNIX™ Research Centre
Bull S.A.

Matt is a night-owl programmer at Bull S.A., and has been for the last 3 years. His domain of predilection is internationalisation, and is involved in standards work. He participates in /usr/group and X/Open internationalisation work, and tries to implement between airplane trips. His latest personal project is implementing a programme to turn water into wine, but although the technique has existed for almost 2000 years, he is still having problems. One major difficulty is in converting the old cubit measurements into cubic centimeters. It is a frustrating project due to the limited success obtained so far, but after each two-hour session of quality assurance, (which involves comparing the results with some of the better French *vignobles*), he doesn't really mind (until he awakes the next morning).

Introduction

This article is a general overview of the A.F.U.U. First the upcoming future events are simply listed, and the rest of the paper deals with a subject that the A.F.U.U. considers fundamental: services to its members, and specifically the workshops (also called study groups) that we have been organising lately.

Future Events

Upcoming events in France over the next few months include:

Journees UNIX de Grenoble

25-27 Oct. 1989 in Grenoble exposition hall "Alpes Congrès". This is the second edition of the UNIX conferences and exposition in Grenoble. It is being co-organised by the A.F.U.U. and the regional developement board. Last year it was held in conjunction with the CONVENTION A.F.U.U. 88, and it was a big success. This year it will be twice as good: twice as many tutorials, twice as much exposition space. Contact BIRP Com.Tec : +33 1 4354 7567

CONVENTION A.F.U.U. '89

24 November 1989 at the new CNIT Paris la Défense. This is the traditional, annual A.F.U.U. meeting of members, which votes the budget, elects replacements to the governing board, etc. There are also workshops, and hopefully there will be the brand-new A.F.U.U. tee-shirts (if we can talk the price down).

CONVENTION UNIX '90

27-30 March 1990 at CNIT Paris la Défense. This is the A.F.U.U.'s annual trade-show and technical conference. There will be displays by most of the major manufacturers and software houses of their most recent products. It is also being held in the CNIT. This marks a big change for us: the floor space is almost twice as big as last year, and there will be more conferences. Also, and certainly not to be forgotten, this will be the year that the conference becomes European: several non-French technical presentations will be made, and almost all European UNIX computer manufacturers will be represented. contact: A.F.U.U. +33 4670 9590.

OTHER FRENCH CONFERENCES

- TELECOMEXPO 89. 19-22 Sept, Paris. The 2nd exposition for telecommunication professionals. contact: +33 1 4665 1115.
- EIFANS '89. 21-2 Sept, Caen. International forum on applications of new superconductor materials. Organised by EC2. Contact: +33 1 4780 7000.
- PIXIM 89. 25-29 Paris. A gala for image technology. Organised by BIRP. Contact +33 1 4742 2021.
- Workshop/conference on data-bases. 26-28 Sept. Geneva Organised by the French National Research Institute for Automata and Information sciences (INRIA). Contact: +33 1 3963 5600.
- INFORA 26-30 Sept. Lyon (Eurexpo showroom). Large exposition for office automation, Desktop Publishing, teletex, and office applications. Contact: +33 7222 3344
- AppleExpo 27-30 Sept Paris. Contact: +33 1 6928 0139
- Productique 2-6 Oct., Paris. Exposition dedicated to Robots and Industrial Automata. Organised by Sepic. Contact: +33 1 4039 1515
- Forum MOS 3-4 Oct., Paris. Forum dedicated to the MOS Operating System. Organised by Infologiciel +33 8844 9020
- HYPERCUBES 4-6 Oct. Rennes (Brittany). Internation conference on hypercubes and distributed/parallel processing. Organised by INRIA research institute. Contact: +33 1 3963 5600
- MICRO 89 9-13 Oct, Paris The big bang of micro-computers. Organised jointly by Sicob, Infomart and Capric. +33 1 4261 5242

SERVICES PROVIDED BY THE A.F.U.U.

As stated in the introduction, the A.F.U.U. feels very strongly about providing services to its users. It is an *association* of users, and exists to serve its members as much as possible. The rapid growth of the association attests to both the increased interest in UNIX, and the quality of service that the users receive. I will give a rapid description of some of the services we provide, with special emphasis on the study groups that we have organised.

