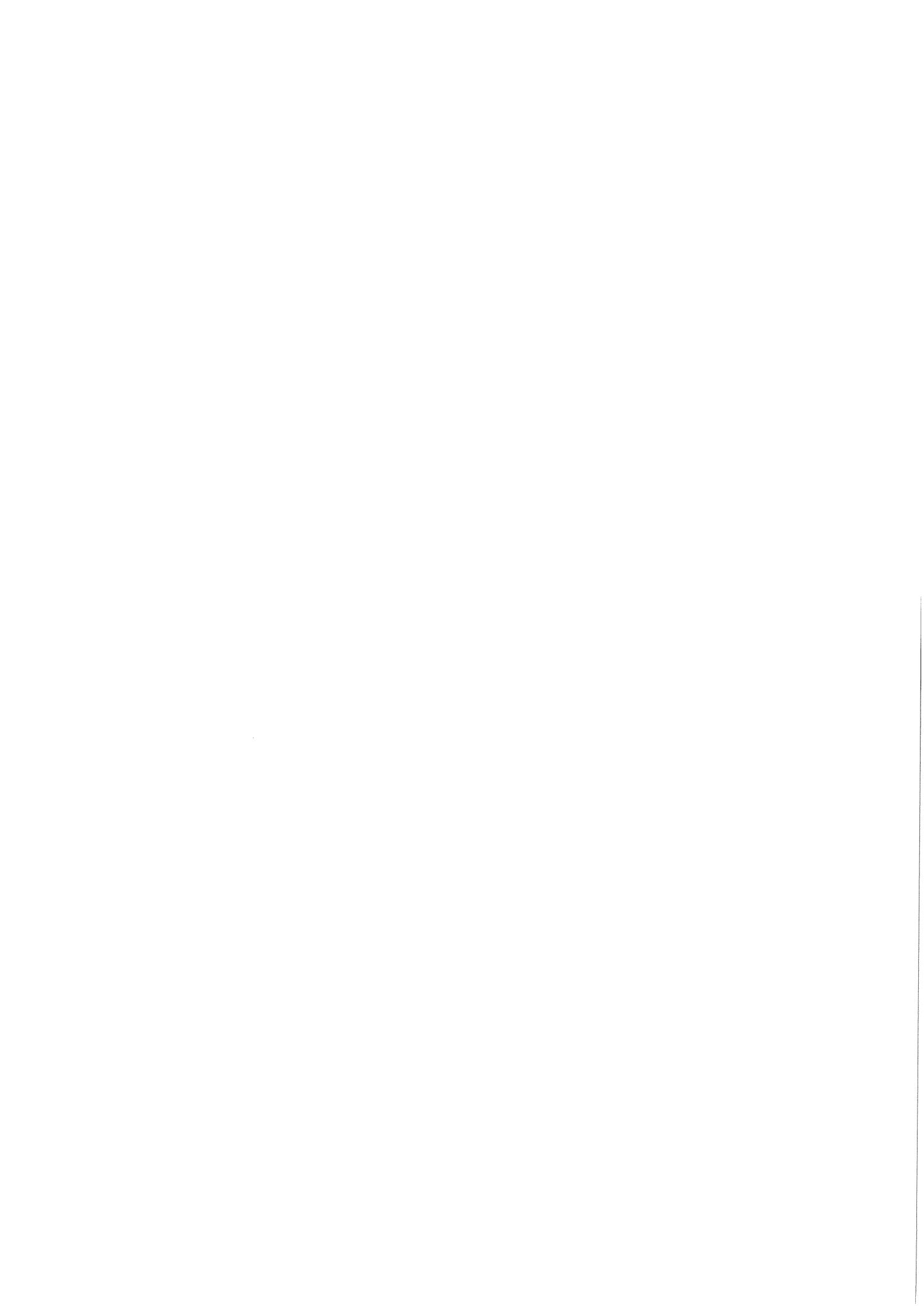


Australian UNIX systems User Group Newsletter

AUUGAN

Volume 10, Number 2

April 1989



The Australian UNIX* systems User Group Newsletter

Volume 10 Number 2

April 1989

CONTENTS

AUUG General Information	3
Editorial	4
President's Letter	6
Secretary's Letter	7
Call for Papers - AUUG '89	8
Adelaide UNIX Users Group Information	10
Western Australian UNIX systems Group Information	11
AUUG Institutional Members	12
USENIX San Diego Proceedings Offer	14
Structured Query Language (SQL) Shell	17
Multi-User Machine Benchmarking	22
Sun's Response	29
From the EUUGN Newsletter - Volume 8 Number 4	33
Changing the *roff Escape Character	34
Optical Disk WORM File System under System V Rel 3.0	38
Portugal EUUG Conference Report	45
Competition Result	49
A First Visit to an EUUG Conference	50
The German EUnet - Dnet	52
The C and UNIX Dictionary	54
Hungarian UNIX Users Group	55
Yugoslav UNIX Users Group	56
News from the Netherlands	57
The Danish Connection	60
The UKUUG National Group	63
EUNET in Belgium	66
USENIX Association News for EUUG Members	71

From the EUUGN Newsletter - Volume 8 Number 4 <i>continued</i>	73
SVR4 Conference London	73
ANSI C Standard and Progress towards an ISO C Standard	75
The POSIX Standard and Its Future Development	77
AT&T System V/MLS	79
UNIX Clinic	83
The European E-Mail Directory	86
EUUG Portugal Conference Abstracts	87
From the EUUGN Newsletter - Volume 9 Number 1	97
A port of the MINIX Operating System to the Atari ST	98
AUUG Conference Report	111
Journees UNIX	119
EUUG Executive Report	123
Dutch User Group Report	125
AFUU Diary	126
UKUUG News	128
Calendar of UNIX Events	130
Introduction to Window Systems	133
Puzzle Corner	139
Software Review	140
USENIX Association News for EUUG Members	157
UNIX Clinic	159
EUnet	165
UNIX is Chauvinistic	168
Overview of UNIX System V Release 4.0	170
