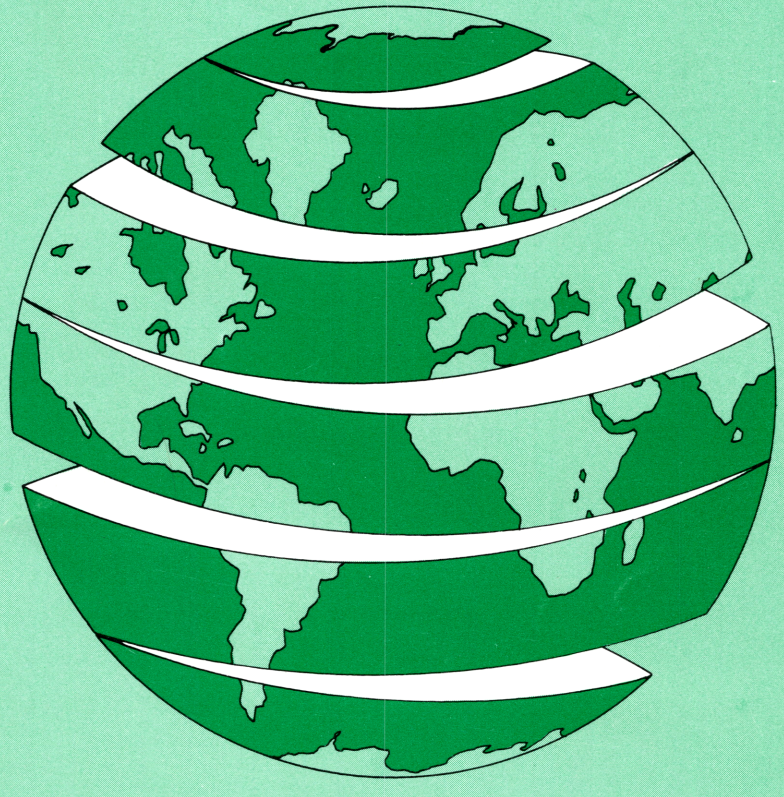
NEWSLETTER

EUUG

European UNIX® systems User Group



Volume 8, No. 4
Winter 1988

CONTENTS

- Changing the *ROFF Escape Character
- Optical Disc WORM File System
- Portugal Conference Reports
- Regional Reports
- E-Mail Directory

EUROPEAN UNIX® SYSTEMS USER GROUP NEWSLETTER

Volume 8, Number 4
Winter 1988

Editorial 1
Changing the *roff Escape Character 2
Optical Disk WORM File System under System V Rel 3.0 6
Portugal EUUG Conference Report 13
A First Vist to an EUUG Conference 18
The German EUnet - Dnet 20
The C and UNIX Dictionary 22
Hungarian Unix Users Group 23
Yugoslav UNIX Users Group 24
News from the Netherlands 25
The Danish Connection 28
The UKUUG National Group 31
EUNET® in Belgium 34
EUUG Executive Committee Report 37
USENIX Association News for EUUG Members 39
SVR4 Conference (London) 41
ANSI C Standard and Progress towards an ISO C Standard 43
The POSIX Standard and Its Future Development 45
AT&T System V/MLS 47
Unix Clinic 51
The European E-Mail Directory 54
EUUG Software Distribution 56
Glossary 63
EUUG Portugal Conference Abstracts 64

Editorial Team:

Philip Peake
Publications Executive
Axis Digital
135 Rue d'Aguesseau
92100 Bonlogne
France
philip@axis.fr

Alain Williams
Parliament Hill Computers Ltd.
7 Prospect Street
Caversham
Berkshire RG11 4DJ
U.K.
addw@phcomp.co.uk

Typesetting:

Brian K. Goldstein
The Instruction Set Ltd.
City House
190 City Road
London EC1V 2QH
U.K.
xobkg@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)

The editor reserves the right to alter any article submitted for publication. This right has been exercised in previous issues of the newsletter.

ISSN 1011-4211

Editorial

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

Parliament Hill Computers Ltd
Berkshire RG11 4DJ
UK
+44 734 461232

Sun, Sea, Sand and UNIX

Back to London after an enjoyable week in Portugal at the very successful EUUG conference in Cascais. The memorable quote of the week was "Oh – I didn't recognise you with your clothes on".

You will find review of the conference on pages 13 and 18, and the conference abstracts on page 65. The paper about optical disk file system (page 6) was originally delivered at the conference.

New Members

The EUUG is growing. Please join me in welcoming the Hungarian group into the club. Sándor Keresztély writes on page 23.

We will shortly be joined by a group from Yugoslavia, Milan Palian tells you about it on page 24.

A New EUUG Service

In January the first European E-Mail Directory will be published. Find out more about this and an order form on page 54.

Software Reviews

Would all those interested in contributing to a section "Section I Review" in future EUUGNs please contact Donal Daly (daly@cs.tcd.ie +353 – 1 – 772941 x1542).

This column will deal with review of public domain software which has appeared on the net or distribution tapes.

Executive Report

To help keep EUUG members up to date with the very latest EUUG news Helen Gibbons has started a regular report on the activities of the Executive Committee. You will find this on page 37.

More Newsletters

Four thousand copies of this newsletter have been printed.

Did you know that corporate members and libraries can now receive more than one copy of the newsletter?

You don't need to wait if you are at the bottom of the distribution list – get your boss to order more copies!

I Am Moving

Please note that my address and phone number have changed. I have escaped from London to a peaceful small town.

Changing the *roff Escape Character

Markku Sakkinen
markku@jytko.jyu.fi

Department of Computer Science, University of Jyväskylä
Seminaarinkatu 15, SF-40100 Jyväskylä, Finland

Graduated in mathematics in 1971 and got the Licentiate degree in 1972. However, had already lost his heart to computers and was not quite bright enough for a mathematical career. Worked at the university computing centre and also at the department of physics (putting up a laboratory computer system). Then did several years of Real Work in the Real World (sort of industrial automation mostly). Returned to his dear old university in 1985, partly in the hope to get a PhD before retirement age, but got to bear hardships, e.g. UNIX™.

Writing this bit was a good way to get my portrait in the Newsletter; I did not mail it to the editor in time for the Autumn issue.

The problem

In my jeremiad [Sakk] in the previous EUUG Newsletter, one thing I complained about (in the section "Fonts and character sets") was the default "escape character" of *roff (not to be confused with the ISO/ASCII control character ESC). It is backslash, which is substituted by some important printable character (typically an upper-case letter) in many national variants of the 7-bit ISO code. Redefining the escape character is possible in theory by using the .ec request, but in practice it conflicts with the standard macro packages.

English-speaking people may wonder why this problem is worth making a fuss about: after all, those funny foreign letters can be output by some means. I will sketch a parallel for them to see the situation. Imagine that the 7-bit character code had been defined on the basis of the pure Latin alphabet. Hence, 'W' and 'w' would be missing; they would be national substitution characters in the British, German, USA, and some other national variants. Suppose further that 'W' was the the *roff escape character '\'. In order to get a 'W' printed, you would then have to write either 'WW' (perhaps 'WWWW' or even 'WWWWWWWW' in some macro arguments), 'We', '(W(VV' (assuming a special character had been defined), or 'W*(VV' (assuming you had this predefined string in a macro package). It would not be convenient to write English text like this. Escape sequences would look confusing, and you would frequently get weird effects by forgetting to write a desired 'W' in an appropriate way so that a haphazard escape sequence would result instead of the letter.

A more fundamental reason why a letter should not coincide with the escape character was pointed out by Seppo Sippu, who has made a couple of Finnish language hyphenation filters. Words that contain the escape character can be uncorrectly hyphenated and, more dangerously, hyphenation indicators can be put into the middle of some escape sequences. (In Finnish and many other languages, very good hyphenation can be obtained purely by algorithm, so there is no dictionary that could exclude those escape sequences.)

A work-around

If you are using any standard *preprocessors* (eqn, tbl, ...) you can forget right away about *really* changing the escape character. They generate lots of *roff input containing standard escapes (i.e.

'\'). Even if you have the source code, modifying a preprocessor would probably require too much time and effort. Suppose that you would like to use '@' as the new escape character; you would have to check both all backslashes and all '@'s that the original preprocessor outputs. We will therefore first present a work-around solution you can take if you use a preprocessor, or if you do not want to fiddle with your macro packages.

When editing your text, use '@' (or whatever your choice) as the escape character and 'Ö' (or whatever your national substitute for '\' happens to be) as an ordinary character. Before feeding the text to the formatter or standard preprocessor, change all 'Ö's into 'Öe's and all '@'s into 'Ö's; you can use a trivial two-line **sed** script as a filter to do this. If you need to print the "pseudo-escape" character '@' also, you must apply some additional tricks. One possibility is to use some string that will certainly not appear otherwise in your source text, and change that string into '@' as the last step in your sed script.

If you use a hyphenation programme for your language that outputs the default '\%' character as the hyphenation indicator, you will have to modify that programme to output '@%' instead. If your hyphenator employs an ordinary printable character as indicator, no modification is needed. Sippu's hyphenator, for instance, accepts the underscore '_' as an *input* hyphenation indicator (useful in words that do not obey the normal rules of Finnish, e.g. foreign words) but *outputs* '\%'s. The hyphenation must be done before the escape character substitution step, of course.

Preliminaries

When Seppo Sippu started a short course on text formatting with Ditroff and associated tools at our department in September, we began thinking about how one could modify a macro package to accept another escape character. After all, it need not be a tremendous undertaking. Very probably somebody has done things like this before, but I have not happened to see a recipe in print, so it could be useful to briefly explain what happened.

There are two obvious prerequisites for this kind of modification. The first one is that all macro packages are plain *roff source text; thus you can process them with your favourite editor and all the other common tools. The second one is that there must be some "spare" printable character to substitute for the backslash: it must have no predefined syntactic meaning for *roff *and* you will very seldom need to print it. Otherwise there will not be much sense in changing the escape character. At least for us Scandinavians, there are some good choices, but not many: the number sign '#' and the underscore '_'. Both have the additional advantage of standing out visually in source text. We tried the number sign because we already had another special meaning for the underscore (cf. previous section). To be honest, we just came to think first about the number sign without realising that it was practically the only sensible choice.

The macro package we use almost exclusively is **me** (from Berkeley). That implies that the main macro file in the appropriate library directory is called **tmac.e** . We decided to call the modified package **mes**, so its main file had to be called **tmac.es** ('s' stands for 'suomi' = Finnish [language], or 'Scandinavian'). However, not everything belonging to the package is in this main file. The definitions of some large macros that are not called very frequently (e.g. only at the beginning of a document) are in separate files in the relative directory **../me** looking from where tmac.e lies. They can be found out by looking at all **.so** requests in tmac.e (these auxiliary files in turn do not contain any more .so requests, but in principle they could).

The .so requests in tmac.e all turn out to be of the form:

.so *(ll/*auxfile*.me .

The name of the directory of those files is thus in the string named 'll', which is set by a **.ds** request at one place in the main file. Accordingly, we made a new directory **../mes** for the modified auxiliary files and modified the .ds request in the new main file (tmac.es). Simply copying all auxiliary files to the new directory would then have lead to an independent "clone" of **me**.

The essential modifications

To our delight, we saw that there were very few '#' characters in all the files in question (except for the SCCS comment lines, which are best left undisturbed). They almost all appeared as first characters in the names of strings or number registers. We reasoned that we could change them to '9's without creating name conflicts; this actually seemed to succeed. After this, all backslashes could be converted to number signs.

Finally, we added the actual escape character change request

.ec #

as the very first line to tmac.es and changed those lines (not many) that had a .ec request without argument (reverting the escape character to the default backslash) to this same form.

At this stage, a test file came through all right, but the above-mentioned request '.ec #' was somewhat disquieting. When we added the same line to our source file, we got a completely haphazard output. What had gone wrong? — Something that would also have resulted if the test file had happened to invoke any macro that contained an '.ec #' request. Namely, when the number sign already *was* the escape, this request line was interpreted as ending with an escaped newline and thus caused the first character of the following line to become the new escape character! All the rest of that line was ignored. Putting a **.eo** request, which disables the whole escape character mechanism, before each '.ec #' should take care of the problem.

Well, even '.eo' was not enough. Most of the .ec requests were within macro definitions, therefore the .eo request could not prevent interpreting the '#' escape already while the definition was being *read*. We had to duplicate the escape character, as so often happens in *roff macros:

.ec ##

Obviously, we could even have done without the preceding .eo requests if we had put *four* number signs here. It does no harm in this request (because it finally uses only the first character of its argument) if one plays it safe and writes a very long sequence of '#'s — who knows how many times any line within a complicated macro package will be processed?

It would have been safer and more elegant in principle to change all original number signs in the macro files reciprocally into *backslashes* than into 9's. That would have required changing each '#' first into some string that certainly did not occur in the file previously, then each '\' into a '#', and finally each temporary string into a '\'.

In order to get Finnish hyphenation to work again, we had to modify the hyphenation programme just as described in the section "A work-around".

Further observations

If you define as the new escape character any possible second character of a *roff escape sequence, say 'l' ('\l' means an 1/6em space), then you cannot use that particular escape sequence. This is so because *roff in these circumstances interprets 'll' as a request to print one 'l'. This again is analogous to the interpretation of '\\' when the escape character has not been redefined, and is necessary for an alternate escape character to function like the standard one. Nevertheless, here we have one factor that greatly restricts the choice of truly usable escape characters.

As a matter of fact, the whole '\\' convention in *roff is less than optimal. Suppose that instead there were an escape sequence with a different second character, say '\>'. The problem of the previous paragraph would then not exist. Also, every processing cycle through which the escape character must subsist uninterpreted requires a *duplication* of the number of characters in the '\\' convention, but would need only *one* additional '>' in the '\>' convention. The existing escape sequence '\e' is fundamentally different from this proposed '\>' in that '\e' is never interpreted in copy mode. Therefore, it cannot result in another escape sequence under any circumstances; it always finally causes *printing* the escape character.

When the modification of **me** looked successful, I tried to do the same to the **mm** macro package from the Documenter's Workbench™. There was a little less work, because **mm** has fewer auxiliary files. The same method worked here, too. I have written the source text of this very article using '#' as the escape character, then processed it with the modified **mm** package and sent the result to the editor as a "galley proof". Finally, I "sedded" the source text to use '\' as the escape and sent that result to the editor by e-mail. Practise what you preach!

During this exercise, I have noted that even when writing in English, it can be advantageous to change the escape character if the original '\' appears often in your subject matter! This holds for instance when you are writing about *roff, the UNIX™ shells, or the C language.

Two notes on my previous article

It looks as if [Sakk] was printed in the Newsletter very much like I expected. The current two-

column format is very good at least for such straight text, I think. Unfortunately, the editorial practice of not numbering any sections can make it somewhat difficult for readers to follow cross-references in the paper, since they were written as chapter/section (or section/subsection) numbers. I should have used names — sorry for being lazy and causing loss of referential integrity.

The subsection "How to get information" in [Sakk] told how difficult it is to get information about the true character sets of PostScript™ printers. Here is some more evidence.

I recently printed some documents on a newer laser printer at our university; as it had PostScript release 47.0A (our regular printer has 47.0), I supposed there would be no problems. But browsing through the output I saw that some special characters were missing: although the new printer had all the same fonts as the old one, it had not got the whole character set of the old printer. One must thus be careful with unfamiliar printers. It would be a good idea to enhance the PostScript programme for re-encoding a character set to tell the names of all characters it does not find. By 'tell' I mean, either send back to the computer or output on the printer itself.

Acknowledgement

This work was supported by the Academy of Finland.

Reference

[Sakk] Markku Sakkinen, The Trouble with PostScript and Device-independent Troff, *EUUG Newsletter Vol. 8 No. 3 (Autumn 1988)*, p. 16 - 27.

A Puzzle

Thanks again to Mick Farmer (mick@cs.bbk.ac.uk) for providing us with a puzzle.

Nowadays no one writes a book without dedicating it to someone. Five UNIX gurus have each written a book and dedicated it to one of the others. No guru is the recipient of more than one dedication.

Mr Kernighan's book is dedicated to Steve Lesk. Mr Lesk's book is dedicated to Brian. Mr Ritchie's to Ken and Mr Bourne's to Dennis. Brian is the christian name of the guru who dedicated his book to Mr Thompson. Mr Ritchie's christian name is Mike.

What is Ken's surname?

Mick will provide us with the answer in the next issue of the newsletter if it is not too hard for him to work it out!

Optical Disk WORM File System under System V Rel 3.0

Caterina Falchi

I.A.N. s.r.l. via Canova, 7
20090 Trezzano s/n MI ITALY

A *WORM File System* (WFS) has been implemented on an optical disk WORM in order to obtain access with standard read/write commands and procedures as for magnetic disks. The *WORM File Management* (WFM) has been directly integrated into the kernel of System V Rel 3.0 via the File System Switch, to ensure that each access to the WFS, via the commands previously developed for a magnetic disk file system, are fully transparent to the WFS itself. The WFM has been tailored to the write once and read many characteristic and optimized in order to obtain:

1. Media transportability. All data and data structures are written and available on the WORM media.
2. Optimized access time and space usage. A virtual magnetic disk partition is used as temporary support for all data (such as superblock, inodes, directories, etc.) subject to frequent changes. A write on the WORM takes place only during "umount", i.e. no more changes are due, to avoid waste of WORM surface.
3. Data integrity. Files are sequentially written along with a header; some information related to files is redundant; special tools for the WFS check have been implemented.
4. Disk block size. The disk block can be dimensioned to optimize I/O transfers and/or WORM usage.

Introduction

The WORM disk drive is a special device which can be both compared with magnetic disk drives and magnetic tape transport units: in fact, it is "random access" as a disk drive and media removable as a tape. Anyway, its special "write once" feature makes it different from both peripherals (which can afford multiple write operations), i.e. any block, once written, can be neither erased or modified. No existing operating system can handle such a write once device, and, even if connected into the system via a SCSI interface (WORM drives are generally equipped with such an interface), its real integration is an overwhelming task.

Three different integration methods can be implemented:

1. To use a WORM as a *magnetic tape* for backup and archive purposes: data are sequentially stored. This method does not take real advantage of the random access characteristic of the WORM, with a consequent slow retrieval process of stored data.
2. To use the drive as a physical copy of a *magnetic file system* in read-only mode, with no way either to change or to update data, in a CD-ROM look-alike mode.
3. To use a WORM via a specific management, tailored for it in order to have all the available advantages without any restriction deriving from its being special. This WORM management can be either realized at
   a. application level,
   b. kernel level.

The **application** method offers a very easy integration of the WORM management into any existing system and can be powerful as well, since it is aimed at the WORM itself. On the other hand, it shows the big disadvantage of using non-standard calls to store and retrieve data, since accessing a WORM is allowed via the application package only and not via the normal operating system calls. The **kernel** method provides without doubt the most powerful, complete and transparent integration of the device, since it offers specialized management of the device along with the usage of standard operating system calls and utilities. On the other hand, it implies actions inside the operating system, the source code of which must be available in order to

operate suitable and proper changes with no consequence to the overall functionality. This techniques also implies portability, full transparency to application and system calls, ease of integration, expansion to the Juke Box†. configuration without any change to the application software. We have implemented this method.

Why System V Rel. 3.0?

This operating system gives the great advantage of being organized in order to insert different File Systems via the File System Switch mechanism. This is done in a way similar to adding a new device driver and can be done without the need for the complete kernel source code. Usually, the management of a FS inside System V Rev. 3.0 consists of two levels:

- A general kernel level, i.e. the interface to the system calls. The switch that automatically selects the "FS dependent" routines has been inserted at this level.
- An FS dependent kernel level, i.e. the routines associated with a specific FS, organized to run its specialized management.

A master file keeps record of the features and identity of every FS. The introduction of a new FS can be achieved by adding its ID and characteristic inside the master file. A simple recompilation of the OS, using the standard makefile technique, yields a "new" OS, where the standard and original kernel libraries coexist with the new FS routines. The system can now be booted, and the new FS addressed and accessed by means of the standard mount call and associated flags. For instance:

```
# mount -f S51K /dev/dsk/0s1 /usr
```

is the mount of a (standard) FS defined on the "/usr" directory;

```
# mount -f W51K /dev/opt/3s0 /worm
```

is the mount of a "new" WFS on the "/worm" directory.

Even if the porting of the WFM to different structures using System V Rel. 3.0 is relatively easy, it can be achieved with the proper knowledge of the kernel, due to the fact that the specific WFM routines heavily interface with the standard routines. This feature makes a program portable to different system architectures with no trouble.

System V Rel 3.0 is not the only OS with such a feature: the Sun Microsystems OS allows the addition of a new file system type in a transparent way via the *v-nodes* structure. The porting of the our WORM File Management (WFM) package to the Sun structures is in progress.

Our solution

The implementation of the WFM in the kernel

The aim of the WORM File Management (WFM) at the kernel level is to offer a WORM File System completely identical to any of the standard File System in terms of:

- hierarchical management of files and directories.
- protected access to files and directories.
- multi user access to the same file.
- contemporary read/write operations of multiple files.

The way to access a WFS (open, close, read, write, etc) are the same as for standard file systems; consequently every application package can be used even if the directories a FS refers to resides on the optical write once media. Obviously, all the commands can be used as well with no change. The only difference is the impossibility to remove a file already stored into the WORM. This operation implies a waste of blocks into the WORM, since it cannot make free a block already written/burned, but it takes one more block to rewrite the modified directory. In our solution, *remove* right can be given to the system superuser only. This command operates at logical level and not at physical one: all instances of a file can

† This is a special peripheral designed to automatically handle the WORM media. It usually consists of two building blocks to house:

- elevator/mechanism to handle media, some basic electronics for a low level interface;
- drives and media, upon the specific applicative needs.

Different types of Juke Box are available for 12", 8" and 5.25" WORM drives. The range of achievable capacities runs from 30/50 GB up to more than 1 TeraByte, depending on WORM capacity and size.

always be retrieved, because they are permanently stored on the media.

The full transparency to the application software makes any system equipped with the WFM the most suitable file server in every LAN environment, using standard communication protocols such as TCP/IP, NFS, PC-Interface, DECnet, etc.

The implementation

Disk layout

The surface of the media, whatever its dimensions, has been split into two areas, the first one is dedicated to the media initialization, the second one to the current and normal usage.

1. The init area consists of the first 100 blocks and contains all information on the media partitioning and its identification name. The initialization block also contains the name associated to the media during its initialization: in this way, the single media can be "software recognizable" and easily managed inside a Juke Box for automatic handling.
2. In the second area, up to six partitions can be set up; each partition will be assigned to a different WFS.

The structure of the WORM File System

The internal structure of a WFS can be described as follows (see figure 1). Each mounted WFS is built up of a WORM partition and a magnetic disk partition (VP). The WORM disk area is structure as shown in figure 1.