NEWSLETTER

This is the traditional service offered by most of the European UNIX groups. But the newsletter has come a long way since its debut: from the original few stapled sheets, to its present 60 page glossy magazine. The newsletter, TRIBUNIX, is printed 6 times a year. The users are very happy with the quality of typesetting and of the articles. Articles include technical presentations (from descriptions of distributed operating systems techniques, to database structures and new product reviews), review of conferences in France, Europe, and the U.S., book reviews, a programmer's column, job offers and machines for sale, and updates on what the working groups have been doing recently. Thanks are due to Philippe DAX the chief editor, and the entire publication committee, for the excellent accomplishments of the last year.

The A.F.U.U. is considering starting a once-a-month flyer (one page news flash) to better inform the members about the important things that have happened during the month.

TAPES

The A.F.U.U. have been shipping public domain tapes for its users for quite awhile now. The types of tapes include archives of specific news groups (such as comp.sources.unix), the S.S.B.A. benchmarks, the FNET tape (everything to get started on e-mail), X-window-system from MIT, etc. The most popular form of media still seems to be 1/2" magnetic tapes, although there are a few 1/4" QIC-24 tapes requested occasionally.

LIBRARY

The A.F.U.U. has been building up an extensive library over the past few years, now having well over 1000 books. The books cover over 15 subjects, from Operating Systems to networking, databases to graphics. There are also the 20 or so weekly and monthly magazines and trade journals you would expect to find in a library. And of course, there are back issues of TRIBUNIX and EUUG newsletter. The materials are available either at the AFUU offices, to read there or check out, or loaned to members via surface mail. The loans are for a duration of 20 days, after which a late fee is charged and/or library privileges

revoked, depending on how overdue items are. Anyone caught photocopying books immediately loses library privileges.

WORKING GROUPS

The last 6 months have marked a great change in the working groups sponsored by A.F.U.U.: many new ones have been created, and I am finding it difficult to keep track of them! The working groups are formed by A.F.U.U. members to research, study, or just provide a forum for discussion of various aspects of today's "UNIX culture". They usually meet once a month in the A.F.U.U. offices, and any A.F.U.U. member is free to participate. Each group decides individually what its objectives are, and the best way to meet those objectives. Some work groups publish papers or information in TRIBUNIX, the A.F.U.U. journal, others invite presentations from vendors and researchers in their topic of interest, and others embark on ambitious projects, such as the Benchmarking group, which I will introduce first.

BENCHMARKS

The benchmarking group set as its goal to produce a standard benchmarking suite available to all. It recognised the need for such a suite due to the many different "versions" of benchmarking programs, such as dhrystone or whetstone. Not to mention the fact that some vendors modify "slightly" the benchmark to get better results. (some definitions of "slightly" include rewriting the inner loops of a program in assembly language!)

The group has worked hard for the past two years, and has produced a tape of benchmark programs, most of them standard, generally available benchmarks, and including configuration and output scripts. This collection is called the S.S.B.A. (*Suite Synthetique des Benchmarks de l'A.F.U.U.*). The S.S.B.A. tape is loaded onto the machine to be tested, the configuration file is set up (which contains such things as the name and type of the machine, compile time options, machine configuration, etc.), and the tester types "make". No modification of the source is allowed.

The A.F.U.U. has started a programme of "verifying proper running of the suite" to make the results more useful and believable. The process is simple. Anyone writes to the A.F.U.U., asking for a member of the benchmarking group to come and verify the correct running of the test. A person will visit the site, and after completion of the suite will provide the requester with a paper that certifies the proper execution of the benchmark suite. The paper does **NOT** serve as a definitive measure of the performance of the machine, only that the benchmarks were run as required by the suite. This includes such things as not modifying the code, and executing the entire suite without stopping and re-starting later (thus it is not possible to run the different tests with different configurations of the machine). Those who work with benchmarks understand that a number, or series of numbers, is not an absolute measure of machine performance. The certification only exists as proof that the tests were not modified or "improved" so that all the results from different machines were run using the exact, identical programs. The suite has been requested by the E.E.C. for use in evaluating machines.

The S.S.B.A. was presented at the last E.U.U.G. conference in Brussels, and readers who wish for more information can find them in the conference proceedings.