Book Review	175
UKUUG Winter Abstracts	177
AUUG Newsletter Back Issues	180
AUUG Membership Catorgories	181
AUUG Forms	183

Copyright © 1989. AUUGN is the journal of the Australian UNIX* systems User Group. Copying without fee is permitted provided that copies are not made or distributed for commercial advantage and credit to the source must be given. Abstracting with credit is permitted. No other reproduction is permitted without prior permission of the Australian UNIX systems User Group.

* UNIX is a registered trademark of AT&T in the USA and other countries.

AUUG General Information

Memberships and Subscriptions

Membership, Change of Address, and Subscription forms can be found at the end of this issue.

All correspondence concerning membership of the AUUG should be addressed to:-

The AUUG Membership Secretary,
P.O. Box 366,
Kensington, N.S.W. 2033.
AUSTRALIA

General Correspondence

All other correspondence for the AUUG should be addressed to:-

The AUUG Secretary,
P.O. Box 366,
Kensington, N.S.W. 2033.
AUSTRALIA

AUUG Executive

President	Greg Rose <i>greg@softway.sw.oz</i> Softway Pty. Ltd., New South Wales	Secretary	Tim Roper <i>timr@labtam.oz</i> Labtam Limited, Victoria
Treasurer	Michael Tuke <i>mjt@conjure@labtam.oz</i> Vision Control Australia Victoria		
Committee Members	Frank Crawford <i>frank@teti.qhtours.oz</i> Q.H. Tours, New South Wales	Richard Burridge <i>richb@sunaus.aus.oz</i> Sun Microsystems Australia New South Wales	
	Chris Maltby <i>chris@softway.sw.oz</i> Softway Pty. Ltd., New South Wales	Tim Segall <i>tim@hpausla.aso.hp.oz</i> Hewlett Packard Australia, Victoria	

Next AUUG Meeting

AUUG89 Conference and Exhibition, will be held at the Sydney Hilton Hotel from Tuesday 8th to Friday 11th August 1989. A Call for Papers is printed in this issue. Further details will be provided in future issues.