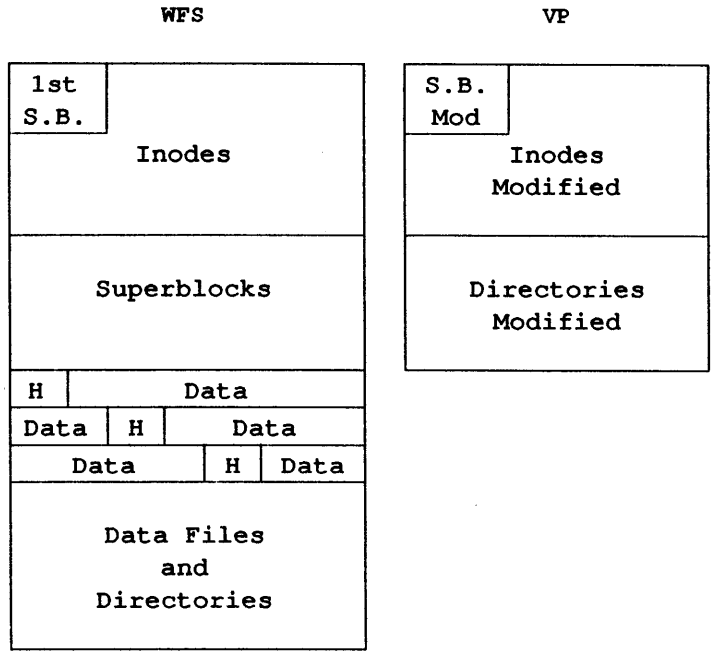


Figure 1

1. 1st Superblock: it is written during make file system (mkfsw) and cannot, obviously, be modified.
2. Inodes area: its size is defined when creating the WFS and, consequently, it cannot be modified. Each inode is as large as 64 bytes. The WORM inode internal structure is different from a standard inode, since it does not use the "i_addr" field to address specific data blocks of a give file. Thanks to the sequential write, a file can be traced by inserting, into the relevant inode, a pointer to the header block associated with the file itself.
3. Superblock updating area: each superblock, once modified, is written into a contiguous block: in such a way, the last written superblock is immediately and quickly traced via the "VERIFY" SCSI command, which gives back the number of the first non blank block available in a specified area.
4. Data area: this area is reserved for files and directory data. Every file is sequentially written along with a front end header for a two-way identification. This header contains the following information:
   - name of file

- number of version
- pointer to the next header
- copy of the inode
- name of parent directory

96 bytes are needed to store this information, but, since the maximum WORM block size is 1024 bytes, every header will occupy one full block. To avoid waste of WORM area for very small size files (smaller then 928 bytes), one block will store header and file data together in the same block. Anyway this header cannot be accessed by standard read/write syscalls of files and directories, but only by opening the raw device; it is useful for filesystem consistency checks performed by special utilities (e.g. optfsck).

Finally, sequential write has been chosen for the following benefits:

- data integrity – files, once written on the WORM, can always be retrieved.
- access time – once a header has been identified and addressed, data blocks are read sequentially, with no need for time wasting seeks. This feature is a must when dealing with a Juke-Box configuration or a heavily loaded system or file server.

The virtual partition (VP) is used as a temporary scratchpad memory for all information related to a specific WFS. The VP is for keeping all information and data subject to frequent changes when accessing a FS, such as the superblock, inodes and directories, in order to avoid wasting a WORM disk block for fast changing. The VP is dynamically matched to a WFS and automatically cleared when the associated WFS is unmounted, ready for a new (possibly different) WFS to be mounted.

The WFS during its life time

To clarify these operations we will examine what happens during the daily operating cycle. First of all, a WFS has to be mounted: during this phase, the VP is initialized by copying the superblock data from WORM-disk; the inode area is built up as well as the hash tables for a fast search of the inodes subject to change during the subsequent operations. Actually, the VP is needed only if the WFS is mounted in read/write mode: when mounting a WFS in read only mode (no change/no write allowed or no more free blocks available) there is no need to associate a VP to a WFS, since the WFS itself carries along on it all the information needed to access its data. This allows us to save magnetic disk partitions when retrieving data only, which becomes a significant issue in large database environments, where many could be monted at the same time.

A file is not directly written on the WORM, but, via the standard write routines to a temporary directory on magnetic disk. When close is executed, it is automatically copied onto the WORM, with no extra operation requested from the user. Obviously this procedure penalizes the write time (a double write is needed to both magnetic and WORM disks). This should not give any real problem, since WORMs are mostly used in read mode. On the other hand, this delayed write offers two major advantages:

1. Simultaneous writes of multiple files.
2. Automatic check of the available free blocks on the WORM. There is no possibility to burn/waste WORM area if not enough free space is available to store the whole file.

All data temporarily stored in the VP are automatically copied onto the WORM when the WFS is unmounted. In such a way a WFS is always in a consistent state, keeping all data and header information on itself. This allows for the safe removal of WORM media.

The advantages

We think that our approach has the following advantages:

- full transparency – this is applicable to stand alone and/or networked systems, with or without Juke Box expansion. This implies the possibility of building either up a data bank, or a mass memory system, an electronic archive or all of them without any change to the existing data base or communication applications.
- full data integrity – the sequential writing of a file provides the means to recovery, under every circumstance, due to the presence of the relevant header. For this purpose, special applications packages supplied with the WFM can completely retrieve any file.
- retrieval / access to the previous version of a file – due to the characteristic of the WORM (i.e. once written data cannot be

cancelled) it is possible to monitor its evolution. The WFM keeps record of this file evolution and modification, in order to allow an access to every previous version. With a very easy syntax, it is possible to specify which version is requested, whilst, by default, the WFM points out the very last version.

Performance

The WORM media so far produced by manufacturers, are preformatted to run a minimum block size of 1KB. Consequently, the WFS developed is based on such a block size. When dealing with large dimension files, a transfer at 4KB block size has been implemented: this method reduces the number of access to the device for better performance: WORM devices can only provide access time in the range of 150 msec, therefore this optimization process yields a significant improvement. When in read only mode, the user himself can change the block size up to 64KB. Furthermore, data transfer does not take place via the I/O system buffers when the File Data Block (FDB) is higher than 4KB: data are automatically transferred into user memory. Choosing the proper FDB affects a lot the overall system performances: based on the National Semiconductor hardware ICM3216, the following behaviour has been measured for a 500KB transfer in the WFS:

WORM 500 KB transfer Test		
FDB	elapsed time	CPU time
1K	10	2.5
2K	9	1.65
4K	8	1.35
8K	6	0.5
16K	4	0.3
32K	3	0.2
64K	2	0.5

and in a standard magnetic FS:

Magnetic Disk 500 KB transfer Test		
FDB	elapsed time	CPU time
1K	8	2.45
2K	8	1.6
4K	7	1.4
8K	7	1.5
16K	7	1.25
32K	6	1.3
64K	6	1.2

As you can see large block sizes greatly improve the I/O performance, without any penalty in functionality.

System administration

For a simplified and easier use of the WORM drives and their related WFM, same support tools have been implemented, as described here:

- optinit – it writes an ID label on the WORM in order to retrace it; it assigns the physical dimensions of the WORM partitions that will be associated with the WFSs. The partitions cannot be overlapped and cannot exceed a maximum number of 6; maximum size is limited only by maximum size of WORM media (1.2 Gbytes presently).
- load – after the WORM is plugged in, "load" makes the drive ready. This command can be executed only if the WORM is already initialized, and, once completed, it disables the front panel switches of the drive in order to avoid wrong handling during operation.
- unload – the drive where the WORM is plugged in exits from ready state and its lock mechanism is released. The WORM can be unplugged. Some WFS could be still mounted at the time "unload" is run: this command takes care of unmounting them and to update the WFSs with all the necessary operations to ensure a correct data and file write, data integrity and complete set of information for the file management.
- optfsck – data are written on the WORM along with the reference parameters needed to retrieve them even the in case of system crash, power down or failure in general. This data structure allows us to retrace and rebuild any files corrupted when the system went down. Optfsck (Optical File System

Check) has been developed for this purpose. It is roughly based on the standard fsck utility, but it has been tailored to the particular structure of an WFS partition (as has already been described). Optfsck first tries to use all data stored on the VP that was associated with the WFS, in order to restore all partially written directories and inodes.

There are also same minor utilities for associated tasks (like managing configurations, handling virtual partitions, etc.).

Applications

Our WFM has been used in same real applications. Some examples are:

- Archiving text and images scanned via a PC – The MS-DOS world offers various low cost and valid application in the optical scanner area: the MS-DOS/ connectivity has been organized by the PC-Interface network package, which translates MS-DOS into commands. The full transparency of the WFM makes it possible to access a WORM drive or a Juke Box from a PC by using standard MS-DOS commands. An MS-DOS image and text retrieval program has been successfully tested and used. The overall configuration, laid onto a Cheapernet cable, is the most standard LAN where the single PC based workstation can access a high capacity server, where all company data (including manuals, data stocks, drawings, etc.) are stored. This configuration has been running inside I.A.N. since April '88 with the target of testing/debugging and organizing the paperless office. No problem has been detected up to now, after heavy duty use of the WFM.

- Databases for CAD-CAM environments – Some tens of CAD-CAM stations already connected via Ethernet have been equipped with our File Server. These workstations are based on Apollo and DEC hardware and can transparently access the off-line volumes kept inside the Juke Box as though they were on-line. No problem has been detected when going from the single WORM to the Juke Box based configuration. The network standard are IEEE 802.3 with the TCP/IP protocols. The server configuration is based on the NSC ICM3216; the Juke Box is equipped with 4 WORM disk drives. The single WORM is handled as a single partition/volume.

- Others – Many places have been working up to now with our complete package, i.e. hardware and software together. Out of them, we would like to point out:

  - University of Genova;

  - BMW (Munich, West Germany): Defined as one of the best packages they have up to now evaluated, the WFM has to be ported to their SUN workstations and network.

  - Cooperativa Informatica (Roma): A complete office automation is on going for large end user environments. The complete connectivity to most differentiated system configuration makes it appealing.

Conclusions

A WFM directly introduced into the kernel has been implemented, along with some support to the user packages at the application level. Its full transparency and compliance with the standard routines and system calls makes it suitable to every application. Its functionality spans a wide range of storage capacity, ranging from a single WORM drive (~1Gbytes) up to a whole Juke Box (~1Tbytes), and it provides the same transparency to any LAN environment/software. Some significant installations/pilot customers have proven the stability and reliability of the WFM.

Acknowledgments

Mr Almondo, IVECO FIAT Group, for his continuous support to identify and evaluate the real needs of huge Data Bank applications in a CAD-CAM environment.

Prof J. Marino, University of Genova, for his qualified assistance during the various steps of our implementation provided either at user and at technical level.

National Semiconductor, for the complete assistance and openings to us, aimed at obtaining the most qualified product.

Bibliography

1. P.J. Weinberger, "The Version 8 Network File System," *Proceedings of the Salt Lake City Usenix Conference*, Salt Lake City, UT, June 1984.

2. S. Leffler, M. Karels, and K. McKusick, "Measuring and Improving the Performance of 4.2BSD," *Proceedings of the Salt Lake City Usenix Conference*, pp. 237-252, June 1984.

3. David Maier, "Using Write-Once Memory for Database Storage," *Proceedings of the ACM Symposium on Principles of Database Systems*, March 1982.

4. Stavros Christodoulakis, "Analysis of Retrieval Performance for Records and Objects using Optical Disk Technology," *ACM Transactions on Computer Systems*, Vol 12, No.2, June 1987.

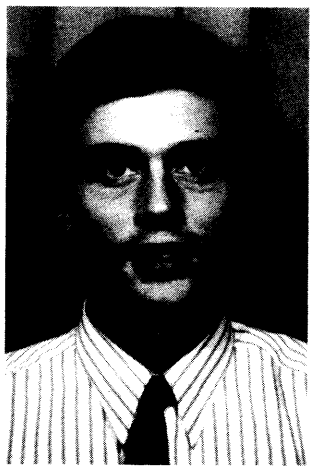
5. Peter Rathmann, "Dynamic Data Structures on Optical Disks," *Proceedings of the IEEE Data Engineering Conference*, Los Angeles, 1984.

6. Maurice Bach, *The Design of the Operating System*, Prentice-Hall International Editions.

Portugal EUUG Conference Report

Tony Smith
ajs@root.co.uk

Unisoft Ltd
Saunderson House
Hayne St
London EC1A 9HH
+44 1 606 7799



Tony Smith is a systems programmer with UniSoft Limited specialising in networking and operating systems.

He studied Information Technology at Leicester Polytechnic and is still searching for somewhere that sells beer as cheap.

After a very busy year, he found time to top up his tan by attending the EUUG conference in Cascais, Portugal.

A Warm Place Far Away ...

(The EUUG Conference at Cascais)

Sunday

Travelling with a colleague from UniSoft, Jeremy Harris, we arrived at Heathrow, just in time, after a very fast drive around the M25. We jumped out of the car and were unloading luggage when a traffic warden (why are they always the bearers of bad news?) pointed out a large puddle underneath the car. Investigation revealed that it was brake fluid. Departure time was looming closer as Jeremy and I tried to locate the problem, without success. Eventually I had to assure my wife (we had only been back from honeymoon one week) that the car would be OK, just go easy on the brakes and clutch.

Relaxing in Air Portugal's club class cabin drinking Champagne made time pass very quickly. The landing at Lisbon was smooth but a little abrupt; the pilot only used half the length of the runway.

Clutching a printout of Neil Todd's instructions on how to get to the hotel by public transport we searched for the bus stop. We eventually found it tucked away around a corner outside the perimeter of the airport. The bus journey into the centre of Lisbon was a real bone shaker. I was particularly impressed at the speed at which the driver loaded passengers on board, sometimes only stopping for a few seconds. Next came a forty minute train ride literally along the edge of the beach to Cascais. A large display on the side of a building revealed that it was 29 °C, no wonder I was breaking out in a sweat.

First impressions of the 'Hotel Estoril Sol' were expensive. Our taxi was met by a smartly dressed doorman who proceeded to carry all our luggage (this was quite impressive as I was having enough problems carrying my luggage let alone Jeremy's as well). Check in went smoothly although I heard of some people having problems because they wanted to extend their stay by one day.

As I slumped on the bed listening to gentle hum of the air conditioning slowly my mind returned to Lisa, has she made it home? I picked up the

'phone and started dialing. One hour later, with a very sore finger, I got through. Fortunately everything was OK. The Portuguese telephone operators had been very unhelpful in getting me connected.

Next on the agenda was food and beer. We ate in hotel on floor 'R' which, as it turns out, was the base for the conference. The food was expensive but OK. Before we turned in for the night we sampled the local, ice cold, beer in a bar populated by some of the dedicated executive.

Monday

Rise for breakfast at 8:30. Bleary eyes meant that I needed that first cup of coffee to kick-start me into life. The waiter poured us coffee in what can only be described as a soup bowl sized container, this hotel was looking promising. The coffee was industrial strength too, a couple of cups of this and even I would fail a drugs test at the Olympics.

Registration was quite painless. Then I made my way towards the "Introduction X Windows" while Jeremy went to sun himself on the beach. The tutorial was well attended, about sixty five delegates, proving to be the most popular by far. This probably reflects the growing number of low cost bit-mapped work stations appearing on the market.

After a short while I decided that those who didn't bother with the introduction were sensible. There was very little information content in the tutorial that couldn't be gleaned from the notes and it was being covered at a very slow pace. One of the three presenters seemed to have a fascination for gathering statistics by getting people to raise their hands; this was beginning to feel like primary school.

As is the way of the Unix industry today a large amount of time was spent discussing politics and in particular Sun Microsystems. This level of discussion reached a point where one of the delegates had to remind the presenters that we were here to learn about X Windows and not to discuss politics – a lesson for us all.

Overall, the tutorial was well presented and came with a clear set of notes. It must be difficult to aim at a level suitable for sixty five delegates from not only different countries but different technical skills. On the whole the tutors coped very well.

In the evening Jeremy and I decided to investigate Cascais, the hotel was situated between Cascais and Estoril with both a short walk away. The hotel tourist book describes Cascais as "The bay of yachts and fishing boats provide a particular charm to the village which grew to be the most fashionable resort in the first decade of the Century."

Well, there were a few boats in the bay and the village did have a certain charm about it, even though it was suffering from end of season dullness. The word fashionable always makes me think of expensive, and Cascais could certainly be expensive if you weren't careful.

We wandered down almost every charming little alley checking out most of the restaurants. Finally, we selected one in a quiet little back alley with what looked like good food at a reasonable price. Unfortunately, tourism was so quiet that we disturbed the owners during their evening meal. We left a big tip as compensation.

Tuesday

Breakfast consisted of croissant, bacon, scrambled egg, industrial strength coffee. How do they manage to get the bacon so tough?

I was booked into the "Programming with X Windows" tutorial and Jeremy the "Systems Performance and Monitoring". Disaster struck. Tutorial notes for the programming with X had all gone. Oh well, further copies were promised by lunchtime. A large group of us spent most of the morning straining to see code fragments displayed on the OHP. At least I managed to get a copy by lunchtime, others were less fortunate. Perhaps there should be a more fool proof system for note distribution to stop this happening again?

Again the tutorial was of a high standard but too slow. Even when asked to increase the pace I didn't notice any improvement. It was a shame that 'no hands' on experience could be obtained, still we could always play with X windows at the vendor exhibition.

The vendor exhibition included names such as Apollo, DEC, Megadata, Siemens, Sun, UNISYS. Following tradition, the largest crowds formed around the vendor with the best graphics display or game.

I met Jeremy for lunch, he was not learning much from his tutorial. At least I was finding mine much more informative than yesterday.

The afternoon was hot and stuffy. Predictably we ended up in the air conditioned bar for an ice cold beer. It was amusing to see one British guy ask for a long island iced tea (a cocktail), a blank look came back from the barman, he repeated it, still no response, British guy then asked for a "G and T" and was given a cute little pot of tea!

EUUG welcome drinks followed. There was a very wide range of wine with smart waiters to serve. Judging by the noise level everyone was having a good time and making new acquaintances.

By now my lack of co-ordination was telling me that it was time for food. The main restaurant in the hotel was shut so we tried the "O Grill". This had subdued lights, man playing piano and corresponding high prices. We managed to annoy the wine waiter by ordering a cheap rosé. The food was good until we considered the price.

Wednesday

Start of the main conference sessions. In the introduction we were told that an attempt had been made to give the conference a European flavour. The conference certainly managed that, once again proving that there are a lot of interesting things happening in Europe. Most of the papers were on "hot" topics.

The morning started with a mix of papers on operating systems followed by a session on security. Generally the standard was very good especially as many of the speakers weren't using their native language.

In the afternoon I felt a little sleepy, not because of the papers being presented but because I had drunk too much wine for lunch. In between short naps I listened to papers on locking in NFS and file systems. One of these papers by David Hendricks from Sun discussed the "Translucent File System" which is a Sun file system with copy-on-write semantics. This allows users to be isolated from each other's changes and preserve disk space. The system sounds as though it solves many of the problems of a shared source tree, however judging by the barrage of questions at the end it does not solve all the problems being faced in the real world.

The day finished with a paper on the OSI transition plans of EUnet and other interesting developments. The impression I got here was that there were plenty of plans that could be implemented as soon as a demand started.

We rounded off the afternoon/evening in the bar. The sign of a good barman must surely be when he prepares us a couple of beers without any prompting. I was getting to quite like this lifestyle.

We investigated Estoril this evening. After walking miles we ended up having a good hot curry.

Thursday

I overslept this morning (intentionally) the thought of listening to papers on "Standards, Proving and Modelling" did not appeal to me at this time of morning.

Next was "Object Oriented Window Systems". Object oriented certainly seemed to be one of the most frequently used expressions throughout the conference. We had object oriented programming languages, databases, toolkits. Papers involving windowing systems appeared in many guises throughout the whole conference.

The problem with conferences is that there are always a few hundred people trying to get to the same room for the same time. The hotel Estoril-Sol was particularly bad as there were only four main lifts. The more adventurous found a service type lift around the corner which was not only faster but hardly used.

Thursday evening was the conference dinner. I had no idea where we were going, all I had been told was to be in reception at 19:30 for travel by coach to Lisbon. During the journey our guide gave us a brief run through Lisbon's and Portugal's history right through to current day. The coach made a brief tour past some of Lisbon's more historic buildings and finally came to rest in a dimly lit street where we were told we would have to walk. As we left the coach a band struck up playing music that sounded as if it would be better placed at a funeral. Just like the Pied Piper of Hamlin, three coach loads of delegates followed the band.

Eventually we arrived in a court yard, somewhat like a close packed Portmeirion, with large tables everywhere covered in typical Portuguese cuisine. The band took their place in the corner and everyone started helping themselves. The court yard was formed by buildings that had been rescued from areas of Lisbon threatened with demolition. Each had been restored and together

with other antique artifacts, such as a beer pump in one corner, blended together to form a unique atmosphere.

Everywhere I looked there was food : pork, chicken, cheese, sausage, salad, soup and fish and plenty of it. Next we watched an ethnic dancing display whilst we stuffed ourselves with dessert followed with a selection of smooth brandies.

Investigation inside the buildings revealed that the place was a complete museum piece; objects, room decorations and tile works all recovered and carefully restored.

I heard one American girl say, "Gee we never has this much fun in America"! A German later described it as somewhat surreal : "a court yard surrounded by ancient buildings, offset by modern party decorations, coloured bulbs and white doves, being played bad music and fed ten-inch fish complete with heads". An enjoyable and memorable time was had by all.

On arrival back at the hotel we headed off for a few beers at the "Duke of Wellington", this gives an idea of the influence of British tourism in the area.

Friday

Last day of the conference. It was now easy to tell those who had spent the conference besides the swimming pool - they had sunburnt faces. It was quite surprising the number of people that had this "guilty" look.

I enjoyed a light hearted paper by Bubbette McLeod entitled "Sacrifices to RA or Learning to Administer a Sun Network". Almost everyone present probably had suffered from one, or many, of the hazards she described. She also described how when laying cables up a ladder wearing a skirt she got too many offers to hold the ladder. I don't think that too many of the conference attendees had suffered from this problem!

As it was Teus Hagen's birthday it could not go by without a customary dunk in the swimming pool. Following the final paper he was thrown into the pool fully clothed. Alain Williams then gave a demonstration of entering the swimming pool from the top diving board, an impressive feat especially as it was voluntary.

Finally, I would like to thank all those who organised the event. I certainly enjoyed it and look forward to my next (whenever that may be).