WORKSTATIONS

This working group is one of the oldest, having been created two years ago, at about the same time as the benchmark working group. Its purpose in life is to study, and discuss, all the aspects of "workstations", among others:

- what is a workstation? Depending on the user, this could mean a PC-AT with a graphics card, or a 30-MIPS 3-D super-station.
- Architecture of workstations. The pros and cons, what architecture speeds up what types of work. RISC, CISC, specialised graphics processors, memory hierarchies, etc.
- Different Operating Systems. UNIX, MACH, Andres, Athena, and more. To what extent are distributed services are available, including fully distributed operating systems such as Chorus®. Co-existence with MS-DOS® and cooperation of different operating systems.

0. Chorus is a registered trademark of Chorus Systèmes.
0. MS-DOS is a registered trademark of Microsoft Corporation

- Compilers and Languages. Optimisation, development tools, CAI and CASE.
- Graphics capabilities, interfaces, and standards.
- Networking. Different protocols for information exchange, networking transparency, network administration.
- Desktop publishing and office automation using workstations.
- real-time. Different approaches and models.
- Provide a forum for exchange of information on workstation products.
- Benchmarking. A tool to help measure some aspects of workstation and graphics performance.

And of course, the goal is to make as much information to A.F.U.U. members as possible, to help chase away the clouds of misinformation surrounding workstations. Many different products have been presented to the group, and this information has been passed on the members: from new CPU architectures, to fully distributed services. The group has made documentation on new products available, and organised conferences on graphics, workstations, standardisation efforts and new technologies. Special presentations have been organised for particular subjects, such as RISC technologies, compiler optimisation, X windows, GKS and PHIGS, as well as round-table discussions with different workstation manufacturers.

SECURITY

This group has two goals, and is divided into two "subgroups" (which generally have the same members) that have different goals. This first group's goal is to study security and secure systems in general. The second group concentrates on security in "standard" UNIX systems, and what can be done to make them more secure. This is also a study group on UNIX administration, and how to do secure system administration. The group has published a security "cook-book" to help administrators guard against attack. Also, Some papers have recently been published in TRIBUNIX to inform the user community, such as X.400 (March/April 89), a glossary of security terms (March/April 89), and *uucp* (May/June 89). Two more articles are almost ready to be published. Papers and transcripts are available at the A.F.U.U. offices. There have been discussions on utilities pertaining to security that have recently been posted to netnews, such as: *setuid*, *access*, *pwdiff*, *tallow*, and *diskhog2*. Future meetings will discuss, among other things, the *crash*, *cmpress*, *dcopy*, *ff*, *ncheck*, and *audit* programs.

NETWORKING GROUP

This working group discusses the evolution of the French UNIX network, FNET, which is the French part of EUNET. The group has published several papers (a few of which where available at the last E.U.U.G. conference) on *uucp* and *sendmail* installation and configuration, hints on how to find and solve problems, etc. Other goals include providing a forum for discussion of FNET and its services to A.F.U.U. members, help solve any problems that might arise, chose technical solutions, provide education to the A.F.U.U. community about FNET, and chose new directions for FNET, such as a connection to the InterNet, or high speed dedicated lines, or gateway service between protocols and/or networks. This group has been instrumental in making the FNET service a reliable, fast service. The group also discusses, and invites presentations on, different aspects of networking. Examples are NCS from Apollo, the ISO/DE implementation, and new directions in networking (such as fibre optic connections).

USER INTERFACE

Considering the multitude of User interfaces available to the user[1], Open Desktop, Motif, OpenLook, DecWindows, PM/X NeXTstep, Macintosh toolbox, and MS-Windows, it is easy to get lost in the world of bitmaps and mouse traps. The group would like to inform users on the different user interfaces available today, and to the different types of possible implementation bases, such as X-11 or Display PostScript[2].

1. I think that most of these are most likely trademarks or registered trademarks. Open Desktop, by the Santa Cruz Operation; Motif, by OSF; OpenLook, by AT&T and Sun (alphabetical order); DecWindows, by Digital Equipment Corporation; PM/X, by Hewlett-Packard and Microsoft; NeXTstep, by NeXT; Macintosh, by Apple; and MS-Windows, by Microsoft.
2. PostScript and Display PostScript are trademarks of Adobe.

The first meeting of this group took place June 7th, and consisted of a presentation of X-11 and working out a plan of how to proceed and the goals of the group. Future meetings will hopefully address the subjects of graphical quality, choice of fonts available, character size type and usage of icons in the different systems, how the mouse is (or can be) used, etc.

SCIENTIFIC COMPUTING

This could also be called the super-computing workshop, but the definition of what exactly defines a supercomputer is almost a religious subject. The first meeting of this working group took place the 27th of February, at the Framentec facility, hosted by Nhuan Doduc. There were 29 participants in this first workshop, which promises to be an active group. The reason for the formation of the group is the explosion of UNIX in the scientific number-crunching world. The goals of the group are three-fold:

1. to allow the members of the group a forum for discussion on new technology in this area and for presentation of new and/or interesting configurations.
2. to be a forum for exchange of view-points, experiences, and hard- and soft-ware solutions.
3. to be a forum for expression of what is needed in the area of scientific computing.

Possible topics are vectorisation in C, rapid I/O methods, Fortan 8X, and scheduling and load balancing. There was a presentation made by Mr. Fred Metry, of Astronautics Corporation of America, who presented their ZS supercomputer. There was an interest expressed by the group in the area of parallel computing, and there will be thought to either creating a separate parallel processing work group, or to including this subject in the areas of study of this group.

MAITRISE CULTURE UNIX

This workshop was formed in the beginning of last year, and its goal was to help distribute information about "the UNIX culture". Everything that you ever wanted to know about UNIX, but were afraid to ask your local guru. This group has published a series of documents and tutorials aimed at helping new users get started, as well as addressing more advanced topics for the experienced user. The group has since "split" into several other groups as the need arrived, the newest being the portability sub-group.

PORTABILITY

This new study group concentrates on portability, as you probably suspected. It met for the first time March 14th in the A.F.U.U. offices, and met again on April 14th. The main objective of the first work shop was to get to know one another and to define the objectives to work toward. The group split the idea of portability into 4 levels:

1. source code application portability
2. utility level, including standard tools, and databases, networking tools, etc.
3. OS interfaces, based on POSIX
4. hardware compatibility, such as Application Binary Interface (ABI)

The group would like to work in the area(s) between levels one and two, to place themselves between application programs and the standard utilities. They do not wish to do the same work as groups already established in the area, such as X/Open, but would rather use these as a basis on which to build. Since the most important thing when trying to accomplish something is to agree on what to accomplish, the group's first task will be to define what *they* mean by the terms *portability*, *compatibility*, and *migration* so as to not have confusion with what others mean with they use these terms. The existing documentation available to the different members was catalogued to give an idea of the work already accomplished in the area and to analyse what future work needs to be done. One of the first realisations of the group was that very often press articles, even in specialised computer magazines, are neither precise, complete, or correct. This reduces the amount of material available, and recommends caution when relying on information gleaned from the press. Only the most serious trade journals should be used, and even they should be verified with care. The work group decided to begin with the following publications:

- standardisation documents (such as the X/Open Portability Guide, and POSIX).
- work done or under study by national, European, or international standards organisations (such as AFNOR, DIN, ANSI, IEEE, CEN, ISO)

- specialised works (such as programming languages or databases)

The group would like to publish a study for information and comparison of what exists today to help portability in the UNIX environment. The deadline set by the work group is the CONVENTION A.F.U.U. 1990, but updates, or reports of the progress of the group should be available periodicly via TRIBUNIX, the A.F.U.U.'s newsletter. A series of informational presentations given to the group is planned. Among these are:

- AT&T SYSTEM V.4.0 (considering portability aspects)
- X/Open Portability Guide
- POSIX
- Programming Language portability (C, FORTRAN, COBOL)
- Shell program portability
- Portable backup methodology
- Portable, yet speaking the French (or other) language(s).
- networking
- databases (Oracle, Informix, Ingres, Unify)
- Sybase, Empress, Clio

DATABASES

The goals of this group are:

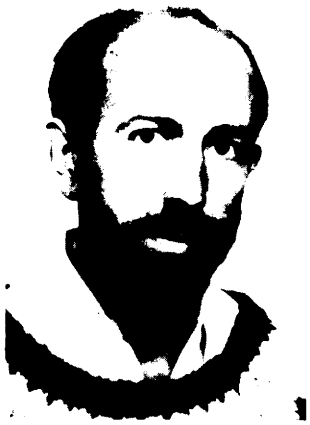
1. compare and categorise the different existing products
2. invite the different companies to present their products
3. conduct a poll of the different types of databases available, and make the results available to the membership
4. author a list of products on the market today that use databases.
5. follow new developments in databases and make that information available to A.F.U.U. members.
6. create a "press-book" containing clippings from newspapers concerning database technologies (this is available for consultation in the A.F.U.U. offices).
7. develop a database benchmark tape in the same spirit as the SSBA.