Competition Result

As is usual at an EUUG conference there was a competition. `/bin/cat` on your machine is damaged. You need to read `/etc/passwd`. What do you do? Marks will be awarded for originality.

The winners were:

1) Paul Dourish, Edinburgh University

Post `/etc/passwd` to eunet.general. Not only will you get your own posting, but you'll also see at least 10 follow ups which include it ...

2) Simon Brown, Meiko Ltd, UK
(also Torstein Beyer, Denmark)

```
rm -f /etc/motd
ln /etc/passwd /etc/motd
^D
... then login!
```

(It was later pointed out that this would only work on a BSD system, as in System V /etc/motd is printed out in /etc/profile by 'cat -s')

Some of the solutions suggested are:

— while read x; do echo $x; done < /etc/passwd

— ln /etc/passwd /tmp/fred.c; cc -E /tmp/fred.c

— nl /etc/passwd | sed 's/^[0-9]*[^I]//'

— cd /dev; ln 'tty' passwd; cp /etc/passwd .

— echo .DS > foo1; echo .DE > foo2; nroff -mm foo1 /etc/passwd foo2

— grep \^ /etc/passwd

— /bin/csh -vn /etc/passwd

— cd /dev; ln 'tty' passwd; cp /etc/passwd .

— Sunview specific:

```
awk '{print "\""$0"\" date"}' /etc/passwd \
    > /usr/lib/rootmenu
    suntools
```

Then press the menu mouse button anywhere in the background ...

— alias cat "grep '.*'"; cat /etc/passwd

— crypt key < /etc/passwd | crypt key

This method is very secure, especially if you do not have trusted pipes.

— diff /etc/passwd /dev/null

— echo .so /etc/passwd | soelim

— dd if=/dev/passwd of=/dev/tty

— tr a a < /etc/passwd

— sort -m /etc/passwd

— tar cf - /etc/passwd | dd skip=1

— echo '0?-1s' | adb /etc/passwd

— rev < /etc/passwd | rev

No guarantee is given of the correctness of the suggested solutions.

Some people called this competition the "dead cat" competition. This led to the comment that it *may* have been inspired by Erwin Schrödinger. This was not so, the truth is that various individuals (who later made up the competition committee) had spent some time in a very similar situation when trying to break into one of the machines on display as part of the conference showcase exhibition.

A First Vist to an EUUG Conference

Anke Goos
anke@unido.uucp

Computer Science Department
University of Dortmund
4600 Dortmund 50
P.O.Box 500 500
W-Germany
+49 231 755 2444

As a member of the German Backbone I take care about end-user information and press-articles (like this). As a student of Journalism I am - not surprisingly - also involved in editing of the quarterly of the German Unix User Group and this year spending some time on the Unix magazine "Topix".

Subject: Letter to the Editor :-)

To be honest, it was not my idea to give my impressions of the recent EUUG-conference in Portugal. On the contrary I disagreed with Philip Peake that I was not at all the "typical" conference participant: Not being a Unix-Guru or expert but a real end-user *and* a member of the German Backbone, for the first time at a EUUG-conference and involved in this EUUG-project called the "E-mail Directory" and lastly, but not least, ... a woman. But Philipe is right in asserting that there is "no" typical conference member and - as responsible for the newsletter of the German Unix User Group - no one ever wants to be the person who writes down comments!

But nevertheless the main influence during the conference was my involvement in this Project which caused me to deal and talk with most of the national EUnet-Backbone people, explaining details about the book or exchanging ideas and the corrected versions of their address parts. These talks with people I had so far only known through e-mail or never at all was the most exciting experience of all. Other people would have had other contacts but I would say that it's hard not get to know a lot of interesting people. All these things that had to be arranged or done, people I had to meet, were the reason that I sometimes didn't even get to the talks! And if eventually sitting down on a seat thousand ideas sparkled inside my head!

So the little person on the monitors really had to do fantastic things to catch my eye... All the better for the conference that I still got the impression that the talks were of a much higher and still quite understandable level than those of national conferences. Sometimes I especially appreciated the way the speakers presented their knowledge. I heard claims from people who missed these "Unix-Stars" who had been consciously left out this time to focus on the good of Europe. Not able to compare this conference to a former one I would still vote for more topics from the research at MIT, Media Labs or other famous US-projects as I will hardly have the chance to attend a US-conference.

But it really wasn't the talks that made the EUUG-conference worth while for me. If you're only interested in some special topics you may better acquire the conference proceedings and spend some nice hours reading.

Of a greater interest were the workshops or 'bof's and then the "people". Although we didn't manage to speak about most of the planned topics at the Newsletter-bof - perhaps a first meeting always is a bit chaotic :-) - but there now is a common mailing-list and some ideas for a better cooperation of the national "information-workers" and I have already exchanged articles with the French Unix User magazine "Tribunix" and dealt other things for a better cooperation.

The "surroundings" made the conference worth-wile. No, not only Lisboa or the beach, although this gives a pleasant mediterranean atmosphere. In one week's time you may only have one day for visits to Arabian influenced palaces and forts and parts of Lisboa and another day for the beach, (counting a whole day for the backbone-meeting). Although there was a little EUUG sight-seeing tour by bus in the dark evening I will never be able to answer this frequent question: What about that burned inner city of Lisboa? And if you don't steal yourself away during the talks for a few hours at a wild beach (which I did) your parallel-processing between "pool" and "talks" is constrained to the water of your hotel. The later still is not too bad, nor where these little EUUG portwein-orgias of my German group late in the night at our hotel...

But the most exalting factor of the EUUG conference was this kind of "cosmopolitan" or "europolitan" atmosphere. Let it be the famous "Network-traveller" John F. Quarterman or Vito, this France-experienced Italian researcher in computer music and this British "Philip" :-) living in France. One evening I actually got the impression that living in Germany as a German really is out of date ... Communication by EUnet already is a great thing, but EUUG-conferences are even better. And I experienced that one fosters the other.

Speaking of communication I would like to say a word about language: On EUUG conferences you sometimes also get this amazing feeling that - listening to English - you're understanding French or Italian in the same time! This way you may also dare to spit out some incomprehensible English-like noises in the beginning. Nevertheless I would recommend not to stay in a national group (like we did the day before the Backbone-meeting) but to take any chance to train in English and French as soon as possible.

I was told that I should not forget to encourage other species of my minority, whose proportion at the conference was still below that of their participation in the Unix-working world. Well, as a woman at the EUUG-conference you still have the chance to be regarded as some kind of exotic attraction. A (male) German unix user told me that international events were only open to the "indian bosses", not to the "poor indian". Maybe this is one reason. One effect of this poor ration is that you even get more attention by your colleagues and a lot more people than I could afford time to speak to! I decided not to take this as a discrimination, but to enjoy the whole conference, keeping in mind, that this "strange" thing of an EUUG-conference really is something special and will last no longer than this week. After it I could scent the atmosphere of EUUG-conferences for the first time.

See you in Brussels.

Anke Goos
anke@unido.uucp

EUUG

European UNIX® systems User Group

EUUG CONFERENCES — The World of UNIX at Your Feet

Twice each year — in the Spring and Autumn — the EUUG holds major International Conferences encompassing all the most interesting developments and activities associated with UNIX.

These events are unequalled anywhere in the world for their content and the very high level of speakers — invited from the leading academic and industrial UNIX centres in the USA, Europe and the Far East.

The importance of the Conferences — which are accompanied by tutorials and exhibitions — is underlined by the fact that well over 2000 delegates have attended the last six events in Florence, Manchester, Finland, Dublin, London and Cascais, Portugal.

If you are at all involved in the field of UNIX, then the EUUG Conferences should not be missed. Indeed, you can make no better start than to put these dates in your diary:

Brussels, Belgium	**Spring 1989 (3-7 April)**
Vienna, Austria	**Autumn 1989 (18-22 Sept)**
Munich, Germany	**Spring 1990 (23-27 April)**
Nice, France	**Autumn 1990 (15-19 Oct)**
Norway	**Spring 1991 (Dates to be announced)**

or call the EUUG Secretariat for further details:

European UNIX® systems User Group

Owles Hall, Buntingford, Hertfordshire SG9 9PL, UK

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

Network address: euug@inset.uucp

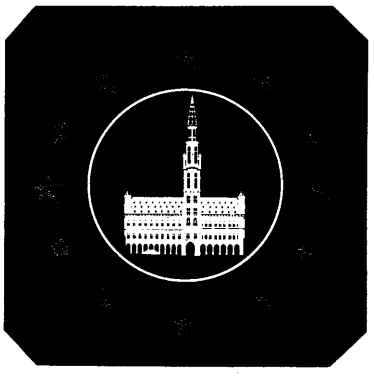
EUUG

European UNIX® systems User Group

EUUG Spring '89 Conference
Exhibition and Tutorials

UNIX: European Challenges

to be held at the



Palais de Congrès
Brussels 3-7 April 1989

The European UNIX® system User Group is to hold its Spring '89 **Technical Conference** at the Palais de Congrès, Brussels, from 3-5 April 1989.

The general theme will be the European Challenges for UNIX and such issues as Real Time; Networking; Security; Graphics; Internationalisation; Distributed Processing; Fault Tolerance; Architecture; Transaction Processing; Window Systems and Environments; Supercomputing; Standards and Conformance Tests are all in the list of topics to be covered.

The event will be run in parallel with an **Exhibition** of many of the latest developments and activities in the UNIX field and will be followed — on the 6th and 7th April — by **Tutorial Sessions.**

Booking forms will be available early January 1989. Further details can be obtained from:

European UNIX® systems User Group

Owles Hall, Buntingford, Herts SG9 9PL, UK
Tel: + 44 763 73039
Fax: + 44 763 73255
Network address: euug@inset.uucp

The German EUnet - Dnet

Anke Goos
anke@unido.uucp
ag@laura.irb.informatik.uni-dortmund.de
ag@laura.uucp
ag@unido.uucp

Computer Science Department
University of Dortmund
4600 Dortmund 50
P.O.Box 500 500
W-Germany
+49 231 755 2444

As a member of the German Backbone I take care about end-user information and press-articles (like this). As a student of Journalism I am - not surprisingly - also involved in editing of the quarterly of the German Unix User Group and this year spending some time on the Unix magazine "Topix".

Crisis, what crisis?

The origins of the German EUnet, sometimes called Dnet, are lie somewhere in the darkness of late 1983. Out of an experimental Status some 13 nodes were connected together in early 1984 and formed the German part of EUnet, the backbone being at the University of Dortmund. The running of the backbone and the routine administration of the network has ever since been in the hands of a postmaster-team of students, together with an official representative, nowadays Ruediger Volk.

Administration

I don't have to tell you what the job of a normal EUnet-Backbone involves. At the moment six students care for the technical and administration support of the backbone-machine *unido* and the fast-growing community of German EUnet-members. In the administration of the network the backbone takes great advantage of the infrastructure provided by the university, although all of the costs of the backbone are paid by German EUnet members.

As of summer 1988 about 180 companies or universities are linked to the backbone, 150 of them by direct links. But due to high fees only 30 sites subscribe to international news. A gateway for mail from Bitnet/EARN to German EUnet members is administrated by the postmaster-team at *unido*. Furthermore, an archive-server is running at *unido* to provide better information, but not that much public domain software has been kept up until now, due to a lack of space.

In spring 1988 the backbone-host was upgraded to a MX500, a Sequent-like machine from Siemens. As with most of the European UNIX network the German part of EUnet is mainly based on the X.25 PPSDN. The remaining traffic, mainly 1200 and 2400 Baud, is carried by the PSTN. A leased line to the central EUnet node in Amsterdam is in the process of being ordered. On this we expect to run IP on top of X.25.

Tariffs

Since the foundation of the German EUnet membership of the German Unix User Group has been made an mandatory to be able to use the services of the German EUnet. Most of the original sites had been universities, but now one about a quarter of the 180 institutional members belong to the academic community. To promote network access by students and other academics it was decided to have a special university tariff: 20 DM per month for the e-mail-access of a whole university and 50 DM for the supply of all

newsgroups. To get a fair picture of these prices: 20 DM is about the monthly price for a Telephone by the German PTT - for socially handicapped people like students.

Companies have to pay four times this much: 80 DM for the E-mail-access plus European newsgroups and 400 DM for all international newsgroups or part of them. Note that no alternative to the somewhat "high" commercial tariff has been provided for individuals. Maybe this is a sign of the quick evolution of Unix. In 1984 nobody would have imagined that a single person would be interested to link to EUnet with their private Unix or Unix-like PC, interested to be reached by a mail-address included in the world map and interested in participation of EUnet/Usenet though a decentralised News-Feeder.

This had provoked some problems for the German EUnet as there were only two regional secondary-feeders, one at *tub* at the university of Berlin and one at the company "Siemens" in Munich (only X.25), that are willing to give some of the necessary support and administration.

In early summer discussions started with a group of 50 to 100(?) individuals, students and hackers for the most, who had hooked together their Unix-hosts to a network called "Subnet", most of them linked to the EUnet/Usenet by official EUnet-hosts in the form of Bulletin Boards or Public Access Boxes. Parts of them have an active interest to participate officially in EUnet services (and the UUCP-maps) at "fair prices" below that of companies.

Action was been taken during the summer, mainly by the postmaster-team to develop a new network topology and pricing scheme that would fit to the interests of individuals and companies who participate in the international newsgroups. Maybe some sort of mixture between the old fixed-price and a volume-tariff for the newsgroups, with a more flexible tariff to let individuals participate legally by second-feeders, e.g. by Bulletin Boards. I'm tired of talking about these things, probably like a lot of formerly enthusiastic users. Change seems to be blocked by the bulky limits of the official university decision-makers. Although the new concepts offers many more promising details I'm not sure that it will soon be realised.

The only thing I'm sure about is: If other national backbones don't care about reasonable contributions and flexible integration of such phenomena as these subnets (coming out in Italian and Swiss, too) and let "all" EUnet users share the costs and advantages of the News, they will get the same troubles we had and are still facing. If they don't then with the coming of trailblazers and cheap company links to US, the News might perhaps easily be shipped over the Atlantic, causing confusion to the steady growth of the well-organised EUnet. And I can assure you that it's difficult to persuade a "Subnetter" to look beyond the horizon of his own screen and pay for the advantages of a EUnet-connection as soon as he gets the News without cost...

Challenges for the Future

For the German EUnet there are still more problems arising by the boundaries of the university. For example there seems to be no way to get network accounting information for individuals, who are interested in paying for their mail-access, directly into accounting-software of the University of Dortmund. The administration of this university is unwilling to take the financial risk for individual users in an international (maybe Megabytes of US-mail) network world. A new leased line from *unido* to *cwi* has been put on ice all through the summer by the university administration, unwilling to decide on sums in these financial "heights". These are evident signs to the postmaster-team that another administrative form, for example a non-profit organisation outside the fearful grip of the university bureaucracy has to be found to deal with the still growing demands of an international network.

More bad news?

Yes. Dnet wants to introduce the domain-addressing scheme with ".de" this year. (I don't believe it until it happens, already having announced it for the summer!) CSnet will hand over the administration of .de to the EUnet-Backbone, in cooperation with the German EARN. Suppose the domain-address were introduced: If you saw an e-mail from ag@laura.irb.informatik.uni-dortmund.de you would be able to expect that there is something from a user, named ag, from host "laura", working in the sub-department IRB (which is some sort of computing service-center in the computer science department) of a department "Informatik" of the university of Dortmund, situated in Germany.

You wouldn't like to know?

I don't like to getting my fingers round these fussy addresses, either. Therefore, I remain:

Anke
anke@unido.uucp

The C and UNIX Dictionary

Mike Bayliss
mike@siesoft.uucp

Systems Development Group
Siemens Ltd.

The C and UNIX Dictionary Kaare Christian, Published by John Wiley & Sons, 1988, ISBN 0-471-60931-5 Price (UK) £12.95, (US) $16.95, Soft Back, 216pp.

In the good old days of UNIX it really was possible to find all the answers in the manual (or the source). Unfortunately the system is just too large now, and besides, do you really expect AT&T manuals to describe BSD features?

For those of us who have been working with UNIX for some time there has never really been a problem since we have gradually filled up our bookshelves with new manual editions, conference proceedings and books. However, for a newcomer to the UNIX world the situation must seem hopeless.

Consider the poor user who heard that the system had crashed. His first reaction was "*Did anybody get hurt?*". When the laughter subsided, he was advised to read the manual to find out what a crash was. He was later seen staring at the manual page for *crash*(8V), which contains the following synopsis: *crash — what happens when the system crashes.*

His chosen introductory text can tell him how to use his current system, but it certainly won't help him understand a typical UNIX discussion.

The C and UNIX Dictionary is a lot more helpful, defining a crash as *an unexpected interruption of service*, which while not telling you what the console messages are is the sort of information that most users want.

The dictionary contains an impressive number of definitions, covering almost all of the UNIX and C *buzzwords* and answering most of the questions asked by newcomers.

The range of definitions is truly comprehensive, and accurate with examples in most places where they are useful. The dictionary does apply to both AT&T and BSD systems and is very careful about points that differ between the systems (I spotted only two errors where an example assumes a BSD system without saying so).

Unfortunately, the cross references are at times incorrect and two pages from *set uid mode* to *shar file* contain an annoying number of typos that really should have been noticed before printing.

However, these minor blemishes should not stop you consulting this book. For somebody new to UNIX, especially users coming from non-UNIX systems, it really is as useful as having an expert next to you.

Hungarian Unix Users Group

Sándor Keresztély

Hungarian Academy of Sciences
H-1502 Budapest 112, P.O. Box 63
Hungary

The Hungarian Unix User Group has been founded under the sponsorship of the John von Neumann Computer Science Society of Hungary. It is currently made up of 11 institutional members.

Its general aims are the same as those of the EUUG, i.e. to promote the UNIX culture, to exchange and communicate ideas, to further develop UNIX related issues, and to help and coordinate the marketing of all newly developed software applications which are UNIX based.

Since becoming a part of the EUUG, a one year working plan has been agreed. This includes the first Hungarian UNIX User Conference which is to be held in September 1989.

The HUUG firmly believes that one of the important keys to scientific, technical and social progress is the free exchange of programs, ideas, and information among specialists with common interests.

HUUG therefore regards joining EUnet as its top priority. The technical problems have been solved by our members – expect to hear from us soon!

Chairman: Dr Elôd Knuth
Computer and Automation Institute
Hungarian Academy of Sciences
H-1502 Budapest 112, P.O. Box 63
Hungary
Telex:
Phone:
Fax:

Secretariat: Mrs Mória Tóth
John von Neumann Computer Science Society
H-1368 Budapest 5, P.O. Box 240
Hungary
Telex:
Phone:

Yugoslav UNIX Users Group

Milan Palian
mpalian@idc.idcyuug.uucp

Iskra Delta Computers

In Yugoslavia we are forming a National UNIX Users Group. This article is intended to provide some information about our activities.

History

Several previous attempts at forming the group failed when it became apparent that this required a considerable amount of time and effort. At this time the UNIX community in Yugoslavia is small, scattered and multilingual, making such efforts even more complicated.

Meetings

Our first meeting was held in May 1988. Over 180 people were present. The most controversial issue was whether we are users of Unix or of the applications built on top of it. As both sides won the argument, we felt free to proceed with renewed confidence.

The primary goal of the groups is to provide a meeting place for UNIX users, as well as e-mail and to facilitate access to public domain software. The group is established as a non-profit making organisation. The address and the backbone site is provided by the "Fakulteta za elektrotehniko in racunalnistvo" if the University in Ljubljane, FE for short.

By the next meeting, held in June 1988, a machine had been donated for the backbone and the following duties were assigned:

Organising Committee:

Andrej Kuscer, Hermes
Miran Zrimex, FE
Milan Palian, Iskra Delta Computers

Network Administrator:

Leon Mlakar, FE

The organising Committee was given the task of setting up the network, contacting the EUUG for affiliation, and preparing the next general meeting.

Current Activities

Our next meeting is to be held in December 1988, where our photocopy of the NLUUG constitution is to be accepted by general acclamation. A number of suitable titles will be bestowed, hopefully making us eligible for affiliation with the EUUG.

At the moment 22 users have contributed a symbolic initial membership fee. Many seem to be waiting to see whether anything will come of it before joining. It is expected that EUUG affiliation and access to EUnet should do the trick.

Contacts

The official address of YUUG is:

YUUG
Fakulteta za elektrotehniko in racunalnistvo
Trzaska 25
61000 Ljubljana
Yugoslavia

For more information contact:

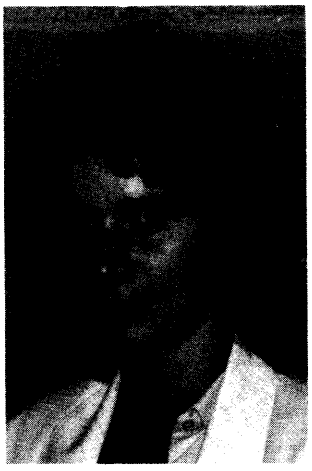
Milan Palian
Iskra Delta Computers
Stegne 15c
61000 Ljubljana
Jugoslavija

Tel: +38 61 574 554
Fax: +38 61 553 261
Telex: 31366 YU DELTA
Net: mpalian@idc.idcyuug.uucp

News from the Netherlands

Frances Brazier
frances@psy.vu.nl

Department of Cognitive Psychology,
Vrije Universiteit,
de Boelelaan 1115,
1081 HV Amsterdam.
The Netherlands
+31 20 5483868 X 3885



I do have a master's degree in Mathematics and Computer Science, I have been working at the Department of Cognitive Psychology for the past 7 years, and yes I do do research, human-machine interfaces and information retrieval being my major fields of interest.

Members

The number of members of the NLUUG is steadily increasing. This year's membership shows an increase of over 20%. We now have 69 academic members, 154 industrial members and 29 individual members (employed by one of the above mentioned members).

Backbone

The NLUUG has been working hard to provide its members with their own backbone and the proper organisation to support the backbone: until only recently mcvax supported both the function of European backbone and national backbone. This all has been done in full cooperation with and with full support of Piet Beertema and Daniel Karrenberg (mcvax). Some problems seem to be dooming with the Dutch branch of EARN - SURF, but we hope to have them solved within a reasonable period of time.

Autumn Conference - November 10th

The autumn conference to be held on November 10th promises to be worthwhile. Contrary to tradition a number of non-Dutch speakers have been invited to share their ideas and beliefs with the audience, complementing the Dutch expertise available. To increase the likelihood of discussion and comparability all speakers have been asked to address a number of the same questions, all concerning the main theme of the conference: *standards*. To give the reader an idea of the types of questions posed: (1) What type of standard is the standard in question?, (2) How standard is the standard, (3) How standard is the standard in relation to UNIX as a whole?, (4) What is the exact relation between the standard and the other standards?, and (5) How has the standard come to develop (its evolution), what position does it take in the field at the moment and what are its future prospects?.