One of the first conclusions of the group was that they didn't use the same terminology to mean the same things! This lead to the writing of an explanation and definition of terms, in a very professional manner, which has been published in TRIBUNIX. The group has started in-depth studying of the different types of distributed systems and the advantages and dis-advantages of each. Many different products have been already presented to the work group, allowing a "census" of available technology (minutes and notes are available at the A.F.U.U. offices), Also, a teletex communication service (using the French minitel system) has been set up to allow dialogue and exchange of ideas.

Trailblazer and TCP/IP Services in Europe

Peter Houlder
uknet@ukc.ac.uk

Computing Laboratory
University of Kent at Canterbury



Peter Houlder has been in the Computing Laboratory at the University of Kent for the last 4 years and looked after day to day UKnet admin work in the last 3 years.

He graduated in Geography from Kings College, London in 1970 and then spent 9 years in business — dropping out in 1979. He then spent a year touring North, Central, South, and Carribean America, became interested in archaeology and spent three years exclavating in Britain and Europe.

Two Masters degrees, the first in Archaeological Sciences and the second in Computer Science, followed in successive years. Maggie in the meantime reduced archaeological funding, so he arrived in 1984 kicking and screaming into the world of Computing. He has since got to quite enjoy it.

He is married with two labradors.

There are two parts to this column: first a European Trailblazer & TCP/IP survey, and next a status report from Greece.

This article is hot from the press of the July 25th EUnet technical meeting. There may be some inaccuracies caused by the official Deadline for this article also being July 25th, so getting corrections from all the backbones in time is impossible. Anyway here goes .....

Until 1988 EUnet developed using public data networks and low speed modem connections. In the last 18 months two major developments have occurred in Europe.

1) In terms of modems, an increasingly large number of EUnet countries now have access to high speed Trailblazer modems using Packetised Ensemble Protocol (PEP), which can theoretically run up to 18000 bps, but typically run at at between 10000 and 13000 bps. UUCP rates up to 14000 bps are obtainable with the g-protocol. These modems can be set up to run at PEP-only and/or slower speeds, but assuming a 9600 bps rate is achieved, then data transmission typically improves eightfold. It should be stated that these modems can also run at 1200 bps (V.22) and 2400 bps (V.22bis).

2) TCP/IP connectivity, initially based on leased lines between some of the EUnet backbone sites, is spreading rapidly in Europe. This connectivity improves the speed of transmission, lowers the costs (in the longer term) and potentially allows the provision of higher level protocols, such as FTP, SMTP and TELNET. Some EUnet backbones are already offering these services to their national members.

It should be emphasised that the rate of TCP/IP developments in Europe, as a whole and, member countries in particular is determined by various factors, such as availability of lines from national PTTs, the cost of leased line and the size of national networks. There are also various political considerations related to TCP/IP and ISO support in individual countries. It should however be emphasised that TCP/IP services over public data networks or dial-up lines are possibilities, so lack of leased lines is not a prohibitory factor.

The European TCP/IP network is distinct from the US Internet, but EUnet sites that are involved in research and development can be members of both networks if they satisfy US registration criteria.

As for networks numbers, official IP addresses, these can be obtained via a simple e-mail to:

HOSTMASTER@SRI-NIC.ARPA

Only networks with numbers that are officially obtained this way can be hooked up to InterEUnet. Everyone can obtain these for free. Your national EUnet backbone may have a template form for you to fill in. Obtaining full US Internet connect status requires a separate procedure. In principal any organisation that has research contacts on the US internet can obtain this connectivity. Organisations interested should prepare a formless application stating the name and purpose of your organisation, technical and management contact addresses, your IP network numbers and most importantly a short description of your contacts to the US organisations including a contact person there. EUnet has established a procedure to smooth these applications through the appropriate channels. It may be able to both play an intermediate role in obtaining full Internet connected status and it may help in speeding up the handling of your request.