All speakers have been asked to submit a paper or at least a copy of their slides for the Conference Notes which will be printed for the first time.

The abstracts below provide an impression of the talks that will be given on November 10th.

System V R4: Bob Duncanson, AT&T UNIX Europe

The contents of this session will be a surprise for all!

BSD 4.4: Keith Bostic, University of California at Berkeley, USA

In the absence of any generally accepted standards in the UNIX industry, 4BSD and System V have traditionally been the only "official" standards. With the advent of POSIX, this is no longer true, and with an increasing demand for standard interfaces, a new approach to designing computer environments is necessary. This talk will provide a brief overview of Berkeley's current and future position in the UNIX marketplace and its role in standardisation efforts. It will also discuss concerns such as the relationship between standards and future research, and the necessity for open systems.

POSIX/ANSI-C: Ed Keizer, Vrije Universiteit, Amsterdam, NL

It seems as though both ANSI C and IEEE POSIX P1003.1 will become generally accepted standards. The C standard provides a precise definition of the constructions possible in C and their interpretation, together with a description of the routines entwined with C, such as 'printf' and others from the standard IO library.

A short history of the C standard and of the progress of the process of standardisation, will be presented. In addition the major lines of difference between this standard and the standard defined in "Kernighan and Ritchie" will be discussed.

The purpose of IEEE POSIX is to define an interface and environment for a UNIX based operating system to provide portability at source code level. The first thing IEEE has done is to define a standard (P1003.1) to describe the concepts and interface at the 'kernel' level. In the future other standards will be defined for, for example, network protocols.

Both an overview of the areas in which POSIX is involved and the P1003.1 standard itself will be given, followed by, a few words on the role of the Netherlands in the standardisation process.

Open Look: Bertram le Duc, Sun Microsystems Nederland BV, NL

Open Look will be introduced as a specification on how a new user interface for applications should be designed. In terms of an application the Unix operating system will be designed with the Open Look specification. At this moment about 60 software vendors have committed themselves to the Open Look standard as their new user interface. The Sunview Window System which is the SUN user interface on our systems, is one of the applications which will make use of the Open Look standard after the merge of BSD4.2 (SUN)S4.1) and AT&T's System V. The first Open Look applications will be based on the graphic libraries included in the UNIX merge mentioned.

OSI/ISO: Peter van Eijk, Technische Universiteit Twente, NL

One of the more important efforts in standardising computer networks is the so-called ISO OSI project. It defines a Reference Model (ISO/RM), which serves as a meta-standard. The 'real' standards are the Service and Protocol standards for each of the layers of the OSI/RM. The OSI/RM and the functionality of each of the layers will be discussed as will the position of UNIX within this framework, e.g. what the relation of the structure of an OSI implementation is to the structure of Unix.

GNU: Richard Stallman, Free Software Foundation, Cambridge, USA

The current status of the GNU operating system project, plans for the future and the relation to POSIX, ANSI C and other standards will be discussed. GNU Emacs, the cooperation between Berkeley and GNU, GAWK, shells, the kernel, the debugger, the C compiler, GNU's C ports, Compiler related programs (C++, assembler, object file utilities, make, C library, profile), the mailer, the window system, other utilities, th documentation system, manuals, funding: all topics to be addressed within this presentation. The importance of the Free Software Foundation, its aims and achievements will not go

unmentioned.

OSF: Gilbert Eloy, OSF Europe

The concept of open systems is one of the most important issues in industry today. Up until now, efforts for the realisation of a truly portable application environment have not yet succeeded. To be successful in the realisation of open systems, we need broad based participation and support by users and producers. In addition an effective decision making procedure is required.

OSF plans to achieve consensus, develop products and create standards will be discussed.

X/Open Technical Overview: Willem Dicou, Philips TDS, Apeldoorn, NL

In the beginning of 1989 the third issue of X/Open's Portability Guide will be published; an important step in the definition of an Open Systems Environment, being X/Open's mission. This brings a complete and comprehensive software environment into sight, enabling the development of application software for a broad platform of hardware architectures, as well as the connection of systems from different suppliers in a simple manner.

Standard activities are, in general, restricted to one single aspect of the total software environment needed to develop applications. X/Open's strategy consists of adopting existing official and de-facto standards, adapting them when necessary to provide integration in the so-called "Open Systems Environment", OSE. To be considered these standards must be supported by existing products so that the OCE concept becomes truly usable.

If no standards yet exist, X/Open supports their attainment, either by initiation or participation in standardisation activities.

Without doubt Unix has been, and still is, the most important initiative of X/Open. The connectivity of X/Open Systems and other systems, in particular mainframes and PC's, is currently being addressed.

In addition to the technical presentations four tutorials will run parallel to the technical sessions, open to all participants:

AWK: Eddo de Groot, Mutad Software, Amsterdam, NL
Shell programming: Chris Oostman, Database Consultants Europe bv, Amsterdam, NL
LAN's on TCP/IP: Hans van Staveren, Vrije Universiteit, Amsterdam, NL
Mail configuration: Henk Hesselink, Associated Computer Experts, Amsterdam, NL

Spring Conference - May 9th

Our next conference will be held on May 9th, 1988 and will focus on ... user-interfaces. Any suggestions for contributions are very welcome.

Rumour

Rumour has it that a crypt competition is to be initiated by the NLUUG (to be supported by the EUUG). More information may follow in the next newsletter.

The Danish Connection

Keld Jörn Simonsen
keld@dkuug.dk

University of Copenhagen
Copenhagen
Denmark

Keld is the chairman of the Danish group & he occasionally writes an article or two for the EUUG Newsletter, where he is also known as one of the 3 pinup great danes from Owles Hall. Currently he is much engaged in the Danish group, the network, and international standards.

Hi there, all members of EUUG,

Time to hear about what is happening up in the small cold country up by the sea. It is a year since we last met here in these columns, and a lot has happened in this time with the Danish group.

About a year ago DKUUG elected a new board, and this has been working hard ever since to produce the best services available for the members... We have grown from about 220 members to 290 in the year so we still have a very large group for such a small and cold country. Well, maybe there is nothing else to do up here...

Forking subcommittees

The first thing the new DKUUG board (with just one new member) did was to fork ourselves into subcommittees: one for the network, one for the newsletter, one for membership services and administration, one for member meetings and one for exchanging DOS software between board members... The forming of subcommittees has sped up the pace of the group. Now we have the same number of board meetings but with a lot of other subcommittee meetings going on, each board member spends much more time on DKUUG matters, adding to the overall quality of the services of the group. We even get some people outside the board to join the subcommittees.

The network

DKUUG took over the Danish EUnet network about a year ago from the Computer Science Institute at University of Copenhagen (DIKU). DIKU wanted us to take over, partly because they did not have the manpower to service all these commercial users, partly because the finances of the network were too unmanageable and risky for the university institute. They still provide us with part time manpower and space for our equipment and staff, according to an agreement of cooperation that we have made.

The DKUUG board then started this new activity by forking a subcommittee for the network with their own budget responsibility. This 'netstyr' subcommittee had to ensure that there be no profit or loss in the running of the network. The netstyr subcommittee consists of 4 members from the board and a couple of backbone managers.

The first thing that we did was to set up correct invoicing and to computerise the production of invoices. The prices were also raised by about 50

% to allow us hire staff. We bought a couple of new 2400 bps modems and got new telephone lines for that. We have borrowed a SUN computer from Ericsson (now Nokia Data), and we moved the Danish backbone to this new 'dkuug' machine. We have acquired new X.25 lines and also a TrailBlazer modem, which has cut our news costs by 80 %. The TrailBlazer can run at a speed above 10.000 bps on a normal telephone line, that is more than 4 times the speed of a 2400 bps modem. We are now connected to the NORDUnet Ethernet, which connects almost all the universities in the Nordic countries via 64 kbps or 128 kbps leased lines, so that a user would think that all these Nordic machines were on the local LAN. This is much like ARPA in the USA, using the same protocols. Many changes have been made to the software to accommodate these new hardware facilities.

We are in the process of setting up two new services, one so that a member can just log into our machine and then read news and mail, and also send something the other way. The other is an enhanced archive service: we are about to buy a 382 Mb hard disk on which we will keep a lot of free software, mainly the EUUG tapes. The archive should be available both to ordinary Danish EUnet sites and also for the new login service people. Both the login service and the archive service are to be run on a 3B2/400 machine on loan from Olivetti Denmark.

We have made several formal agreements with other parties: The agreement on cooperation with DIKU, another agreement about regional news distribution with the Computer Science Institute at Aarhus University, loan agreements with Nokia and Olivetti, administration agreements, and some agreements with other networks in Denmark as mentioned below. To come are some agreements with NORDUnet and EUnet.

DKUUG registered the ".dk" domain with SRI NIC in USA and we then made a "name agreement" with some other academic oriented networks in Denmark, including EARN and the X.400 based R&D MHS, such that they could use the ".dk" domain as well. DKUUG is the official administrator, but what really is going on is that each of the administrators of these networks sends the other network administrators a list of the names of sites on their national network. There is no indication in the name of the site about which network they are actually on (in some cases they are indeed on several, but with the same name). The domain name just before ".dk" is to be something representing the firm or institute, not containing the name of a dog, old Nordic or Greek gods or name of operating system or machine type. The networks refund each other the costs they have on behalf of the other networks, eg. incoming USA mail coming thru EUnet to an EARN site.

Overall the network has grown in a year from 43 on mail and 5 on news to 70 on mail and 20 on news.

Having the network managed by an independent subcommittee of the user group board has proven to be very efficient. I wonder how many of the decisions and agreements we have made, which would have materialised if it still had to be managed by the computer science institute. Effectively the subcommittee decides everything by themselves, but has a legal body to lean on in the form of the user group, along with a place to look for liquidity...

The Danish UNIX catalogue

DKUUG affiliated to the /usr/group association at the start of 1988, and we are now trying to sell their catalogue.

Talking about catalogues, we have made our own: the Danish UNIX market overview. It is already in its second edition, with a quite fresh issue from late September. It contains information on nearly 1800 UNIX related products sold in Denmark by nearly 100 vendors. Compared to the huge /usr/group 1988 catalogue having about 4300 products we think we have a very thorough coverage of the Danish market. We had to nag people a couple of times on the phone to get this coverage! It was free of charge for a vendor to be represented, and vendors who are not members of DKUUG were also allowed to be there, to get the widest possible coverage.

The catalogue has information on hardware, software, services, training and literature. Well, our catalogue is not as big and glossy as the /usr/group one. It just contains a list of the products and the vendor's name and telephone number and possibly which machine or operating system it is intended for. There is a list sorted by the product name, and a list sorted by product group (which is the same as /usr/group uses), and at last there is a list for each vendor showing what they sell in each product group. This information

is not a lot, but it will tell people who, in Denmark, is selling a specific product they know from foreign literature, and it will point them to Danish vendors selling products in the product category they need, and also tell them what other kinds of service they can expect from the vendor. If costumers then want more information they could look up the product in the /usr/group catalogue, or just phone the vendor.

Member meetings

In the last year we have had 5 member meetings: the yearly meeting with about 40 people attending, a meeting on UNIX and OS/2 and DOS and realtime with 65 attendees, a meeting on user experiences with converting to UNIX with about 60 attending, UNIX standards drew about 115 people, and a combined network meeting with EARN and X.400 got 80 interested people. We have throughout the year improved the quality of the meetings considerably, with good abstracts on the talks, advertisements in the computer press and good registration procedures. People seemed to be quite interested and a lot of good discussion goes on at the meetings. The food has improved too...

Sundries

Other activities have involved our business administration, with a new contract on proposal, 3 issues of the newsletter, a booth at one of the big computer exhibitions, 3 telephone answering machines, a new tape copying service hopefully faster than EUUGs, participation in standards activities with proposals for C and POSIX getting rid of Danish (and other European) problems with non-English characters...

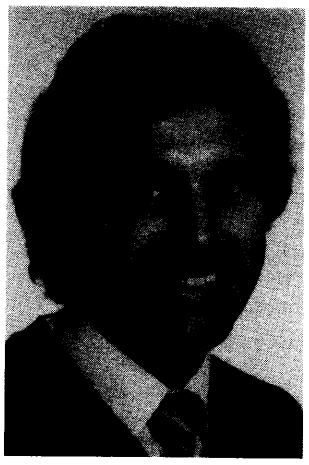
The next event is the yearly meeting on the 23rd of November and a seminar on the network and its new services also about that time. The board has been working hard and some of its members are leaving it at the general assembly. We thus have to find some new hard working people for the group. One of the big issues is improving the newsletter, maybe going to an issue a month...

That completes the news from the Danish Group.

The UKUUG National Group

Sunil K Das (Chairman)
sunil@cs.city.ac.uk

City University
Computer Science Department
London EC1V 0HB



Sunil Das is the chairman of the United Kingdom Unix Users Group. He is a lecturer at the City University (London). For many years he has been an active member of the Unix Travel Club.

The highly successful EUUG Spring 1988 Conference was hosted by the UKUUG in London in April, so we did not hold our traditional Summer Technical Meeting in June. However, having been revived by the jaunt to Portugal, we are on the move again. Links with the UKNET backbone site have been fostered and developed and are now stronger than ever. Distribution of UKUUG documentation and UKNET documentation has been centralised at the University of Kent and distributed by Peter Houlder. This new documentation comes in a glossy folder so if any potential or current UKUUG member, or other EUUG member would like a copy, please contact Peter (uknet@ukc.ac.uk) or Bill Barrett at Owles Hall.

To hook into UKNET, one is required to be a UKUUG Institutional Member. The number of academic sites who are part of the network has stabilised in the UK. However, the commercial site activity is on the increase, so the UKUUG membership has improved healthily over the last few months.

Other literature in the form of a one page flyer has been produced by Zdrav Podolski (Treasurer) and Mick Farmer (Secretary) and distributed at trade shows and exhibitions in an attempt to promote our membership numbers. An appointment of a part-time Publicity Officer is under consideration to help with this objective.

The three executive officers can now be reached using the address ukuug-exec@ukc.ac.uk. The ukuug@ukc.ac.uk address still reaches the executive, but includes the eight or so Advisory Board.

The next event on the calendar is the 1988 Winter Technical Meeting which is to be devoted to UKUUG and UKNET presentations over the two days 19th - 20th December at the University of Kent at Canterbury. We will be pleased to welcome Daniel Karrenberg as our overseas speaker on "OSI Transition Plans of EUNET". A new departure for UKUUG is an afternoon devoted to a tutorial on UKUUCP, what it is and how it relates to other UUCPs. UKUUG, EUUG, /usr/group/UK and USENIX members qualify for a discounted entrance fee. Full details of the programme can be found at the end of this article.

Another UKUUG initiative is to formalise the activities of our local UUGs. The officers of the UKUUG have been looking at ways of adding to the value of UKUUG membership. One idea

grew out of the habit of UNIX users in the London and South East of England meeting on the last Thursday of every month (except December) in a London pub to have a pint, a chat and a curry or Chinese afterwards.

Our thought was to extend this social activity by including a lecture / tutorial / discussion session before hand. There are many subjects which could usefully be covered in this way, filling a gap between the informal LUUG sessions and the more formal (and infrequent) EUUG and UKUUG conferences. Consideration was given to:

Venue	near to the pub
Subjects	have to be interesting
Timing	people have to have time to finish work and get there
Publicity	what do you think this article is about? :-)
Costs	hire of venue, speaker expenses, administration and publicity; UKUUG will bear these
An Organiser	naturally the UKUUG Executive could not do everything, so we have been lucky to find a willing helper in Andrew.Findlay@brunel.ac.uk.

Relations with /usr/user/UK although good have been dormant in recent months. The major stumbling block to merging proved to be the EUUG affiliation fee, as we reported at the last EUUG Governing Board Meeting.

Finally, on my return from the EUUG conference in October, I was requested to give a talk to 200 AIX techies at IBM in Hampshire. My talk centred around UNIX in the Universities, in Networking and in User Groups. Naturally, they have an intense interest in OSF and were therefore keen to learn that EUUGs policy is not to align with OSF or AT&T/SUN but to influence matters from a detached standpoint. IBM UK stated their intention to join the UKUUG, which we heartily encourage.

Stop Press

The LUUG on the evening of 20th October proved highly successful. We had some 30+ attendees and the talk went down well. One attendee came from Nottingham and two came from Reading, so they came from far and wide. Even Lee ventured out of IC :-).

The usual drinks session followed and a group of LUUGers who went to the pub only, were somewhat surprised to see the influx at 7:30 pm. Some of them had been to a rival meeting held by someone named AT&T on V.4 (I understand it was a sales pitch :-).

For the London EUUG Conference, Sunil (UKUUG) had 10 sweatshirts made displaying the V6 comment:

```
"You are not expected
  to understand this!"
```

The bang was Sunil's so he wouldn't be sued by AT&T for exhibiting their source code :-).

Sweatshirts went to Dennis Ritchie and John Lions, and 5 EU/UKUUG officers associated with running the conference. Unfortunately, the 3 intended for Ken Thompson and others went "missing" from the conference office, even with QE II security.

So since they can never be worn at a UNIX conference without being spotted, Sunil would greatly appreciate receiving them back by post with no names or questions asked.

Programme for the UKUUG Winter Technical Meeting

This is a 2 day event. The conference session starts after lunch on Monday, and ends with lunch on Tuesday. The tutorial is on Tuesday afternoon. To book please contact Owles Hall, admission to the conference is £35, and £39 (member rates).

Mon 19th December 1988

UKUUG Welcome Chat – Mick Farmer (bbkcs)
A low cost bitmapped terminal on the Atari ST – Peter Collinson (ukc)
Mail interface enhancements to MH – Donal Daly (tcdcs)
Developing and Adapting UNIX Tools for Workstations – David Barnes, Mark Russell & Mark Wheadon (ukc)
Direct Manipulation Tools for UNIX Workstations – John Bovey, Mark Russell & Olav Folkestad (ukc)
Installing and Operating NFS on a 4.2 BSD VAX – Jim Reid (strath-cs)
Process Management in STIX – Aaron Gull (citycs)
NeWS and X, Beauty and the Beast? – W.T.Robberts, A.Davison, K.Drake, C.Hyde & M.Slater (qmc-cs)
UKUUG Business Meeting + Election of Officers
Dinner

Tue 20th December 1988

Uknet Welcome Chat – Peter Collinson (ukc)

Uknet Overview – what UKnet is, national and worldwide links on-line information services, news, email, authorisation and charging. Also a general overview of mail systems and transport services. – Peter Houlder (ukc)
UK Coloured Books – Current facilities and transition plans – Prof P.F.Linington (ukc)
EARN and BITNET – Paul Bryant (rutherford)
News – present and future developments – Chris Downey (ukc)
OSI transition plans of EUnet and other interesting developments – Daniel Karrenberg (eunet)
UKUUCP Tutorial (includes printed notes).

EUNET® in Belgium

Marc Nyssen
(Secretary of the BUUG)
marc@vub.uucp

Vrije Universiteit Brussel
Medische Informatica

Marc Nyssen is Associate Professor at the Medical Informatics Dept., Vrije Universiteit Brussel, Belgium. Since 1978, he has been an enthusiastic UNIX™ user and as colleague of Erik Blockeel, the software specialist who introduced UNIX in Brussels in 1978. He works mainly on biomedical applications ranging from data-acquisition to networking. In 1986 he became a co-founder of the BUUG, the non-profit association of UNIX users in Belgium. In collaboration with AT Computing, he teaches UNIX courses at the V.U.B.

Introduction

In Europe many different institutions have set up long-haul networks, connecting computers or local networks.

To manage, coordinate and interlink these different networks, several initiatives were taken, some of which have resulted in operational gateways. In our experience, and according to the documentation that we have, only two really international logical computer networks are now in operation (and open to a wide user-community) in Europe at this moment: EUnet and EARN.

While EARN is mainly limited to one vendor (IBM) and to academic institutions, EUnet is an open cooperative network and although all important sites run the UNIX operating system, PC's can be connected via local networks.

Through gateways, the long-range international networks are mutually interlinked in several sites and they are connected to smaller networks (national nets) and their "counterparts" in North America, Australia and Asia.

Besides the logically organised long-haul networks, the recent generalisation of X.25-networks all over Europe has greatly enhanced the physical links between machines and the access to publicly or privately managed database systems.

EUnet in Belgium

The V.U.B. was the first Belgian site connected to EUnet, back in 1983. With a simple 300bps modem we were dialed-in daily by "mcvax" over the telephone modem line, installed to be used by DEC Field Service.

Since 1985 the Philips Research Laboratory Brussels in Watermael Boitsfort, with the sitename "prlb2", has full filled the role of Belgian Backbone. Site managers Michel Lacroix and Jean-Jacques Quisquater have set up many national and international links, much to the benefit of the Belgian EUnet users. Lacroix is involved as "network officer" in the BUUG. It should be stressed that EUnet (as the other UUCP- and USENET networks) is mainly operated on a voluntary basis for the time being. The efforts of these pioneers cannot be appreciated enough!

Under their coordination, the X-25 network DCS was chosen as physical support. DCS proved to be a major improvement over telephone lines and modems, regarding speed, reliability and cost-effectiveness (in contrast with other countries!). The network has grown steadily and the table at the end of this article shows which sites are connected at present (October 1988).

® EUnet is a registered trademark of EUUG.

Most of the Belgian sites have X-25 connections (PTT DCS service), with local PAD's connected to serial ports of their UNIX machines of various vendors.

An important development in Belgium this year has been the gateway between EUnet and EARN, established in April by the site "kul-cs". Also, on the "political level" a working group consisting of EUnet and of EARN users are working out the establishment of a common top-level domain ".be", comprising both networks.

Back to some statistics: according to prlb2 representative Daniel Wybaux, between January 21st and September 13th, 1987 prlb2 received 9588 messages and sent 7961 messages to other sites in Belgium.

During the months June, July and August, 233 megabytes were exchanged between prlb2 and Belgian sites, 130 megabytes passed our borders via the backbone.

As of July 1988 there were 20 sites on the Belgium network consisting of an estimated 100 UNIX machines. This compares with figures of 11 and 29 respectively for April 1987.

During 1987, the BUUG and representative users and managers of EARN in Belgium, formed a joint committee to discuss the establishment of a top-level Belgian network domain, with a common name-space for EUnet and EARN. These discussions have lead to a report by BUUG "special negotiator" Prof. Pierre Verbaeten. This report was accepted by the BUUG Board in April 1988 and by the EARN committee about the same time. An official request for the establishment of the ".be" domain was sent to the Network Information Centre at SRI. We were prompted for further information and some more "official" backing of this request, which was readily obtained by Pierre Verbaeten. The National PTT stated that it was not interested in networking other than X-400 and thus backed our request.

The ".be" top-level domain will get an ".ac" academic subdomain which will encompass the actual "EARN" sites and the present academic "UUCP" sites. To illustrate this: my Email address probably will become:

marc@minf.vub.ac.be

which quite clearly situates me in this "domain" world.

A second recent development in Belgium is the "saturation" of the backbone due to the ever increasing demand for connections. The BUUG is looking for solutions such as alleviating the backbone workload by setting up "secondary" backbones, to which new sites will be connected. Moreover, Philips Research will start invoicing the connected sites for their share of the traffic costs. The exact terms and modalities are still quite controversial, although BUUG has agreed upon the principle of a reasonable participation in costs.
Finally, BUUG is starting to look much more actively for industrial support (hardware, services and sponsoring) of EUnet in Belgium. It is our aim to run the network as "professional" service, unfortunately, the total mass of some 20 sites is as yet not sufficient to support full time person, but the support work is becoming too big to continue on an ad-hoc basis.

Conclusion

In my opinion, the network in Belgium has reached a critical stage in its development, mainly thanks to the exponentially growing interest in UNIX. In the near future, we will be able to make it a professionally run service of the BUUG. Right now, we are trying to build up sufficient momentum to reach this goal. Although the number of "talkers" is greater than the number of "doers" (typical for WAN's ??) I expect a major breakthrough! The wide interest beyond the "small club" of initiators of EUnet in Belgium is quite encouraging.

Personally, I am convinced that many European institutions and firms are becoming aware that EUnet has tremendous potentials and that using it and sponsoring (especially by UNIX vendors) will lead to both immediate and long term benefits.

Acknowledgements

This article was realised thanks to several people: Michel Lacroix, Vincent Marlair and Daniel Wybaux provided the prlb2 statistics, Pierre Verbaeten reported on the latest ".be" news and Erik Blockeel ran the computer.

EUnet sites in Belgium (July 1988)		
site name	Organisation	Town
agrib	CEC	Brussels
atmos	Belgian Inst. for Space Aeronomy	Brussels
bbri	Belgian Building Research Institute	Limelette
cstb	MBLE Philips Software Technology Centre	Brussels
fun-cs	Facultés N.D. Namur, Computer Sci. dept.	Namur
imec	Interuniv. MicroElectronics Centre	Leuven
intgrb	Intergraph BV	Brussels
kulcs	Kath. Univ. Leuven, Computer Science Dept.	Leuven
kulet	K.U.L.-Eurotra	Leuven
lln-cs	Univ. Cath. de Louvain, Computer Sci. Dept.	Louvain-la-Neuve
prlb2	Philips Research Laboratory Brussels	Bruxelles
rubens	DGS / UCMB Univ. Libre de Bruxelles	Bruxelles
rug	Rijksuniversiteit Gent	Gent
siebru	Siemens Brussels (Dept. VD43)	Brussels
sunbim	Belgian Institute for Management	Everberg
syteke	Sytek Inc.	Brussels
taibru	CEC - Termin. et Applic. Informatiques	Brussels
uclfynu	UCL Nuclear Physics	Louvain-la-Neuve
uclmnuc	Univ.Catholique de Louvain, Nuclear Med. Dept.	Louvain-la-Neuve
uiag	Univers. Inst. Antwerpen	Wilrijk
vub	Vrije Universiteit Brussel	Brussel

EUUG Executive Committee Report

Helen Gibbons
euug@inset.co.uk

Secretary to the committee

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

As we are approaching the new year it seems a good time to remind our members, especially those who do not regularly attend EUUG "get-togethers" at conferences, of how the EUUG actually functions.

As a large European Group, with a membership of over three thousand, made up of all the members of all the affiliated national groups, it requires not only a lot administration, but also a great deal of policy making, decisions and hard work.

The day to day administration is of course carried out from the EUUG central office at Owles Hall in the UK, but that is the lowest link in the chain. The top link in the chain of administration is actually the **Governing Board** Which is made up of representatives from each national group in membership. The Governing Board sits twice a year, once in the Spring and once in the Autumn, to monitor what is happening within the group and to set down policies, both general and financial, and to pass these, in the form of actions, to the middle link in the chain, which is the **Executive Committee**.

The Executive Committee fulfills all the functions of a general manager within a company. It sees that things **do** get done and it advises the Governing Board on what **ought** to be done.

It meets several times a year and carries a very heavy workload.

Under the Chairmanship of Teus Hagen, who is the Chairman of the EUUG, the Executive members serving at present are Michel Gien as Vice Chairman, Neil Todd and Ernst Janich covering conferences, Nigel Martin covering finance, Daniel Karrenberg covering the network, Philip Peake covering publications and Kim Biel-Nielsen covering Public Relations.

One of their most recent decisions was to keep the membership advised of the activities of the Executive Committee and this is, therefore, the first of a regular column updating all our members on committee proceedings. Comments are welcome and should be addressed to Helen Gibbons at the EUUG Secretariat.

The last Executive Committee meeting was held on 1st October, just prior to the Portugal Conference, which we hope everyone enjoyed. Work which the committee still has in hand from previous meetings includes the preparation of an historical overview of the EUUG, the computerisation of conferences and accounts, the widening of the reader base for the newsletter by arranging distribution to European libraries, the production of a glossy brochure on EUnet and the preparation of an Electronic Mail Directory.

An application was received from the Hungarian UNIX User Group. This was subsequently agreed by the Governing Board and so we are pleased to announce the affiliation of HUUG (Hungarian UNIX systems User Group) represented by Dr. Elod Knuth.

Some time was given to the discussion of a report from Mr. Hagen who had attended a meeting in Luxembourg of the UNIX System V User Forum. Mr. Hagen felt that EUUG should play some part in this since one of the prime roles of EUUG is to act as a forum for all types of user groups on a technical front. Thus the committee agreed to keep a watching brief and also to try to take perhaps a more active role in OSF. Talks with X/Open should also be encouraged.

As always, the Executive spent some time examining the financial situation present and future and from the very detailed accounts presented it was agreed that income was at present at an acceptable level and a reasonable and safe operating amount remained on reserve. Concern was however expressed that some national groups continued to pay their subscriptions late and that this was unfair to those groups who paid on time. The treasurer was therefore engaged in looking into practical ways of charging interest to late payers. Subscriptions will be calculated in ECUS next year.

Please note that as a result of earlier decisions Access (Mastercharge) facilities have now been added to the Barclaycard (VISA) facilities already operated by the Secretariat, and both will in future be available for all conferences. Conferences are always reviewed at Executive Committee meetings. The Spring Conference in London was deemed to be a great success with 460 delegates and should yield a profit once all accounts are settled. The Brussels conference 3-7th April 1989 is well advanced. Other conferences being worked on in the preliminary stages are Vienna 18-22nd September 1989, Munich 23-27th April 1990, Nice 15-19th October 1990 and Norway in Spring 1991.

Mr. Karrenberg gave the committee a very comprehensive report on the EUnet situation which showed that sites in Europe had grown from 978 in 1987 to 1351 in 1988. There are so many activities connected with the network and so much that it would be nice to do, that Daniel cannot possibly do it all on his own. He is seeking volunteers to help. Anyone with some time to spare and a real interest in the network should contact him on dkf@cwi.nl

The next meeting of the Executive Committee will be held on 12th December when more time will be given to the subject of public relations and liaison with other groups. Watch this spot for further reports.

STOP PRESS - STOP PRESS

BRUSSELS CONFERENCE TUTORIALS

The programme of technical tutorials for the EUUG Brussels Conference was announced too late to be included with the main conference announcement (see earlier in the newsletter). The current tutorial titles are as follows:-

Thursday 6th April 1989

- Programming in ANSI C
- Writing System V device drivers
- Open Network Computing and NFS
- Xlib

Friday 7th April 1989

- System V Administration
- Internationalisation
- UUCP and Sendmail Configuration
- Open Look

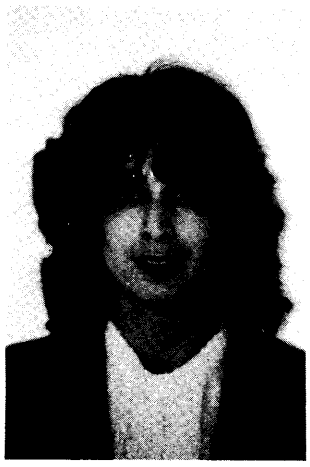
Early Booking Date

To qualify for the "Early booking discount" your booking form must be returned to the Secretariat at Owles Hall no later than **28th February 1989.** Places on the tutorials are strictly limited – a double reason to book early.

USENIX Association News for EUUG Members

Donnalyn Frey
donnalyn@uunet.UU.NET
donnalyn@uunet.uucp

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.

C++ Conference

The USENIX Association's First Annual C++ Conference, held October 17-20 in Denver, Colorado, was a success. Over 500 people attended the conference. The first two days were devoted to tutorials, with the next two days for technical sessions. A special implementor's workshop was held on Friday.

Bill Joy, Vice-President of Research and Development at Sun Microsystems, presented the Keynote Address. Mr. Joy spoke on the Programmer Productivity Crisis in UNIX. He began by noting that "The computing community is faced with an enormous problem in programmer productivity in that we have faster machines but that programmers are still at the same productivity level." In addition, he said, new computer hardware can be built faster, but the software still takes the same development time. He noted that "the challenge is to improve programmer productivity to take advantage of the newer, faster computers."

Mr. Joy went on to say that the solution to this challenge is developing newer languages to enhance programmer productivity, languages such as C++. He said that "the name of the answer to the problem is C++," not only because of its technical merit, but "because it is acceptable to our management." Mr. Joy noted that "the language is not simple... but it supports multiple programming styles." He then predicted that C++ will be very successful in the future.

Mr. Joy then went on to discuss UNIX, stating that people are "no longer arguing if UNIX, but which UNIX." He said that his team on the Menlo Park project is rewriting UNIX for the future, using a subset of the C++ language.

To request copies of the Denver C++ Conference Proceedings, send email to {ucbvax,uunet}!usenix!office.

UUNET Communications Services

Computer Upgrade

The UUNET Communications Services has upgraded its computer to meet the growing needs of its subscribers. UUNET had been a 14 processor Sequent Balance B21. However, UUNET's rapidly increasing subscriber base required a more powerful computer to meet future needs.

Operationally, UUNET now consists of a 4 processor (Intel 80386) Sequent Symmetry S81

computer with 28 dialup modems (16 accessible via an 800 number), over one gigabyte of disk space, a T-1 connection to US Sprint, and a 56 KBPS dedicated connection to Tymnet.

The system will continue to be upgraded to meet the needs of its subscribers. The system can handle up to 4,000 subscribers with the addition of processors and memory and additional disk space as required.

The FaceSaver Project

The FaceSaver project was not continued by the Board of Directors at their last meeting in October. It will not be available at the San Diego Winter 1989 conference due to an apparent lack of interest by conference attendees in the past. The database of faces and identifying information may be placed on the UUNET computer sometime in the future.

Winter 1989 San Diego Conference

The next USENIX Association technical conference is scheduled for January 30 - February 3 in San Diego, California at the Town & Country Hotel. Tutorials will be given on Monday and Tuesday, January 30 - 31. The technical sessions will be presented Wednesday, Thursday, and Friday, February 1 - 3. The keynote speaker will be William T. O'Shea, Vice-President of Product Development at AT&T.

Tutorials will be presented on many subjects, including new tutorials on 4BSD TCP/IP Performance Improvements, Open Systems Interconnection (OSI), Security Issues in a Distributed UNIX Environment, Using Postscript as Yet Another UNIX Tool, Network Computing System and Architecture, and Object Oriented Design on UNIX.

Papers will be presented on distributed systems, file systems, operating systems, window systems, internetworking, objects and memory, processes, security, and two special interest sessions, including a paper discussing the birth, life, and death of a UNIX virus. A Work in Progress session will be held on Thursday.

For full information on attending the San Diego USENIX Association conference, contact the USENIX conference office at P.O. Box 385, Sunset Beach, CA 90742, USA and request a copy of the conference registration information. You can also request the registration information by sending email to {ucbvax,uunet}!usenix!office.

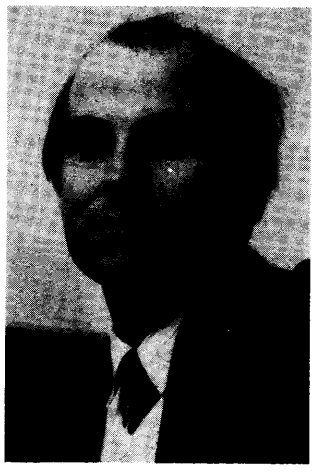
Further Information on Conferences and Workshops

If you need further information on upcoming annual USENIX Association conferences or workshops, contact the USENIX conference office at P.O. Box 385, Sunset Beach, CA 90742 USA. 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 annual C++ conference and the semi-annual technical conferences.

SVR4 Conference (London)

Dominic Dunlop
domo@sphinx.co.uk

Sphinx Ltd.
Maidenhead
United Kingdom



Dominic Dunlop is the Research and Development director of Sphinx Ltd, a UK software distribution and services company he co-founded in 1983, after experience in supporting Zilog's range of super-micro computers. Sphinx centers its operations around non-proprietary operating environments, selling in a variety of third-party and self-written software products across hardware from name different vendors.

Dominic's current role is that of bringing complex new products into Sphinx' offering by first understanding the technical and marketing issues involved, then working to address them in the context of the company's current capabilities and activities.

"Crafted with pride in the USA" reads the label inside the red flight bag too small to hold the nine kilos of paper handed out to each of the three hundred delegates at the *System V, Release 4, Developer's Conference* held in London at the end of October. SVR4 is still in the process of being crafted with pride in various parts of the USA, and won't be available in source or binary form until the fourth quarter of next year. In the mean time, AT&T and Sun Microsystems are distributing around fifteen tons of paper among the nearly two thousand developers prepared to spend five hundred pounds and three days to learn about next year's model of the UNIX operating system at one of eight conferences held around the world.

Next year's model looks good — and it looks big. A clue can be found in the four *Migration Guides* handed out at the conference. Surprisingly, the guide for UNIX System V users, at 150 pages, is almost as long as the guides for BSD, SunOS, and XENIX put together! Why? Because SVR4 represents the long-awaited re-unification of the AT&T and Berkeley (latterly Sun) variants of UNIX. BSD being the richer of the two environments, its users need to be told less about the characteristics of System V than do System V users of the extra facilities available to them, now that the BSD heritage has at last been granted its place out of the Sun, and in the SVID. (Actually, edition 3 of the System V Interface Definition won't appear until a couple of months *after* the SVR4 source tapes, in order that it can accurately describe the software as distributed.)

As well as being SVID-conformant, SVR4 will conform to IEEE 1003.1, the POSIX kernel interface standard, will have a C compiler which implements the recommendations of the ANSI X3J11 working group, and will provide MIT's X Window system, a TCP/IP protocol suite, and Network File System (NFS). Not content to stop at these public or *de-facto* standards, SVR4 will also deliver AT&T's Remote File Sharing (RFS), Sun's Network-extensible Window System (NeWS) plus the new Open Look user interface, and support for Berkeley-style sockets as well as the newer Transmission-Level Interface (TLI). And they'll all be codified in the SVID. Add to this ELF, an Extensible Linking Format, replacing the sadly non-extensible COFF (Common Object File Format), and a raft of Application Binary Interfaces (ABIs), and it's clear that developer's bookshelves around the world will soon be groaning with documentation.

One also suspects that AT&T and Sun are anticipating groans from members of the Open

Software Foundation faced with demands to satisfy customer expectations which have been raised by the pre-announcement of all these SVR4 features. Strangely, the OSF and its relationship with the newly-announced "UNIX System V Industry Association", mentioned briefly in the opening session of the conference, was not a hot topic of conversation among attendees, who were much more interested in programming than in politics.

One thing that will definitely be groaning is the memory of older systems called upon to run the new UNIX operating system. While SVR4 is expected to be bootable on a two megabyte 3B2 (for example), four megabytes will be required if it is to run well. And don't wait for SVR4 for your PC-AT, as many of the new features require a paging virtual memory environment. Most interesting of these is memory-mapping for files, a facility which is pressed into service to provide position-independent sharable libraries. This implementation, adopted from SunOS, is more flexible than the fixed-address shared libraries of System V release 3, and should cut down on the amount of memory and disk space occupied by utilities and application programs.

On the other hand, the message files needed to handle internationalisation could make disk space requirements expand — if they are provided. While SVR4 at last completes the "eight-bit clean-up" of the kernel and utilities, and provides a number of tools to help developers of multi-lingual applications, the conference presenters were a little vague about delivery dates for localised versions (localised, that is, for languages other than American English) of the new UNIX — although you can now buy release 3.2 supplements for French, German, Japanese and Korean, and get at least some documentation in these languages. (Contact your local AT&T sales office...)

Hopefully, things will be a little clearer in a year when the source has been delivered to licensees, allowing them to start work on their binary ports. Until then, I can follow the advice of the *How to get started now* sections which closed most of the presentations. As far as I can see, I only need a 3B2, a Sun, a couple of 386-based systems (one for UNIX, one for XENIX) and an Ethernet to connect them all if I want to cover all the bases.

Yes, that should keep me busy for a few months...

ANSI C Standard and Progress towards an ISO C Standard

Cornelia Boldyreff
corn@cs.brunel.ac.uk

UK POSIX Panel Convenor

Cornelia Boldyreff is a member of the British Standards Institution technical committee on Application Systems, Environments and Programming Languages. She acts as Convenor and Chairman of the BSI C Language Panel; and is one of the UK Principal Experts on the ISO Working Group on C. She is also Convenor and Chairman of the BSI POSIX Panel; and is one of the UK Principal Experts on the ISO Working Group on POSIX.

Recent Meetings

X3J11 Progress

At their September 1988 meeting, X3J11 processed comments from the third public review; and made a number of editorial changes to the draft standard, but no substantive changes were made to the draft. This means that X3J11 has nearly completed its work on the draft C standard which can now be submitted to X3 for final processing and approval as an ANSI C Standard.

It is anticipated that the final version with the editorial changes from the last meeting will be sent to X3 in December or January. If there are no changes from the X3 level review, the ANSI C Standard should be official in May or June as X3.159-1989.

Below is a fuller account of the recent X3J11 meeting by Doug Gwyn of the USA Ballistic Research Laboratory:

[Start of Quote]

At the September 1988 X3J11 meeting, a number of editorial changes to the third public review draft were approved, but no substantive changes were made.

Approval to submit the proposed ANS to X3 is contingent on review of the three official documents (Response to third public review, Standard, and Rationale), which will be in progress for the next couple of months.

Submission of the final proposed Standard and Rationale documents, along with any replies by recipients of our official responses to third-round comments, to X3 should occur by 05-Dec-1988 if all goes well.

Perhaps a note about "editorial" vs. "substantive" changes is in order, to forestall some possible complaints.

A change was deemed "editorial" in nature if it served merely to clarify wording that could have been reasonably interpreted as meaning other than what the Committee had intended, so long as the previous intention of the Committee was preserved.

On a couple of issues, the Committee had to settle whether a proposed change was editorial or substantive by voting on this question (if 1/3 of the voting members present thought a change was substantive, then it was taken to be so).

We did vote whether or not to adopt some substantive changes, but there was not sufficient support for them to attain the 2/3 majority required for a substantive change to the proposed Standard.

In fact, none of them even came close to that level of support; the Committee believed that the current specification is "good enough" for use as the official C Standard, and the editorial changes that were approved just serve to make the Committee's intentions clearer in a few places.

In some cases, notably with respect to what a signal handler may safely do in a portable program, the clarification of intent may surprise someone who had not understood what actually had been intended (many believed the former wording to have been unambiguous, but not what we meant to say). Therefore, although the third public review draft Standard is substantially the same as the one being submitted for official approval, you should wait for the official Standard to see the exact wording before making any irreversible decisions based on it.

Note that only Committee members participating in the final document review will really know the exact wording that gets sent to X3; some of the approved "editorial" changes may in fact not be made, depending on the judgement of the reviewers.

I had reservations about this, but since I'm reviewing two of the three documents (and will have input to the third), I should be able to satisfy myself that nothing gets broken at the last minute.

We'll see...

[End of Quote]

All people who submitted comments at the third public review will have an opportunity to review the X3J11 response to their comments; and if they feel that their comments have not been appropriately dealt with by X3J11, are entitled to lodge a compliant with X3 whose job it is to ensure that X3J11 has drafted the proposed standard with due process.

Future Meetings and Projected Targets

If the ANSI standard is acceptable to the ISO community, WG14 will put it forward for registration as a DIS; otherwise there will be another ISO ballot on the draft as a DP.

Because the recently approved X3J11 draft does not include substantive changes allowing European character sets to be accommodated in representing C source code, there is likely to be another DP ballot on the draft at ISO level.

Future Meetings

BSI IST/5/14 C Panel	8 Nov 1988	London, England
ANSI X3J11	12-16 Dec 1988	Seattle, Washington
ANSI X3J11	10-12 April 1989	Phoenix, Arizona
WG15	with X3J11 as above in April	

The POSIX Standard and Its Future Development

Cornelia Boldyreff
corn@cs.brunel.ac.uk

UK POSIX Panel Convenor

Recent Developements

The IEEE has given approval to the POSIX P1003.1 Standard. Copies of the recently approved P1003.1 POSIX Standard may be obtained from the IEEE at the address below:

IEEE Service Center
445 Hoes Lane, PO Box 1331
Piscataway, NJ 08855-1331
USA

The cost is $32.00 plus $4.00 postage and handling charge. The order number to quote is: SH12211. If you are an IEEE member the cost is discounted by 50%.

In the UK, the ISO DIS (the IEEE P1003.1 document with the line numbers and change bars remaining to facilitate comment) will be published as a BSI Draft for Public Comment; and will be available from the BSI through the usual channels.

Recent ISO WG15 Progress

WG15 recently held its second meeting in Tokyo from 20-21 October 1988. This meeting was attended by experts from Austria, China, Denmark, France, Japan, the Netherlands, the UK and the USA.

Following approval of the IEEE P1003.1 POSIX standard, the group worked to resolve any outstanding comments on this document at ISO level where it has been circulated as DP9945. These were resolved; and it was agreed to put forward this draft for registration as a DIS. This will mean that early next year there will be a six month ISO ballot seeking approval of the DIS prior to registration as an IS. The importance of having a DIS for POSIX is that this means it can be quoted as a requirement in government procurement contracts.

The other primary objective of the meeting was to clarify the proposed division of work which is the subject of an ISO ballot. This was accomplished; and a model relating the new work wth existing work was developed. Most of the new work will take the form of supplements to the existing base POSIX standard as shown below.