The next section gives a backbone by backbone ordering of sites starting with the European Backbone.

EUROPE—The European Backbone (nic.EU.net, cwi.nl, mcvax) provides full IP support for any national backbone that is able to connect to it. It is also has full connected status on the US Internet. At present it has no trailblazers, but there are no restrictions on Dutch use (see NETHERLANDS below), so trailblazers could be used if the need arose.

The Netherlands—The Dutch Backbone (Netherlands.EU.net, hp4nl.nluug.nl, hp4nl) has full TCP/IP connected status to the European and US networks. At present 1 commercial and 8 academic sites have linked to the Dutch Backbone with IP, but at least 5 others plan to do so in near future. Trailblazers are now used in 40% of all Dutch sites and the Dutch backbone has 10 of its own.

France—The French Backbone (France.EU.net, inria.inria.fr, inria) also has full TCP/IP connected status to the European and US networks. At present it has 25 TCP/IP sites 22 of which have full connected status. France has the fastest growing TCP/IP network in EUnet and supports TCP/IP over leased lines, dial-up and PSS. Trailblazers in France are not yet approved.

Sweden—The Swedish Backbone (Sweden.EU.net, sunic.sunet.se, sunic) also has full TCP/IP connected status to the European and US networks. It is part of both EUnet and Nordunet (the Scandanavian Research and Development network). Nordunet has a mutual backup arrangement with EUnet, so Europe to US links can run over the Swedish link to NSFnet or the mcvax link to uunet. This is mutually advantageous to both networks. In national terms all universities have full IP links, but there are no commercial links at present. Trailblazers are legal; they are actually sold by the Swedish PTT. The Swedish backbone is at present using a Trailblazer for SLIP testing.

Denmark—The Danish Backbone (Denmark.EU.net, dkuug.dk, dkuug) also has full TCP/IP connected status to the European, US and Nordunet networks. Denmark offers full academic IP services and is investigating commercial services. Dkuug will have a pilot ISDN service from early September. The Danish backbone has 2 Trailblazer modems and there are no legal problems related to these modems in Denmark.

Finland—The Finnish Backbone (Finland.EU.net, tut.fi, tut) is similar to Sweden ad Denmark in academic terms, but it additionally offers IP services to commercial companies doing research. It has one experimental Trailblazer, but no Finnish PTT approval has yet been given.

Norway—The Norwegian backbone is on the move as I write, so things should soon be clearer, but Nordunet based IP services are available in Norway. No details of Trailblazer availability/legality are known at present.

Iceland—The situation in the final Scandanavian backbone (Iceland) is similar to Norway. Hopefully details may be available for later articles.

Italy—The Italian backbone (Italy.EU.net, i2unix.dist.unige.it, i2unix) has ordered a leased line to inria (France). It has 2-3 sites waiting to connect and it also has links to CNUCE the EARN site at Pisa, which in turn is linked via leased line to NSFnet. It has 3 trailblazers at i2unix and several Italian sites also use them.

Spain—The Spanish backbone (Spain.EU.net, dit.upm.es, goya) has local IP connectivity in Spain. The leased line position is unclear and Trailblazers are at present illegal. The situation

with leased lines is however being actively pursued, so things may change in the not too distant future.

Portugal—The Portugese backbone (Portugal.EU.net, inesc.pt inesc) is investigating leased lines and trailblazer links. As with Spain the availability and cost of leased lines is a problem, especially as the national networks are still relatively small. It is hoped to start Trailblazer trials from inesc in the near future.

Germany—The German backbone (germany.eu.net, unido.informatik.uni-dortmund.de, unido) should have a leased line to mcvax in the next few weeks. Several sites have plans to use this service. There are already test connections within Germany. Unido's connect status is as a fully registered part of the European and US Internets. Trailblazers are under test as part of a German PTT project. It is fairly certain that they will be legal in 1990. Unido has 2 on test.

Belgium—The Belgian backbone (Belgium.EU.net, prlb.philips.be, prlb2) has no leased line. It had plans for to set-up a leased line, but the decision had to be postponed due to the BUUG possibly assigning the backbone function to another institution. Whatever the final choice of the BUUG, it is expected that Belgium will have a leased line in the very near future. There are discussions in Belgium about the setting up of an academic and research network based on IP. Trailblazers are not yet legal.