Part I will provide a functional definition of
- Process Primitives and Process Environment
- Files and Directories
- Input and Output Primitives and
  Device- and Class-Specific Functions
- System Databases & Data Interchange Format
- Supplements
  - Shell and Utilities
  - Realtime
  - Security
  - System Adminstration
  - Distribution Services

Part II will consist of language binding to C
Part III will consist of language binding to Ada

The need for three rapporteur groups was agreed by the meeting; these will be concerned with Security, Internationalisation and Validation.

Future Work

A plan of future work was refined at the recent ISO WG15 meeting. The group has developed a model relating existing and proposed work on the POSIX standard. Below is a simplified version of the model developed at the recent Tokyo meeting.

Recent and Future Meetings	
Nov 9	POSIX Panel
May 1	WG15
Oct	WG15
Apr	WG15
Oct	WG15

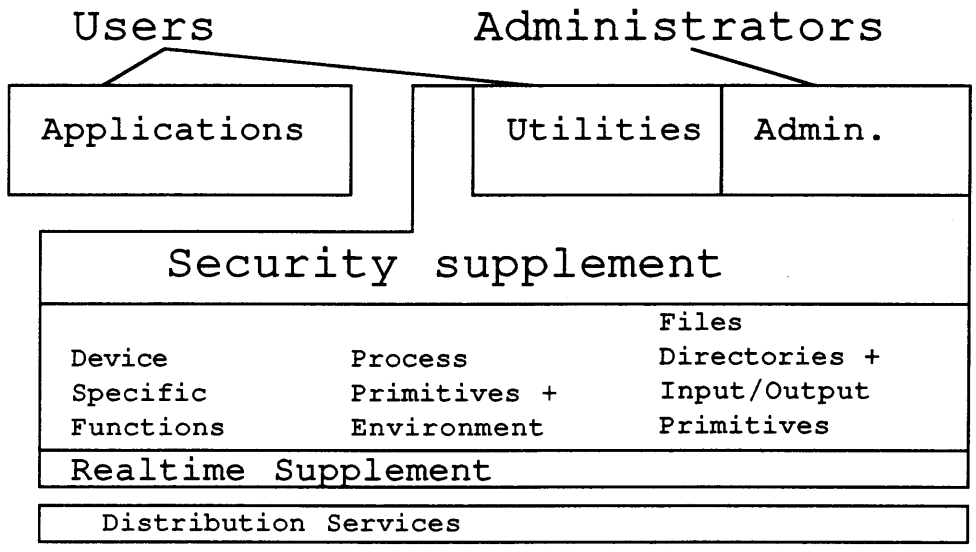


Diagram developed by ISO WG15 to explain their proposed future work. Tokyo, 20-21 October 1988.

AT&T System V/MLS

Terry Hart

AT&T Bell Laboratories
Government Systems Division

Multi-level Security with UNIX

The popularity of UNIX systems stems from their powerful features and the high degree of application portability they have achieved across a variety of computer hardware. The evolutionary design path of UNIX, however, has favoured openness, flexibility, and ease of data sharing, often at the expense of security. The problem now is: "How do we implement a trusted kernel, auditing, data labeling, and mandatory access control without compromising the compatibility, performance, and friendliness of UNIX System V?"

System V/MLS

System V/MLS has been developed by AT&T Bell Laboratories over the past two years to achieve a B-division level of security. It is a multi-level, secure version of UNIX that maintains full SVID compatibility while providing enhanced security features, a security audit trail, data labeling, and mandatory access control. The performance impact due to the addition of these features has been verified to be less than 4% for typical applications.

Enhanced Security Features

Earlier versions of UNIX have been subject to a variety of attacks in which users could gain superuser privileges. Measures were taken to prevent all such attacks from succeeding on System V/MLS. In addition, permissions were tightened throughout the kernel to prevent users and administrators from inadvertently giving away information or privilege. A random password generator was added to prevent users from selecting easily guessed passwords, and the encrypted passwords are hidden with other sensitive security information in files that cannot be read by users. Print jobs are processed with the first and last sheets containing the label of the information being printed. The superuser privilege is restricted in several ways. No "root" logins are permitted; a superuser must first login as a normal user, then su to root. The su command can also be restricted to select ports. While operating with superuser privilege, only trusted code can be executed, eliminating the possibility of a Trojan horse gaining superuser privilege. Secure multi-level e-mail has also been provided with the system and may be set up by the administrator for any label defined on the system. The 630 Multi-Tasking Graphics Terminal is supported by allowing its windows to operate with different labels; "cuts and pastes" between the windows are permitted within the constraints of the security policy.

Secure Audit Trail

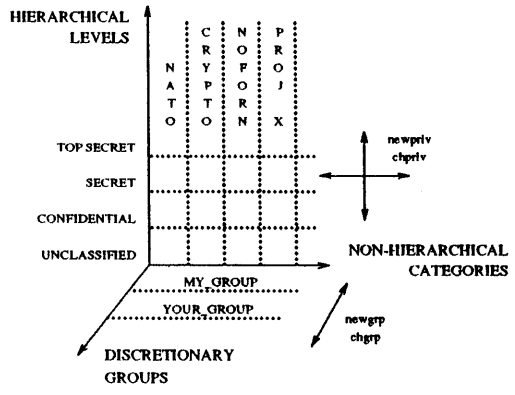
System V/MLS generates an audit record for all security relevant events and all data accesses. There are twenty selectable trace channels (see table), and the records may be stored on disk, tape, printer, or a remote system. A formatter is provided for use in analysing audit trail data in two levels of detail. It can, for example, search for all accesses by certain user, or for all users who accessed a certain file. Some enhanced tools are currently in development to provide reports and summaries to the administrator as well as real-time security alarms.

Data Labeling

Security labels are provided on all data structures. Up to 255 levels and 1024 categories can be combined to define as many as 60,000 different labels on the system. All objects are labeled when they are created with the privilege of the process that created them. A privilege is a combination of the label and the group associated with the process. One new command, *chpriv*, has been added to relabel files. It functions in a manner similar to the *chgrp* command, which changes the group associated with an object. Members of a special group, "secadm", may declassify objects that they are allowed to access.

Mandatory Access Control

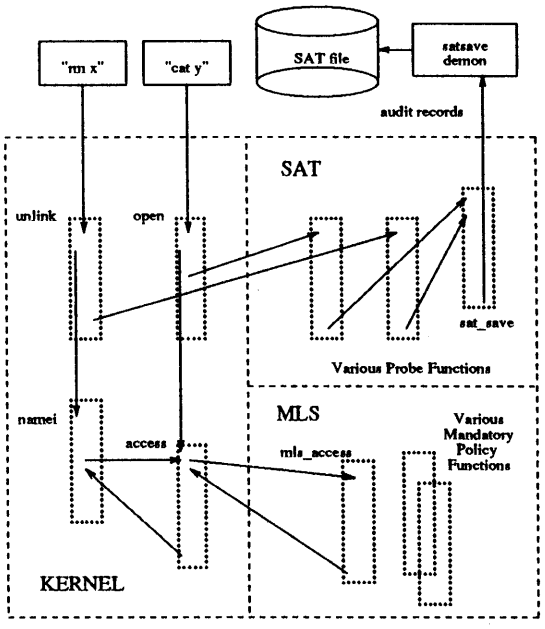
Users initially login with a privilege and are given a shell that operates with that privilege. A new command, *newpriv*, allows them to change to a new privilege in the same way *newgrp* allows them to change to a new discretionary group. Each *newpriv* to a higher level generates a new shell, which is stacked upon the login shell and runs with the new privilege. In this manner, the user may change privileges while maintaining the previous environment (see diagram).



A security policy of "no read up" and "no write down" is enforced. The user can look back and read the lower environment but cannot write to it; likewise, information labeled higher than the current privilege cannot be read. To further protect the operating system, "SYSTEM" is placed below the normal user level and cannot, therefore, be accessed by users, even if they were to obtain the superuser password.

System V/MLS Modules

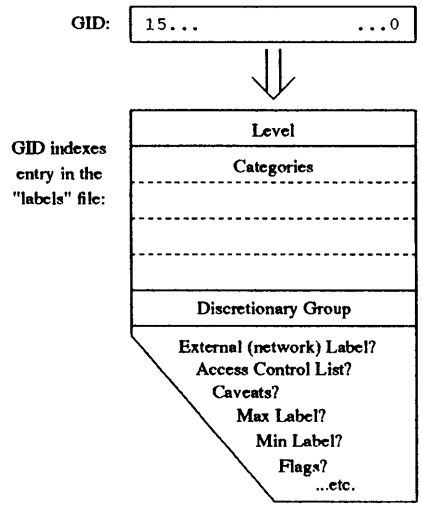
System V/MLS is implemented by two modules, which are called from about fifty "probe points" and "hooks" that have been placed in the UNIX kernel (see diagram).



This approach allows for ease of portability across various UNIX kernels and computer hardware.

Indirect GID-based Labels

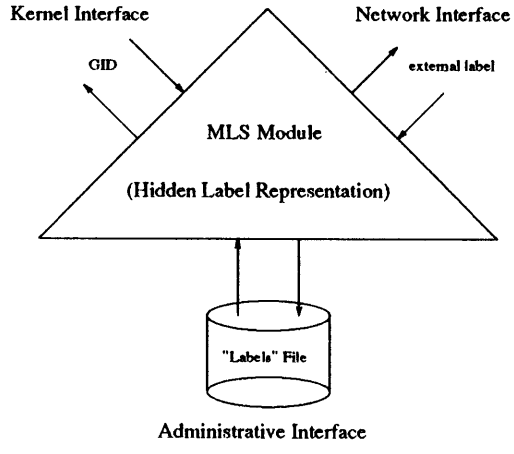
Data labeling is accomplished by using the group identification (GID) as a pointer to the labeling information, which is maintained in the /mls/labels file (see diagram).



This scheme has enabled System V/MLS to achieve a high degree of application compatibility since the GID has been a traditional mechanism in UNIX systems. It also has caused little performance degradation since most processes run with the same GID as the objects that they access; if the GID pointers are identical, there is no need to compute the label.

Hidden Label Representation

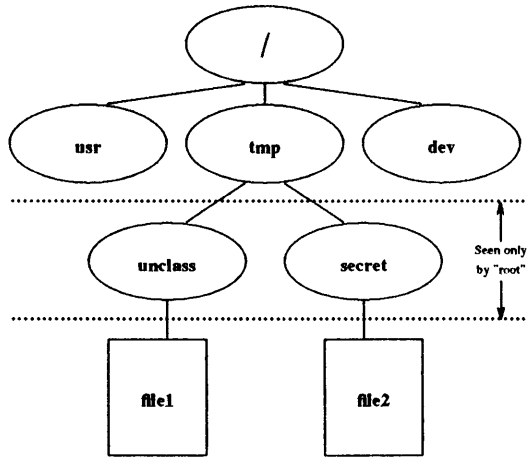
The MLS module "hides" the internal label representation and translates for each external interface (see diagram).



By doing so, conversions can be made as necessary to export labeling information to external networks or devices in their required formats.

SECURED Directories

Many UNIX directories must contain files with different labels (e.g. /tmp and /usr/mail). Since a directory can only contain files with identical labels, an approach was needed to maintain compatibility with these multi-level directories. To achieve this, System V/MLS inserts for each label a subjective directory that is invisible to the user (see diagram).



This subjective directory is labeled according to the process that created the files to be stored in it.

NCSC Evaluation

A developmental evaluation began in September 1987 and is concluding in September 1988 with the completion of the Interim Product Assessment Report. This begins the formal certification process for a B1-class rating. The evaluated system will be System V/MLS Release 1.1 based on UNIX System V Release 3.1.1 for the 3B2/500 and 3B2/600 computers. The 630 MTG intelligent terminal is included in the certified configuration. System V/MLS will be maintained by AT&T and its rating will be preserved using the NCSC's Ratings Maintenance Program (RAMP).

Porting Status

System V/MLS has also been ported to AT&T's 3B2/400 and 3B15/4000 computers as well as two other OEM's computers. Ports are planned for the 3B20, 6386 WGS, 3B2/700/800, and seven other OEM's computers in the near future.

System V/MLS Evolution

Release 1.2 is currently planned for December 15 and will include network interfaces for RFS, TCP/IP, and the STU-III. Release 1.3 is

scheduled for the second quarter of 1989 and will introduce compatibility with POSIX multi-groups, multi-processor support, integration with SVR4.0, access control lists, and fine-grain superuser privileges. Release 2.0 may be a candidate for a B2-class certification.

Secure Applications

Any application can be ported to System V/MLS, but some modification is generally required before it can be used as a multi-level secure application. Single-level applications are limited to operating at one privilege when run by a user and can easily be ported without modification. Multi-level applications, however, must operate with all privileges and therefore must be "trusted" to enforce the System V/MLS security policy. This typically requires some modification of the application to call System V/MLS library routines.

Application Ports

Several applications have been ported to run multi-level on System V/MLS. The data base management systems include CATALOG with plans for TRUDATA, UNIFY, INFORMIX, TUXEDO, and INGRES. Office automation systems include Q-Office with plans for Prelude. Networks interfaces include RFS and the STU-III w/UUCP with plans for TCP/IP, DATAKIT, and STARLAN.

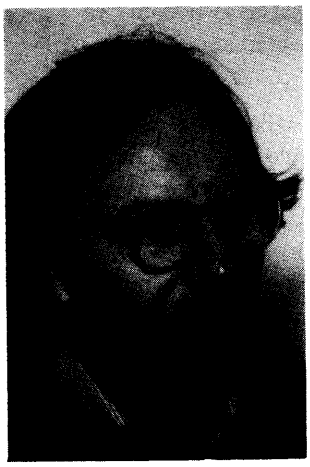
Secure Systems Strategy

If a desired level of security is to be achieved in the design of a computer system, the design must: use secure components with well-defined interfaces between the operating system, the application software, and the network protocols. If it is designed with an end-to-end view of the security requirements, it will be possible to avoid "weakest link" problems, eliminate the need for "patches," avoid system performance penalties, and enhance user acceptance. This should be the goal for designers of secure systems.

Unix Clinic

Colston Sanger
colston@olgb1.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. In his spare time he is an art historian and has recently organised the J.L.Agasse exhibition opening in Geneva in November and at the Tate Gallery, London in February 1989.

My terminal isn't working

OK, you know the scenario: one of your users complains that his or her terminal isn't working. Wearily, you walk around and take a look. You check the obvious (it's already plugged in and switched on, the contrast level is turned up and the terminal set-up seems all right, the line is connected and the `getty` is enabled). But still no response. The stupid cursor just sits there blinking at you from the top left of the screen. So what do you do? More often than not, you either kill the login `shell` or - worse - turn the terminal off, wait ten seconds and turn it on again.

What I mean to introduce by this scenario is a whole class of apparently simple, but really quite tricky problems of UNIX system administration.[1] For instance, staying with terminals for the moment, imagine that you have just bought a new terminal. Being the once-bitten, twice-shy system administrator that you are, you checked beforehand that you have a `termcap` or `terminfo` for it. You install the terminal and test it. `vi` works fine, but the **Whizzo-Office+** package you use for office automation doesn't. When you invoke the package, the terminal screen looks - how shall I say? - all *wonky*. What's the problem? The problem is the terminal has a magic-cookie glitch, *i.e.* the escape codes sent to the terminal to manipulate screen attributes (such as reverse-video) actually take up a character position on the screen. As a result, everything is off by one, the effect is cumulative and there's nothing you can do about it.

The only lesson to be learnt - and this applies primarily to application developers - is that you should design your screens defensively, leaving plenty of space around the reverse-video prompts. You should be aware that some terminals (such as the Wyse 50) have a magic-cookie glitch and some (such as the Wyse 60) don't.

Take another example. This time imagine you use **Whizzo-Master** for stock control. It's a COBOL package that prefers to have your Wyse 60 terminals set up to use *application* rather than *normal* keypad. It also uses a 25-line rather than 24-line screen, and it doesn't use `terminfo`. One day, sooner or later, something goes wrong. I don't know, maybe somebody forgets his or her

1. Dominic Dunlop discusses some of the common "gotcha's" in his paper on the interaction between third-party packages and standard UNIX utilities (*I come to bury UNIX ... and to praise it*, EUUG Spring Conference, Helsinki-Stockholm, 12-14 May 1987).

password. So, on the user's terminal you type:

```
$ su root
Password:
#
# vi /etc/passwd
```

(Sure it's dangerous, but you know how it is, the menu-driven `sysadm` just takes too long.)

Anyway, problem no 1: `vi` doesn't work properly. Instead of scrolling, the last line on the screen appears to overwrite itself. Why? Because your `terminfo` entry assumes 24 lines.

Problem no 2: when the user eventually goes back into **Whizzo-Master**, that doesn't work either - because `vi` has reset the terminal to use the *normal* keypad.

The printer isn't working

The other week we were trying to connect an Olivetti laser printer to a machine (not a 3B) that we had here temporarily. We went through the usual steps: as *root*, turn off the `getty`; then, as *lp*:

```
$ lpshut
$ lpadmin -v/dev/tty01 -mlaser -ppr1
$ lpadmin -dpr1
$ accept pr1
$ enable pr1
$ lpsched
$
$ cat /etc/group | lp
```

Nothing happened. We tried the standard kludge:

```
$ su root
Password:
#
# sleep 10000 > /dev/tty01 &
# stty 9600 -parity -tabs clocal \
> -echo ignbrk cread opost onlcr \
> ixon -ixany < /dev/tty01
#
# cat /etc/group > /dev/tty01
```

Still nothing. Obviously a cable problem. So we put a breakout box on the cable, and very quickly it became clear what the problem was. When the printer's buffer was full, it dropped Data Terminal Ready (DTR), which then dropped Request to Send (RTS), which in turn dropped Clear to Send (CTS), which dropped Data Carrier Detect (DCD) - which killed the print job. The solution was to wire Pin 8 (DCD) and Pin 20 (DTR) together at the printer end, but not actually to connect them to Pin 20. We tested it and it seemed to work. When the printer was on it was on, and when it was off it was disabled - which is what's supposed to happen. So that was that.

Later the same day, however, our AT&T Starlan network crashed. One of the 3B's went down with a **DOUBLE PANIC: system parity error interrupt**. It's been a long time since anything like that happened, and I'm not sure what caused it. The Error Message Manual says it could be dirt on the memory card connectors, but I doubt it somehow.

After we rebooted, one of the laser printers attached to that system wouldn't work either. `lpsched` was running and the printer was enabled, and all the other printers were working. So why wouldn't that one? In the end, it turned out that we use that particular printer under `lp` and under **Whizzo-Office+** (the third-party office automation package I mentioned earlier). The wonderful **Whizzo** creates a lockfile to secure exclusive access to the printer. When the computer crashed, the lockfile got left behind; when the computer was rebooted, the lockfile wasn't removed. Solution: remove the lockfile. Moral: it's probably not a good idea to use the same printer under `lp` and under a third-party package unless you can integrate the two somehow. In our case, I thought we had, but obviously there was still a loose end there.

Could you load this disk for me?

One bright, sunny morning, Joe User wanders into your office with a floppy disk in his hand. His shoes are freshly polished, hair neatly brushed, tie straight - you know you ought to be suspicious. 'I wonder if you could possibly load this disk for me?' he asks, ever so politely. That does it. You are definitely suspicious.

In fact, since you are running the AT&T DOS Server package (or Locus's **PC-Interface** for that matter) it's no problem. You load the disk on a PC and transfer the files on it to your 3B. Once they're there, you `dos2unix` them to get rid of the Carriage Return/Linefeeds and Control-Z's. End of story. Easy.

In fact, it's never that easy. The files are **Wordstar** files, and what Joe wants to do is edit them some more and then `troff` them. You play about with `deroff` to get rid of the 'dot commands', and with `col -b` and `tr` or something like this to get rid of the control

characters:

```
/*
* ctrlstrip.c -
*   silly little program
*   to strip control characters
*/
#include <stdio.h>
#include <ctype.h>

main()
{
 int     c;
 while ((c = getchar()) != EOF)
   if (!iscntrl(c) || c == '\n')
       putchar(c);
}
```

But it's not really satisfactory, and Joe is going to have to redo all the formatting.[2] To be fair though, most of the popular PC packages these days include file conversion utilities (**Multimate** comes to mind). And in any case, isn't this the problem X.400 is supposed to tackle?

Could you load this cartridge tape for me?

Finally, to the first, genuine reader's contribution to this column. Gert Illemann (gerti@ncrdk.dk) of NCR Denmark writes:

Reading streamer-tapes created on different brands of machines can be a real pain. (Maybe the same applies to other types of tapes, but I have the impression that it's not that bad.) Often it's outright impossible. QIC 11 and 24, 9 and 4 tracks - not to mention swapping of bytes and 'half-words' - are some of the phenomena that one has to wrestle with, even with machines from the same manufacturer. Frequently, the whole thing is done on top of a very unstable matrix, built out of pure guesswork, as to which utility was originally used to create the tape, which options the artist happened to seed it with and whether one's own or the originator's tape-drive needs cleaning or maybe even adjustment. Now and then the irksome job is crowned with success though.

There is a great need for a collection of heuristics. An E-mail friend and I have between us pieced together this beginning of a matrix:

+ = ok
- = no dice
? = don't know

Machine	NCR	UNISYS	PR1ME	3B2/600	SUN
NCR	+	+	?	+	-
UNISYS	+	+	?	+	-
PR1ME	?	?	+	+	?
3B2/600	+	+	+	+	?
SUN	-	-	?	?	+

When transporting from Tower XP, 32/400-/600 to Tower 32/200-/800 and vice-versa, bytes have to be swapped, *e.g.* use `dd conv=swab`. Also, some of the UNISYS machines are simply Towers adorned with another colour; the basic software is likewise the same. But UNISYS uses other manufacturers as well, so UNISYS ought to be asked.

Thanks, Gert. Now, how about all you other gurus out there? Surely you must have some helpful hints you'd like to share with the rest of us?

Good books

It's probably worth mentioning a few good books. No less than five books on UNIX system administration have been published in the last year or so. N.G.Backhurst and P.J.Davies, *Systems Management under UNIX* (London, Sigma Press, 1986) and Frank Burke, *UNIX System Administration* (London, Harcourt Brace Jovanovich, 1987) are disappointing. Eric Foxley's *UNIX for Super-Users* (Wokingham, UK, Addison-Wesley, 1985) is based on Version 7, and Martin D.Seyer and William J.Mill's *DOS / UNIX Systems: becoming a super user* (London, Prentice-Hall, 1986) is not bad, but the best is David Fiedler and Bruce H.Hunter, *UNIX System Administration* (Indianapolis, Hayden, 1986, **ISBN**: 0-8104-6289-3).