Austria—The Austrian backbone (Austria.EU.net, tuvie.at, tuvie) has local IP, but Trailblazers are illegal. The situation with leased line connectivity is at present unclear.

Greece—The Greek backbone (Greece.EU.net, ariadne.csi.forth.gr, ariadne) is negotiating for a leased line to inria or i2unix. There is a possibility of sharing a leased line with EARN running TCP/IP over SNA. In terms of Trailblazers it has 4 but no lines to run them on.

Yugoslavia—The Yugoslav backbone (Yugoslavia.EU.net, idcyuug) has local IP in Lyubljana, but leased line costs at present prohibit any international links. There are no trailblazers in Yugoslavia.

Ireland—The Irish backbone (Ireland.EU.net, tcdcs) has local IP and plans to connect to CWI via leased line, cost however, is a problem. No Trailblazers are currently in use.

Great Britain—The British backbone (Britain.EU.net, ukc.ac.uk, ukc) hopes to have full backbone IP connectivity in the next few weeks. Extensions of IP services beyond ukc is unclear at present. Ukc has 2 trailblazers and the use of Trailblazers in Britain is rapidly expanding.

Status Report from Ariadne

Stergios Alexiou
Stelios Sartzetakis
netman@csi.forth.gr

Foundation of Research and Technology Hellas
Computer Science Institute
Crete

The GRUnet

There is a new small UNIX Network in Greece, the GRUnet, which is based on dialup lines. GRUnet consists of fourteen sites and its rate of growth is rather slow, about five to eight sites per year. Three sites are receiving news but eight to ten sites will receive news towards the end of this year. This new network mainly offers services to universities but also to some companies as well.

Current Status on node ariadne (CSI-FORTH)

Last year there ware many changes, both in software and hardware. First of all we have swapped the old Vax with a Sun-3/280 running Sun-OS 4.0 with one GByte disk space (temporarily we ran on an old Sun-2/120).

The dialup lines increased this year to three: two modems hayes compatible at 1200bps and one Trailblazer modem. These dialup lines are used mostly by Greek sites.

Ariadne on x25: there is an x25 connection to the x25 network of the local PTT called Hellaspac, with baud rate 19200bps. There is also another one connection with x29 protocol with transfer rate 1200bps. These lines are used to communicate with the backbone sites of EUnet.

Evidently we registered the top level domain GR for Greece.

From the software point of view: we have tested (not yet installed) an mcvax-like router in conjunction with the sendmail and pathalias programs. We maintain the Greek maps here, in ariadne, getting all the other European maps from

the other backbone sites. A few months ago ariadne became a node on the EARN network. Ariadne connected to the Greek backbone of EARN, GREARN (4800bps, running urep V3.3).

GRUnet and EUnet

We currently maintain links to following backbones: goya, inria, mcvax, sunic, tuvie, ukc, unido. Except from the links to goya and unido which are bidirectional all other links are unidirectional starting from ariadne. A very good idea is using Trailblazers as backup of the x25 links and we would like to hear the thoughts of the other backboners on this idea.

Future Plans

We are planning to become an archive server. This will be done in one of two ways: the first is based on a video tape drive and the second on Optical disks. To retrieve the desired software there will be a remote query program. We do not have any experience on that so we would expect some help from other experienced backboners on implementing this idea.

To improve our services we are thinking of starting up an EARN gateway for GR domain (GREARN, the central node for Greece is also maintained by another group at CSI-FORTH).

The GRUUG group is under our consideration. We would like to get some precise information on how other UNIX Groups are organising and are providing services.

Our basic intention is to get an IP connection to join InterEUnet. This is rather complicated because the cost of a leased line is too high. For this purpose there are two possibilities. The first one is to get a leased line to i2unix (Italy), which will be connected to inria (France). The other one solution is to share the existing line from Greece to France with Greek backbone of EARN, GREARN. We hope that first we can convince Montpellier people to pass our traffic to Sophia Antipolis and together with INRIA we could find a way to get our traffic routed on the European IP. We also hope that we will have an IP connection in either of the above ways in the next three or four months.

Typical Statistics per month (Mail and News)

The following table shows the traffic on node ariadne:

Mail & News Statistics (1-22/7/89) (in Kbytes)

Site	Sent	Received
goya	12.1	320.1
inria	25	459.8
mcvax	757.6	20522.6
sunic	0	0
tuvie	7.1	0.5
ukc	87.5	417.4
unido	87.4	515.6
Total	976.7	22236
Est. total to the end of July	1200	35000
Greek Sites	4342.7	251.3
Est. total to the end of July	12000	350
Total	5319.4	22487.3
Est. total to the end of July	13200	35350

The news part of the above statistics is : from mcvax we got this month about 20250 Kbytes news and we send to Greek sites about 9160 Kbytes news.

There was a problem with sunic this month. Their PAD was out of work.

There is as yet no estimate of traffic via GREARN.

AUSTRIA - UUGA
Friedrich Kofler
Austro Olivetti Ges.m.b.H
Braunhuberdasse 26
AUSTRIA
Tel: +43 222 743575 Ext 680
uuga@tuvie.at

BELGIUM - BUUG
Marc Nyssen
Department of Medical Informatics
VUB, Laarbeeklaan 103
B-1090 Brussels
BELGIUM
Tel: +32 2 477 4111
Fax: +32 2 477 4000
buug@vub.uucp

DENMARK - DKUUG
Mogens Buhelt
Kabbelejevej 27B
DK-2700 Bronshoj
DENMARK
Tel: +45 2 86 55 33
mogens@dkuug.dk

FINLAND - FUUG
Anu Patrikka-Cantell
Finnish UNIX Users' Group
Arkadiankatu 14 B 45
00100 Helsinki
FINLAND
Tel: +358 0 494 371

FRANCE - AFUU
Miss Ann Garnery
AFUU
11 rue Carnot
94270 Le Kremlin Bicetre
Paris
FRANCE
Tel: +33 1 46 70 95 90
Fax: +33 1 46 58 94 20
anne@afuu.fr

GERMANY - GUUG
Dr Anton Gerold
GUUG-Vorstand
Elsenheimerstr 43
D-8000 MUNCHEN 21
WEST GERMANY
Tel: +49 89 570 7697
Fax: +49 89 570 7607
gerold@lan.informatik.tu-muenchem.dbp.de

HUNGARY - HUUG
Dr Elöd Knuth
Computer and Automation Institute
Hungarian Academy of Sciences
H-1502 Budapest 112, P.O. Box 63
HUNGARY
Tel: +36 1 665 435
Fax: +36 1 354 317

ICELAND - ICEUUG
Marius Olafsson
University Computer Center
Dunhaga 5
Reykjavik
ICELAND
Tel: +354 1 694747
marius@rhi.hi.is

IRELAND - IUUG
Norman Hull
Irish Unix-systems User Group
Court Place
Carlow
IRELAND
Tel: +353 503 31745
Fax: +353 503 43695
iuug-exec@cs.tcd.ie

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

NETHERLANDS - NLUUG
Patricia Otter
Xirion bv
World Trade Centre
Strawinskylaan 1135
1077 XX Amsterdam
THE NETHERLANDS
Tel: +31 20 6649411
nluug@hp4nl

NORWAY - NUUG
Jan Brandt Jensen
Unisoft A.S.
Enebakkvn 154
N-0680 Oslo 6
NORWAY
Tel: +47 2 688970

PORTUGAL - PUUG
Legatheaux Martens
Avenue 24 de Julho, nº 134, 7º
Lisboa
PORTUGAL
Tel: +351 1 673194/609822
Fax: +351 1 7597716
puug@iuesc.pt

SWEDEN - EUUG-S
Hans E. Johansson
NCR Svenska AB
Box 1206
S-164 28 KISTA
SWEDEN
Tel: +46 8 750 26 03
hans@ncr.se

UNITED KINGDOM - UKUUG
Bill Barrett
Owles Hall
Buntingford
Hertfordshire SG9 9PL
UNITED KINGDOM
Tel: +44 763 73039
Fax: +44 763 73255
ukuug@ukc.ac.uk

YUGOSLAVIA - YUUG
Milan Palian
Iskra Delta Computers
Parmova 41
61000 Ljubljana
YUGOSLAVIA
Tel: +38 61 574 554
mpalian@idcyuug.uucp