That's it. The next issue's column is provisionally titled **No space on integral hard disk drive 0, partition 0.**

2. If your users are continually wanting to convert files from one format to another, you might want to look at: Jeff Walden, *File Formats for Popular PC Software: A Programmer's Reference*, (Wiley, 1986, **ISBN**: 0-471-83671-0).

The European E-Mail Directory

Anke Goos
ag@laura.uucp

Computer Science Department
University of Dortmund
4600 Dortmund 50
P.O.Box 500 500
W-Germany
+49 231 755 2444

Do you not yet know what this is? It's the printed output of an EUUG-project for an compendium of electronic addresses of the most important European networks. For the first issue this means that the communication between EARN and EUnet will be improved by a common book of addresses. The whole book of about 200 pages includes a main part of all "mail"-able organisations, one index by keywords and another by site-name to help find organisations in the main part. And of course, there are texts which guide you on how to use the book and the most importantly how to address from one network into the other!

The main part of the book contains all electronic addresses of EARN and EUnet, site entries are sorted by country and towns. In case you forgot, for example, the electronic address of the editor of this famous journal, only knowing that Alain is sitting somewhere in Berkshire, you may look under UK, after that Berkshire and find an address like:

EUUG-Newsletter - Parliament Hill Computers
phcomp.uucp phcomp.co.uk
Alain Williams addw@phcomp.uucp
+44 734 461232

Each address shows the name and e-mail-address of the contact-person at this site. The postal code, street and any detailed information has been consciously left out to avoid this list being used for any sort of advertising mail.

If you only know one part of the organisational name, there are key-words in the form of a "permuted index", this perverse thing you all know and love in Unix-Manuals. Given one part of a name, you may use the whole organisation name or its country or town in order to find it in the main part. The keyword index also gives you a first glance whether there is any organisation or university with special interests in A.I., astronomy, biology or computers. (Not surprisingly there are hundreds of them in the last group :-).

And if you would like to know which organisation belongs to a strange site-name, a look into the "domain-index" should provide you with the corresponding organisation, town and country. I must confess that I always try to find the location and organisation of an unknown site, not yet being used to this network-anonymity that there's someone from host "somewhere" asking you something...

The whole project of the E-Mail Directory was started in August 1988 by Daniel Karrenberg and yours truly, all in cooperation with the national EUnet-backbones giving their corrected address-parts and ideas. A second draft had been presented at the EUUG conference in Portugal. For this first edition EARN has given its "yes" to participate in the common E-mail Directory. For the next issue - network-addresses grow fast - there is already some interest from other European networks like Hepnet... The book being printed in December of this year, the more or less cost-of-production price of 18 British pounds should promote the spread of the E-Mail Directory. Anyway, you have no choice as there is nothing else like this in Europe:-). Your feeling is right: We're proud to present...

Order-sheet for The European E-Mail Directory

To: European Unix systems User Group,
Owles Hall,
Buntingford,
Herts, SG9 9PL
United Kingdom
Tel: Royston +44 763 73039
Fax: Royston +44 763 73255
euug@inset.uucp

European E-Mail Directory

I would like to order a copy of the European E-Mail Directory to be published in January 1989.

Name _____

Address _____

_____

Post Code _____

Country _____

Telephone _____

Fax _____

E-mail _____

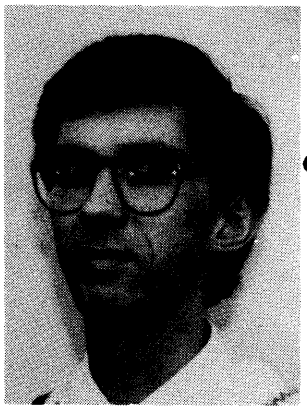
Price £18.00 including postage and packing, payable by VISA, Master Card, UK cheque or Eurocheque.

DO NOT SEND PAYMENT YET

(This is an order form only!)

This page may be photocopied for use.

EUUG Software Distribution



Frank Kuiper
euug-tapes@cwi.nl

Centre for Mathematics and Computer Science
Amsterdam

Now that the Portugal conference is over, and everyone is back at their desks, and when looking out of the window, seeing all those grey clouds and the leaves turn yellow, green, red or anything in between, we can yet again put all our attention towards the dark winter evenings. Evenings when we want to do more than just watch TV. We want to do something useful, and that is where this column comes in. Below is the list of currently available EUUG Software Distributions. Look at them carefully, there might be something in it for you.

The last year has been quite stormy in regards to the EUUG distributions. A couple of new distributions, and even more to follow, and all distributions are now available on Sun cartridge format.

In the near future there are the following interesting bits and pieces to look forward to. First, the Portugal conference has made people enthusiastic about some subjects mentioned during the conference. For one, the ET++ material from the University of Zurich made a remarkable impression. As soon as some legal issues have been solved, this material will be available in a new EUUG distribution. Another thing I am trying to arrange, is to have the next X windows release (X11R3) available as a separate EUUG distribution. The guys at MIT don't feel like makeing this available, other then via ftp. Hardly anyone in Europe has ftp access to the USA, so this way of distribution will not reach much of Europe. Hence the EUUG distribution.

In my article in the previous Newsletter, I mentioned that Usenix and EUUG would exchange their software distributions. Well, that's true all right, but things are just going a bit more slowly than expected, so you will have to wait a little while longer before the Usenix distributions will be available. I hope to have more news about this in the next EUUGN.

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 in the public domain, and you might even become famous!

This is a list of all the current (August 1988) 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. They do contain postage cost for surface mail within Europe. Any special shipment costs, like with DHL, will be billed to you.
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 larn.
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.

1. Commands and Application Programs

X	X-window 11 release 2
afio	manipulate archives and files
cake	maintain groups of related files like make
calctool	window based calculator.
changed	looks for files that have changed after given date
chat	Chat in real-time with other users (System V)
config	print details of machine and C compiler configuration
copytape	copy magtapes on systems with only one tape drive
cpmod	copy modes, ownerships and times of file
g++	GNU c++ compiler
gawk	GNU awk
gcc	GNU C compiler 1.18
gdb+	GNU debugger
gemacs	GNU emacs editor version 18.50
graphedit	a general purpose graphic editor
isode	ISO Development Environment
kermit	The Kermit file transfer program version 4E(067)
lem	interactive package for editing simple graphics
little-st	little Smalltalk system
magtapetools	magtape handling package
mcc	merge C compiler
modemcap	a modem independent dial(3) package
mstools	read and write MSDOS formatted diskettes from Unix
mush	The Mail User's Shell for electronic mail
news	the netnews system
nrofftab	nroff driver table compiler/de-compiler utility
pd-diff	public domain diff with the -b and -c options
pdtar	tape (or other media) file archiver
perl	Practical Extraction and Report Language (C,awk,sed,sh)
postscript	PostScript interpreter
psfig	a troff preprocessor for PostScript figures
pshalf	produce PostScript pages two to a (paper) page
ptc	Pascal to C translator
qsubst	qsubst - query/substitute strings in files
qterm	query a terminal to determine its name
revgrep	search a file for a pattern backwards, provide tails
ru	compact list of users on all
sc	spread sheet calculator
semex	interactive exerciser for System V semaphore operations
sets	performs set operations on its arguments
shar	create shell archive file for extraction by /bin/sh
shsem	Utilize System V semaphores from the shell
substr	extract a substring from the input arguments
vmail	tty interface to MH
w	display users and processes (for System V)

2. System Calls

3. C Library Subroutines

pc-curses	screen/window management library
regexp	regular expression handler
rpc	library routines for remote procedure calls
syslog	logging routines

4. Special Files

pty	pseudo tty driver for system V machines

5. File Formats

6. Games

moria4.85	a dungeon adventure game in the manner of rogue
omega	another adventure game

7. Miscellaneous

8. System Maintenance

arc	a general archive utility
autobaud	terminal speed detection
backup	perform tape backups
enable	enable, disable getty on tty lines
smail	UUCP mailer with routing
survey	generate simple plot of system load and # of users
sush	restricted shell to grant specific limited privileges
watcher	system monitoring program

Price : DFl 120,-/180,-

EUUGD14: This is version 4.0 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.

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 5924056 (or: +31 20 5929333)
Telex: 12571 mactr nl
Internet: euug-tapes@cwi.nl

Please note that for distributions D1, D2, D3 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. Tapes and bill will be sent separately.

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. 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: .....

EUUG

Glossary

There are approximately 5000 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.

acclamation	Loud or enthusiastic applause
acquire	Obtain, get
adorned	Furnished with ornaments
amenable	Liable to, capable of
attainment	To succeed at ...
besides	Next to – or Moreover, anyway
bestowed	Given to ...
blurb	Text - often in a manual
candidate	Something/one who hopes to pass a test/examination
Caveats	Warnings
chat	Talk (informal)
Chinese	A Chinese means: A meal in a Chinese restaurant
clarify	Remove doubts or questions
coincide	Occupy the same place/time/value
collaboration	Do (something) with
compendium	Collection, list
composed	Made up of
concept	Kernel/basis of an idea
consensus	Agreed opinion
contemporary	At the same time
dormant	Asleep, inactive
eligible	Have enough qualifications
emanations	Radiation, something output from ...
entwined	Bound/tied with ...
futile	Pointless exercise
glance	Quick look
gotcha's	Unexpected problem
inception	Beginning
indemnify	Insure/protect against ...
inevitable	... will happen without fail
innovation	New/bold ideas/actions
inordinately	Much too ...
Interim	Temporary (item/action/measure)
irksome	Tedious, tiresome
jeremiad	Lamentation, doleful complaint
latter	The second of two objects
persistent	Continues, lasting
preach	Say what others should do
redundant	No longer useful
refund	Give back (money)
severe	Strict, unkind
staring	Looking fixedly
weird	Very unusual

Here are the abstracts of the papers delivered at the EUUG Autumn 1988 Conference in Cascais, Portugal. Please contact the authors if you would like a copy of a paper.

Copies of the proceedings are available from Owles Hall at £20 each excluding post & packing. You may either send a purchase order to Owles Hall who will send an invoice with the copy of the proceedings, or you may telephone them and pay by Access (Mastercard) or Visa (Barclaycard).

Thanks are due to Dave Edmondson <dme@cc.ic.ac.uk> and Jan-Simon Pendry <jsp@cc.ic.ac.uk> or <jsp@doc.ic.ac.uk> who also typeset the proceedings.

Priority and Deadline Scheduling on Real-Time UNIX

Peter G. Bond
Ferranti Computer Systems Limited
Manchester
M22 5LA
UNITED KINGDOM

A Real-Time System can be defined as one which interacts with a real external activity and must respect deadlines imposed by that activity. Typically the system specification specifies a deadline for each system service, plus a required probability that the deadline should be met. There is therefore a requirement for a real-time scheduler which takes deadlines as well as priorities into account, and which raises an exception, or UNIX signal, as soon as any deadline is missed.

The Ferranti Real-time Extension to UNIX therefore includes a pre-emptive scheduler which implements "Earliest Deadline Scheduling" (EDS) for real-time processes, as well as "Static Priority Scheduling" and "Time Sharing" for other types of process. This paper describes the scheduler and also other features of the Extension, including fine timer resolution, fast reliable file handling, and deterministic and distributable inter-process co-operation.

NOFS – The NFS Server for the Optical File System

Paulo Amaral
INRIA – GIPSI
GIPSI-SM90†
Domaine le Voluceau Rocquencourt
78153 LE CHESNAY CEDEX
FRANCE
+33 1 3963 5511 ext.5179
paulo@gipsy.gipsi.fr

NOFS is a network server that implements the Network File System (NFS) protocol. It works above the Optical File System (OFS) to provide network transparent access over files stocked on optical disks. NOFS implements write once multi-version files keeping complete compatibility with the UNIX file system (UFS). It has been working since March 1988. The design and implementation of the Network Optical File Server (NOFS) is presented here. The development of the various NOFS modules and their integration with OFS is explained and cacheing is explored.

† GIPSI-SM90 is sponsored by the French Ministry of Research and Technology under the contracts 83-B1032 84-E0651 85-B0524

CHORUS, a New Technology for Building UNIX Systems

Frédéric Herrmann
François Armand
Marc Rozier
Michel Gien
V. Abrossimov
I. Boule
M. Guillemont
P. Léonard
S. Langlois
W. Neuhauser
Chorus systèmes
6 Avenue Gustave Eiffel
F-78183 Saint-Quentin en-Yvelines CEDEX
FRANCE

The CHORUS technology has been designed for building "new generations" of open, distributed, and scalable Operating Systems. CHORUS has the following main characteristics:

- a communication-based technology, relying on a minimum Nucleus integrating distributed processing and communication at the lowest level, and providing generic services used by a set of subsystem servers to provide extended standard operating system interfaces (a UNIX interface has been developed, others such as OS/2 and Object Oriented systems are envisaged).
- a modular architecture providing scalability, and allowing in particular dynamic configuration of the system and its applications over a wide range of hardware and network configurations,
- real time services provided by a real-time executive, and accessible by "system programmers" at the different

† CHORUS is a registered trademark of Chorus systèmes

system levels.

CHORUS–V3 is the current version of the CHORUS Distributed Operating System, developed by Chorus systèmes. Earlier versions had been studied and implemented within the Chorus research project at INRIA between 1979 and 1986.

This paper summarises the facilities provided by the CHORUS–V3 Nucleus, and describes the UNIX Subsystem built with the CHORUS technology that provides:

- binary compatibility with UNIX,
- extended UNIX services supporting distributed applications (light-weight processes, network IPC, distributed virtual memory) and real-time facilities.

Developing and Adapting UNIX Tools for Workstations

David Barnes
Mark Russell
Mark Wheadon
The Computing Laboratory
The University
Canterbury, Kent CT2 7NF
UNITED KINGDOM
djb@ukc.ac.uk

This paper describes our experiences in developing tools for high performance graphical workstations. In particular we concentrate on the ways in which some tools and concepts familiar to users of glass-teletype UNIX systems have been adapted and exploited within a new environment. The tools described are a file differencer, an execution profiler and a file browser.

Secure the Superuser

Mohamed Benakli
Université P. et M. Curie
Laboratoire M.A.S.I
Tour 65-66
4 place Jussieu
75252 Paris CEDEX 05
FRANCE
mcvax!inria!litp!bm

Our previous research has tried to provide the UNIX operating system (System V) with security features equivalent to the C2 level as defined by the DoD (Department of Defense). Our system doesn't obstruct the user's action, unlike other systems where the user is often limited to a restricted environment. The security issue is brought into question by the ever possible eventuality that the superuser's password might be disclosed. In that case, it seems obvious that all the security efforts will be reduced to nothing. Among the various improvements already made by us in the field of the UNIX security, we have realised the development and the implementation of the algorithm as proposed by Amos Fiat and Adi Shamir during the SECURICOM '87 Congress ("Unforgeable Proofs of Identity") . This implementation would allow a supplementary access control for the superuser, thus increasing the potentiality of making UNIX secure.

Clouds – A Distributed, Object-Based Operating System Architecture and Kernel Implementation*

José M. Bernabéu-Aubán
Phillip W. Hutto
M. Yousef
A. Khalidi
Mustaque Ahamad
William F. Appelbe
Partha Dasgupta
Richard J. LeBlanc
Umakishore Ramachandran
School of Information and Computer Science
Georgia Institute of Technology
Atlanta, GA 30332-0280
USA
clouds@gatech.edu

Commercial UNIX is a direct descendent of a research operating system that used simple, innovative design concepts (e.g., pipes and a hierarchical file system with devices as files) that effectively exploited new hardware trends (e.g., minicomputers and low-bandwidth networks). *Clouds* is a research operating system that draws on the twenty years of software innovation and hardware evolution which have occurred since UNIX first appeared.

Clouds is a native operating system intended for large, heterogeneous hardware environments consisting of inter-networked workstations, compute-servers and data-servers (file-servers). We intend for *Clouds* and UNIX to coexist cooperatively, each system benefiting from the other's advantages. *Clouds* is not implemented "on top of" UNIX nor is it intended to replace or emulate UNIX. A new kernel for the *Clouds* operating system called the *Ra* kernel has recently been completed. *Ra* provides three primitives, *segments*, *virtual spaces*, and lightweight processes called *isibas*, which can be composed in various ways to construct components of the *Clouds* operating system.

This paper describes the architecture and organisation of the *Ra* kernel and details of its implementation. We sketch the implementation of *Clouds* services (objects, threads, distributed shared memory, etc.) using *Ra* primitives to demonstrate the versatility and power of the *Ra* kernel. These constructions use *system objects* and *kernel classes*, two novel features of *Ra*. Finally, we discuss our experience of using an object-oriented language (C++) to build a distributed, object-based operating system kernel that is both portable and minimal.

The Arabisation of UNIX

Pascal Beyls
BULL
1 rue de Provence

* This work has been funded by NSF grant CCR-8619886.

38432 Echirolles
FRANCE
beyls@echbull.uucp

The concepts and mechanisms provided by the internationalisation of UNIX take into account mainly the characteristics of the European languages. These concepts can be used for other languages using an 8-bit codeset, such as Arabic. However, some particularities of the Arabic language must be addressed. This paper relates the implementation of an Arabic UNIX called AB.CIX.

Along with a brief presentation of the principal characteristics of the Arabic language and the state of the art in terms of standardisation, we described the codeset used, the difficulties bound to the direction of the language (right to left), the alphabet, and the vowelisation, and their impacts on the subroutines and utilities such as *ctype* or *vi*. In addition, some new subroutines and utilities are needed to provide a complete solution.

Priority and Deadline Scheduling on Real-Time UNIX

Peter G. Bond
Ferranti Computer Systems Limited
Manchester
M22 5LA
UNITED KINGDOM

A Real-Time System can be defined as one which interacts with a real external activity and must respect deadlines imposed by that activity. Typically the system specification specifies a deadline for each system service, plus a required probability that the deadline should be met. There is therefore a requirement for a real-time scheduler which takes deadlines as well as priorities into account, and which raises an exception, or UNIX signal, as soon as any deadline is missed.

The Ferranti Real-time Extension to UNIX therefore includes a pre-emptive scheduler which implements "Earliest Deadline Scheduling" (EDS) for real-time processes, as well as "Static Priority Scheduling" and "Time Sharing" for other types of process. This paper describes the scheduler and also other features of the Extension, including fine timer resolution, fast reliable file handling, and deterministic and distributable inter-process co-operation.

Direct Manipulation Tools for UNIX Workstations

J. D. Bovey
M. T. Russell
O. Folkestad†
The University of Kent at Canterbury
jdb@ukc.ac.uk

† Now at *Ingenioerene Bond & Co, Treschowsgate 2B, 0477 Oslo 4, Norway*

Direct manipulation is one approach to the creation of software which can make use of the high resolution graphics and pointing device available on a workstation like a Sun 3. A direct manipulation tool is typically used to manipulate a complex system like, for example, a file system, and works by presenting a graphical image of the system which the user can manipulate in order to manipulate the system itself.

The paper starts off by discussing direct manipulation in general terms and then goes on to describe three examples of direct manipulation tools which were written at the University of Kent. The tools described are a file system editor, a graphical debugger and a front end to SCCS.

The remaining sections of the paper discuss the implementation of direct manipulation tools, outline some of the user interface techniques that are applicable, and suggest a few systems which may be amenable to the direct manipulation approach.

Installation Documentation Documentation

Linda Branagan
Information Technology Center
Carnegie Mellon University
lb1y+@andrew.cmu.edu

David Tilbrook
Information Technology Center
Carnegie Mellon University
Pittsburgh, PA
USA
dt+@andrew.cmu.edu

There is no single definition of UNIX. In spite of the variations, vendors are expected to supply their products to a wide variety of UNIX environments. In addition to creating code portability problems, the wide range of target systems complicates the installation process. Unlike proprietary single machine type operating systems, for which installations can be fully automated, UNIX installations are characterised by a great need for human participation. Typically, installers must edit configuration files and makefiles, create target directories and diagnose the problems that inevitably arise. Worst of all, there are almost as many installation procedures as there are UNIX products.

A client's first impression of a software product is the ease with which it installs on his/her system. Because the process requires a large amount of human intervention, UNIX software should come with documentation that makes clear exactly what is expected of the installer. This documentation needs to be accurate, complete and, above all, concise. Lengthy, hard-to-follow installation documentation will not be consulted until something goes wrong – and will prove inadequate even then.

It is not hard to produce effective installation documentation, but there are many pitfalls for the unwary. (This paper contains several horror stories which demonstrate these pitfalls). Avoiding them is especially difficult because installation documentation is typically created by programmers (not writers) who are usually working under tremendous time constraints.

We have applied techniques from the field of document design to produce a set of *installation documentation guidelines*. Our guidelines, which are presented here, include a generic outline of a good installation document and a discussion of strategies for the unintrusive (i.e., painless) integration of the development of installation documentation into the development of the software itself.

Precise Standards through Formal Specifications: A Case Study: the UNIX File System †

O. Declerfayt
B. Demeuse
F. Wautier
P.-Y. Schobbens
E. Milgrom
Université Catholique de Louvain
UNITE D'INFORMATIQUE
Place Sainte-Barbe 2
B-1348 Louvain-la-Neuve
BELGIUM
lpg@lln-cs.UUCP

System standards, which are means for defining how a system should behave, are used by a wide range of people for various purposes. Standards must therefore be precise, complete and unambiguous. Informal specifications written in natural language cannot lead to definitions having these properties, but formal ones can. This principle has been applied to the UNIX system: we present here a formal specification of the UNIX file system at the command level and at the system calls level. The *System V Interface Definition* has played the role of the informal starting point for this specification. The development of a formal specification has emphasised the deficiencies in the definition of UNIX and the differences between the various versions of UNIX, even for this supposedly well-known part of the system. The specification has been used to prototype (to simulate) the behavior and the effects of the various commands and system calls. This case study can be considered as a first step towards a formal specification of the complete UNIX system, providing a complete and unambiguous definition, which we feel is the better way to define a standard.

Guide: An implementation of the COMANDOS object-oriented distributed system architecture on UNIX

D. Decouchant†
A. Duda†
A. Freyssinet‡
E. Paire‡
M. Riveill†
X. Rousset de Pina†
G. Vandôme‡
IMAG-Campus
BP 53X
38041 Grenoble CEDEX
FRANCE
gerard@imag.imag.fr

The present paper describes the implementation of an object-oriented distributed operating system on a network of workstations operating under UNIX. The system is called Guide (Grenoble Universities Integrated Distributed Environment) and embodies the object-oriented architecture defined in the COMANDOS Esprit Project (Construction and Management of Distributed Office Systems). First, a brief presentation of the general principles of the COMANDOS architecture is given. Then, the Guide implementation is described showing how UNIX facilities are used to implement the COMANDOS architecture. Finally, the adequacy of UNIX to support object-oriented systems is discussed. The current state of the implementation and some conclusions are given.

Can Big be made Beautiful?

Richard Duckworth
Information Technology plc
The ANSA Project
24 Hills Road
Cambridge CB2 1JP
UNITED KINGDOM
ansa@gm.rl.ac.uk
+44 223 323010

This paper briefly explores the architectural requirements upon operating systems that have to support distributed processing, with particular reference to transaction processing systems, which are a major concern to many companies, including our own, Information Technology plc. The limitations of UNIX as the vehicle to support distributed processing are then explored and a new architecture which can deliver the required functionality is outlined. In conclusion, the paper explores how a transition can be made from current UNIX to a better regime without undermining all of the investment that has gone into UNIX as a system development environment and application platform.

An Implementation of Optical Disk WORM File System under System V Rel 3.0

Caterina Falchi
I.A.N. s.r.l. via Canova, 7
20090 Trezzano s/n MI
ITALY

A *WORM File System* (WFS) has been implemented on an optical disk WORM in order to obtain access with standard

† Project supported by the "Services de Programmation de la Politique Scientifique", contract ARC 84/89-73.

† Laboratoire de Génie Informatique

‡ Centre de Recherche Bull

read/write commands and procedures as for magnetic disks. The *WORM File Management* (WFM) has been directly integrated into the kernel of UNIX System V Rel 3.0 via the File System Switch, to ensure that each access to the WFS, via the commands previously developed for a magnetic disk file system, are fully transparent to the WFS itself. The WFM has been tailored to the write once and read many characteristic and optimised in order to obtain:

— Media transportability. All data and data structures are written and available on the WORM media.

— Optimised access time and space usage. A virtual magnetic disk partition is used as temporary support for all data (such as superblock, inodes, directories, etc.) subject to frequent changes. A write on the WORM takes place only during "umount", i.e. no more changes are due, to avoid waste of WORM surface.

— Data integrity. Files are sequentially written along with a header; some information related to files is redundant; special tools for the WFS check have been implemented.

— Disk block size.
The disk block can be dimensioned to optimise I/O transfers and/or WORM usage.

Modelling UNIX with an Object Oriented Model

Conception Fernandez
L.I.S.T. boite 1000
Université P. et M. Curie
4 Place Jussieu
75230 Paris CEDEX 05
FRANCE
mcvax!inria!litp!jl

HBDS (Hypergraph-Based Data Structures) is a modelling tool based on Abstract Data Types (ADT) and is used to represent knowledge structures. In this paper, we use this formalism to show the structure of UNIX. The UNIX kernel behaviour is also described by algorithms working on the abstract data types resulting from HBDS models.

This general approach allows us better understanding of UNIX concepts and mechanisms, and can thus be applied to point out UNIX deficiencies, to describe some algorithms and even to teach UNIX.

ET++ – An Object-Oriented Application Framework in C++

Erich Gamma
André Weinand
Rudolf Marty
Institut für Informatik
University of Zurich
Winterthurerstr. 190
CH-8057 Zurich
SWITZERLAND
gamma@ifi.unizh.ch
gamma@unizh.uucp

ET++ is an object-oriented application framework implemented in C++ for a UNIX environment and a 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.

The paper describes the graphic model and its underlying abstract window system interface, shows composite objects as a substrate for declarative layout specification of complex dialogs, and presents a model for editable text allowing the integration of arbitrary interaction objects.

Implementation of a Locking Protocol for Resource Locking in a Stateless Environment

Peter Gloor
Rudolf Marty
Martin Zellweger
Institut für Informatik
University of Zurich
Winterthurerstr. 190
CH-8057 Zurich
SWITZERLAND
gloor@ifi.unizh.ch
gloor@unizh.uucp

There is a noticeable trend towards stateless distributed filesystems, the best known example of such a filesystem being Sun's NFS.

File and record locking is one of the dominant problems for a stateless file server. By definition, a stateless server does not maintain any information about its clients. Therefore, it is not allowed to lock any resources for them by storing lock information on behalf of its clients. This is the reason why filesystems with locking capabilities are frequently implemented following the stateful approach.

We introduce a new locking method for resource locking in a stateless environment. Our method combines the advantages of the stateless server (easy crash recovery) with the advantages of the stateful server (easy locking) without relinquishing the statelessness of the server.
The algorithm we propose (called Dynamically Synchronised Locking, DSL) can be used, for example, to implement locking facilities in a network of workstations loosely coupled by a high speed LAN. In the following paper we describe the implementation of a locking facility for a stateless distributed filesystem, namely Sun's NFS.

The Translucent File Service

David Hendricks
Sun Microsystems, Inc.
2550 Garcia Ave.
Mountain View, CA 94043
USA
hendrick@sun.com

The Translucent File Service (TFS) is a Sun Operating System (SunOS) filesystem with copy-on-write semantics. The TFS allows users both to share a file hierarchy and to have a private

hierarchy into which files from the shared hierarchy are copied as they are modified. Consequently, users are isolated from each other's changes, as files in the shared hierarchy are guaranteed not to change. Files are only copied when they are modified, conserving disk space. The TFS was built to support Sun's version configuration and management tool, the Network Software Environment (NSE).

The TFS requires no modifications to existing programs to use it. The TFS also preserves the file name space, so that a user doesn't have to connect to a funny directory to use the TFS. For example, it is possible to view the directories /usr/src or /bin through the TFS.

The TFS is currently implemented as a user-level server process; it is not part of the SunOS kernel. Even though the TFS is not a kernel-based filesystem, it shows reasonable performance. This paper describes the current implementation of the TFS and the pros and cons of this implementation. It concludes with some ideas for future enhancements, including areas where performance can be improved.

EUnet and OSI Transition Plans

Daniel Karrenberg
Centrum voor Wiskunde en Informatica
P.O. Box 4079
NL-1009 AP
Amsterdam
NETHERLANDS
dfk@cwi.nl

EUnet, a pan-European cooperative R&D network, is described in terms of applications, protocols and topology. A strategy for the introduction of OSI applications and protocols into this network is then presented. The actual talk will provide additional up to date information about other developments in EUnet.

Implementing a POSIX Compatible Operating System on a Multi-Transputer Supercomputer

Paul J. King
Real Time Systems Ltd.
PO Box 70
Douglas
Isle of Man

This paper describes an implementation of IDRIS, a POSIX conforming UNIX-like operating system, for the Parsys SN1000, a multi-transputer supercomputer. It also highlights the differences between this implementation of IDRIS and others running on more conventional architectures.

The topics presented outline software strategies that allow IDRIS to run without a hardware interrupt system, to distribute processes across transputers and to handle alien processes (i.e. occam programs).

An Object Base for Attributed Software Objects

Andreas Lampen
Technische Universität Berlin
Sekretariat FR 5-6
Franklinstr. 28/29
D-1000 Berlin 10
WEST GERMANY
andy@coma.uucp

Axel Mahler
Technische Universität Berlin
Sekretariat FR 5-6
Franklinstr. 28/29
D-1000 Berlin 10
WEST GERMANY
WEST GERMANY
axel@coma.uucp

The UNIX filesystem supports a fixed set of attributes for filesystem objects, stored in inodes and directory entries. The (path-)name attribute is the sole means to identify and access a filesystem object. This turns out to be a rather severe limitation for certain complex applications such as large scale software development, where software objects typically evolve in a considerable number of versions.

An approach to improve the situation is introduced by the *attribute filesystem* (AFS), the system described in this paper. The AFS combines the notion of immutable objects (versions) with the possibility to attach any number of user-definable attributes to any attributed software object (ASO). AFS objects can be identified by specifying any set of attribute value(s) as retrieve pattern. The *name* of an AFS object is treated as *just another attribute*. The AFS is equipped with a proper retrieve interface that allows non-unique identification of *sets of objects* and provides operations on those sets.

The AFS is a significant extension to the UNIX filesystem interface providing applications with a unified, consistent view of attributed filesystem objects comprising immutable versions and ordinary UNIX files. The concept of *persistent objects* makes AFS a basis for object oriented applications in an UNIX environment. We used AFS as a basis for the implementation of the toolkit, a collection of programs supporting software configuration management in multi-programmer software development projects.

One important objective of AFS is to abstract from the actually underlying data storage system. This paper will briefly discuss two different implementations of AFS – one on top of the UNIX filesystem and the other based on , a dedicated software engineering database.

Establishing a Harmonised Testing Service for POSIX

Jon Leigh
The National Computing Centre Ltd
Oxford Road
Manchester M1 7ED
UNITED KINGDOM
jon@leo.ncc.co.uk

This paper will outline the principles being used, and activities being undertaken within the European Commission project to provide Harmonised Testing Services for POSIX. The general

objectives, principles and benefits of conformance testing are explained with reference to the activities of the CTS-2 POSIX Project. The wider implications of the test service with relation to future POSIX related standards are also covered.

Distributed Light Weight Processes in MOS †

Amnon Barak
Dalia Malki
Department of Computer Science
The Hebrew University of Jerusalem
Jerusalem 91904
ISRAEL
dalia@humus.huji.ac.Il

Integrated multicomputer systems consist a set of loosely coupled processors, each with its own local memory, into a single machine environment. In the distributed systems model, various user processes may run concurrently on different machines and possibly communicate to achieve a common goal. This form of concurrency encourages a programming style that uses large grain-size computation blocks. Such *distributed programs* consist of a set of execution entities (called *threads* or *tasks*) that perform considerable amount of work independently and communicate infrequently through messages. Threads are a convenient way of expressing concurrent programs and therefore, many programming languages embody thread-like entities in their syntax, *e.g.* Occam [IN84a] and Linda [ACG86]. However, the overhead of handling processes by the operating system is costly. For instance, it has been noted that the UNIX processes are *heavy-weight* in that they carry much associated state information. Therefore, operations on them (*e.g.* context switching) are slow.

Light Weight Processes (LWP) has been suggested by Kepes [Kep85] as a programming tool for supporting cooperating processes on a uniprocessor. In the LWP mechanism suggested by Kepes, a runtime support library provides the coroutine primitives within a single, heavy-weight-process (HWP). Another alternative for supporting LWPs is at the kernel level. On a multiprocessor, the kernel support version has a primary advantage of allowing real parallelism. One of the most recent operating system kernels that support LWPs is Mach [ABG86]. However, none of the kernel or user level LWP mechanisms provide concurrency in distributed environments.

This paper describes the Distributed Light Weight Processes (DLWP) mechanism, a facility for supporting distributed programs in MOS, a multicomputer operating system [BaL85]. The goal of the Distributed Light Weight Processes mechanism is to be able to exploit concurrency in a distributed environment. The mechanism is designed to be able to support a variety of application types by supporting processes as a programming tool. It exploits concurrency up to the level available in the system and provides additional, virtual concurrency through time sharing. In this way, it can be used both for efficient utilization of concurrency and for experimenting with large scale concurrent programs.

The DLWP mechanism is implemented immediately above the operating system kernel level, in the form of a user-level runtime library. It extends the uniprocessor Light Weight Processes mechanism through a new operation, *split*, which adapts the classical Light Weight Processes mechanism for distribution and dynamically disperses the workload among processors. A LWP *pod* within a HWP may *split* to create multiple pods that execute in different HWPs. The MOS dynamic load balancing [BaS85] automatically assigns the HWPs to different machines and provides concurrency.

The partitioning strategy takes into consideration past behavior of the LWPs, in terms of CPU consumption and communication. This profile information is used to reach a partition that splits the load evenly while incurring minimum communication overhead. For this purpose, the profile information is kept in a graph and a heuristic graph partitioning algorithm is employed.

† This work was supported in part by the U.S Air Force Office of Scientific Research under grant AFOSR-85-0284, in part by the National Council for Research and Development, grant no. 2525, and in part by the Foundation for Research in Electronics Computers and Communication, Israel Academy of Science and Humanities.

IMAGES – an approach to an Object Oriented UIMS

Luis Simoes
J. Alves Marques
Nuno Guimarães
Luis Carriço
Manuel Sequeira
INESC
Lisboa
PORTUGAL

This paper describes the User Interface Management System (UIMS) of the Somi Workstation. IMAGES is based on a model which takes a comprehensive approach on the most important aspects of the user interaction: functional model, visualisation model and interaction control.

The model addresses all the relevant issues in the Interface design enforcing a clear separation between the interaction and the execution. Based on an object oriented approach a specification model for the application designer was defined. A description of how the model is used is presented.

Finally the run-time support is described together with the implementation environment based on X Window System and C++.

Securing UNIX

Philip Martin
Gould Computer Systems
London
UNITED KINGDOM
+44 1 643 8020

There is growing recognition that information held on computer systems needs to be protected against unauthorised access. The requirement is particularly acute where Government systems are involved and national security could be compromised. There are many ways of ensuring the security of a computer system and any one method should not, of course, be utilized in isolation, but a selection of methods employed according to the threat. These methods include:

— Physical and personnel security.

— Correct administration of the computer system.

3) Preventing computer systems from leaking data into the environment through uncontrolled electrical and electromagnetic emanations.

— Computer software that provides controlled access to authorised users.

Proper organisation of the system, ensuring that information is not emitted by the computer radiating unintentional signals, physical security and ensuring that the computer software does as much as possible to limit access of the information to those authorised to receive it. In the case of the latter, it is important to provide audit trail facilities, to ensure proper monitoring and management.

This paper deals primarily with making the computer operating system software secure and gives an overview of how Gould Computer Systems has produced a version of UNIX that achieves this. The paper goes on to describe (as a real life example), Gould's current secure UNIX (UTX/32S) product and the intended route for further development.

Sacrifices to Ra
or
Learning to Administer a Sun Network

Bubbette McLeod
Silicon Compiler Systems
16 Independence Blvd.
Warren, NJ 07060
USA
bub@yquem.arpa

Having administered to a couple of 11/70's, a VAX, and a few assorted UNIX boxes for several years, I didn't really expect significant surprises or different problems in coming to administer a network of Sun workstations for a CAD tools company. As it turned out, my previous experience hadn't prepared me for suddenly being up to my ears in boards, transceivers, and cables, and the challenge of file servers and NFS.

As well as confronting the hardware itself, having to fit it together like a bunch of overcomplicated Meccanos, I had to learn to recognise problems with incompatibility. Not only was it necessary to match the correct board with the appropriate Sun computer, but what sometimes appeared, at first glance, to be a software problem could be any of a number of hardware problems. One ethernet board might not be dealing successfully with the ethernet board on another file server, or a board might need to be jumpered for compatibility with a certain type of transceiver.

With any complex and sophisticated system, there's a lot to learn in order to be able to keep it running correctly. There were the problems dealing with the network file systems themselves. Though file servers seemed a perfectly reasonable concept, learning to make them work correctly was another story. Another challenge was learning to diagnose problems with the Sun yellow pages, which transports assorted databases between the various computers. Sometimes it worked, sometimes it didn't. When it didn't, which machine was at fault? In fact, when anything went wrong with the network, which machine was at fault? Distributed file systems cause distributed problems.

As time went on, all these and other problems became a great deal easier to diagnose and solve. This paper will discuss the problems of dealing with a large local area network in an attempt to make others who find themselves in similar situations feel better.

Real-time multiprocessing under UNIX

Peter L. Petersen
Institute of Electronic Systems
Aalborg University
DENMARK
pp@iesd.dk

This paper describes how real-time facilities can be made available in a UNIX environment using multiple processors with different operating systems. The system implements general semaphores and message semaphores which permit synchronisation and communication between any processes on any CPUs within a VME-bus environment.

NeWS and X, Beauty and the Beast?

W. T. Roberts
A. Davison
K. Drake
C. E. Hyde
M. Slater
P. Papageorgiou
Department of Computer Science
Queen Mary College
190 Mile End Road
London E1 4NS
UNITED KINGDOM
liam@cs.qmc.ac.uk

NeWS and X11 are the two best known distributed window systems. This paper presents experience of both NeWS and X11 at Queen Mary College, highlighting strong and weak points of both systems and looking at future developments, including the much-heralded X/NeWS combined server.

Eolas — The Implementation of an Office Information Server

Mark Sheppard
Brian Caulfield
Sean Baker

Ann Barry
Janet Dillon
Marysia Cahill
Distributed Systems Group
Department of Computer Science
University of Dublin
Trinity College
Dublin
IRELAND
doeois@cs.tcd.ie

Office Automation has emerged as a significant area of research and of commercial importance. The concept of the automated office varies greatly; from office environments that use small computer systems, running applications such as spreadsheets, wordprocessors and database utilities in their day to day operations, to those that use an integrated system comprised of a number of computers, or nodes, connected together by a network, supporting various degrees of distribution transparency and which provide convenient means of sharing resources, utilities and information.

Within this framework there exists the concept of an information storage and retrieval subsystem. This we term an Office Information Server (OIS), which may be described as a persistent repository for office information objects of varying complexity, ranging from standard types, for example integer, real and string, to highly structured object types such as documents which may be composed of voice, graphics and text. This server provides a set of services, described by its functional interface specification, whereby client application programs, and other office system sub-components can access and manipulate the stored information.

This paper initially describes various aspects of such an OIS; the data model developed (the Fact Model), the mechanism for automating office tasks (office procedure support), the synchronisation and recovery of the system (transaction support) and the client/server interaction model, which provides distribution transparency.

Then, the implementation of a prototype OIS† is described, together with the OIS abstract architecture, and its realisation under UNIX. The emphasis of the paper is on the realisation of the OIS under UNIX, and establishing its suitability for applications such as the OIS. This work was performed in a 4.2BSD UNIX environment, using readily available tools and applications. For example, the Fact Model Language compiler was developed using Lex and Yacc, the persistent store for the office data is supported by a relational dbms (INGRES), and the client/server interaction model was built using an available RPC package (4.2BSD implementation of Courier).

Hardware and Software Aspects of Tightly Coupled Symmetrical UNIX Multiprocessors.

Brian E. J. Clark
Pyramid Technology SA
SWITZERLAND
bejc@pyrltd.uucp

Ali Shirnia
Pyramid Technology Ltd
UNITED KINGDOM

In the quest for improved performance from UNIX systems, system designers have been drawn to solutions based around tightly coupled symmetrical multiprocessors. On the surface this solution is a very attractive way of realising performance levels that are beyond that of a single processor. This paper reviews the hardware and software design techniques that must be adopted in order to realise these designs.

For the hardware designer the increased processor throughput places a far higher load on the memory, I/O and bus structures. The paper will examine how the improved cacheing and other techniques, can be used to meet the increased demand within an existing system structure. It will also indicate other potential problem areas in the hardware design.

For the software engineer implementing the UNIX kernel, the major problem is one ensuring the consistency of key kernel structures. The system must provide mechanisms that allow critical areas of the kernel code to be "owned exclusively" by only one of the system processors. the paper will examine ways in which this can be achieved in software only, and by a combination of software and hardware. It will also look at how the the different methods of "interlocking" between the processes running on different processors can affect the overall performance of the complete system.

The paper will conclude with a review of the author's experience in running multiprocessor systems in a variety of UNIX application environments, with particular emphasis placed on the differing performance characteristics observed for each type of application load.

CommonView, a Windows Library in C++

Alan Sloane
Glockenspiel Ltd.
19 Belvedere Place
Dublin
IRELAND

CommonView is a class library with the following objectives:

— It must run on OS/2, X11, NeWS and possibly Macintosh

— It must present the same programmer interface on each system

— It must respond to events within specified budgets

— It must support light weight processes and parallel systems

— It must be extensible and scalable, using inheritance

— It must support persistent objects and dynalinking

The paper describes how these objectives shaped the architecture of CommonView, and enumerates important problems encountered in moving from system to system.

Network Services in the Athena Environment

† The work reported in this project was partially supported by the Esprit programme, sponsored by the CEC. It took place within the framework of Esprit 231.

Jennifer G. Steiner
Daniel E. Geer, Jr.
Project Athena
Massachusetts Institute of Technology
USA
steiner@athena.mit.edu
geer@athena.mit.edu

As a prerequisite for achieving its mission of fostering educational innovation at MIT, Project Athena must support a large network of independently owned and controlled workstations. At the Project's inception, systems software to support such a configuration did not exist. As a result, a large part of the systems development effort during Athena's first five years has been devoted to the design and implementation of network services to fill this need. This paper describes the use of network services in the Athena environment, including three new systems level services developed at Athena: the Kerberos authentication service, the Service Management System, and the Hesiod name service.

AUSTRIA - UUGA
Friedrich Kofler
Schottenring 33/Hof
A-1010 Vienna
AUSTRIA
+43 222 34 61 84
uuga@tuvie.at

BELGIUM - BUUG
Marc Nyssen
Department of Medical Informatics
VUB, Laarbeeklaan 103
B-1090 Brussels
BELGIUM

DENMARK - DKUUG
Mogens Buhelt
Kabbelejevej 27B
DK-2700
Bronshoj
DENMARK
+45 1 60 66 80
mogens@dkuug.dk

FINALND - FUUG
Johan Helsingius
OY Penetron Ab
Box 90
02201 ESPOO
FINLAND
+358-0-427 632
julf@fuug.fi

FRANCE - AFUU
Miss Ann Garnery
AFUU
11 Rue Carnot
94270 Le Kremlin-Bicetre
FRANCE

HUNGARY - HUUG
Dr Elôd Knuth
Computer and Automation Institute
Hungarian Academy of Sciences
H-1502 Budapest 112, P.O. Box 63
HUNGARY
+36 1 665 435

ICELAND - ICEUUG
Marius Olafsson
University Computer Center
Hjardarhaga 4
Reykjavik
ICELAND
+354-694747
iceuug@rhi.hi.is

IRELAND - IUUG
John Carolan
Glockenspiel Ltd
19 Belvedere Place
Dublin 1
EIRE
+353 1 364515
john@puschi.uucp

ITALY - i2u
Ing Carlo Mortarino
i2u
Viale Monza 347
20126 Milano
ITALY

NETHERLANDS - NLUUG
Patricia Otter
Xirion bv
World Trade Centre
Strawinskylaan 1135
1077 XX Amsterdam
THE NETHERLANDS
+31 20 6649411
patricia@xirion.uucp

NORWAY - NUUG
Jan Brandt Jensen
Unisoft A.S.
Enebakkvn 154
N-0680 Oslo 6
NORWAY

SWEDEN - EUUG-S
Bjorn Eriksen
SUNET/KTH
S-100 44 Stockholm
SWEDEN
+46 8 790 65 13
ber@kth.se

UNITED KINGDOM - UKUUG
Bill Barrett
Owles Hall
Buntingford
Hertfordshire SG9 9PL
UNITED KINGDOM
+44 763 73039

The European UNIX systems User Group
Owles Hall
Buntingford
Hertfordshire SG9 9PL
UNITED KINGDOM
+44 763 73039