AUUG Newsletter

Editorial

Welcome to the Newsletter.

As many of you may realize, the last issue of the AUUGN - February 1989, arrived in the mail in April, nearly two months after it was compiled by your AUUGN Editor. My own personal copy, as of the time of writing is still caught in the post. Obviously, the AUUGN printing and distribution is still experiencing teething problems even a year after the printing was moved to Melbourne. The distribution point has now been changed to the Melbourne Mail Centre which I am assured will make delivery more reliable. I can only apologise for delay and attempt to make sure it is not repeated.

In this issue there are several interesting articles. The domestic contributions are from Jack Dikian concerning a SQL shell, and Steven Bodnar about Multi-user benchmarking.

This month I have reprinted the contents of the latest issues of the EUUGN. One thing that caught my eye was Jim Reid's comprehensive report of AUUG 88.

Also the AUUGN has in this issue, is an USENIX Proceedings Offer and the Call for Papers for AUUG 89. AUUG 89 is not very far way in August so think about writing a paper and/or attending.

Nomination for AUUG Committee and election time is upon us. I ask you to consider standing for committee, nominating someone or at LEAST vote. Forms will be in the mail soon.

Since Tim Roper, the AUUG Secretary has organised that you are now sent reminders to renew your AUUG Membership, I do not have to tell you about highlighted labels anymore.

Finally, by the time you read this, I will have left my position at Webster Computer and have started at Labtam on Monday the 15th of May 1989. Please note the change in AUUG Editor's address.

AUUGN Correspondence

All correspondence regarding the AUUGN should be addressed to:-

John Carey
AUUGN Editor
Labtam Information Systems Pty. Ltd.
P.O. Box 297
Mordialloc Victoria 3195
AUSTRALIA

ACSnet: john@labtam.oz

Phone: +61 3 587 1444

Contributions

The Newsletter is published approximately every two months. The deadline for contributions for the next issue is Friday the 16th of June 1989.

Contributions should be sent to the Editor at the above address.

I prefer documents sent to me by via electronic mail and formatted using *troff -mm* and my footer macros, troff using any of the standard macro and preprocessor packages (-ms, -me, -mm, pic, tbl, eqn) as well TeX, and LaTeX will be accepted.

Hardcopy submissions should be on A4 with 35 mm left at the top and bottom so that the AUUGN footers can be pasted on to the page. Small page numbers printed in the footer area would help.

Advertising

Advertisements for the AUUG are welcome. They must be submitted on an A4 page. No partial page advertisements will be accepted. The current rate is AUD\$ 200 dollars per page.

Mailing Lists

For the purchase of the AUUGN mailing list, please contact Tim Roper.

Back Issues

Various back issues of the AUUGN are available, details are printed at the end of this issue.

Acknowledgement

This Newsletter was produced with the kind assistance and equipment provided by Webster Computer Corporation.

Disclaimer

Opinions expressed by authors and reviewers are not necessarily those of the Australian UNIX systems User Group, its Newsletter or its editorial committee.

President's Letter

By the time this letter gets into print, nominations and possibly votes for the new committee of AUUG Incorporated will already be in. So this letter is a bit of a retrospective on the year just passed.

First I must say what an honour it has been to hold this position, and to be able to work towards the improvement of this group.

Like all people who are in this sort of a position, I think, there are times when I get frustrated as well. A number of things cause this frustration.

- All of us on the committee have great ideas to improve the group, but we must always temper them a bit to ensure that everything we do really *is* in the best interests of the membership as it exists.
- We don't always agree on the directions either...
- All of us, not least me, wish we had more time to put into the group, but we and our respective employers have to eat.
- There are always nagging feelings that more could have been done, if only more hands had been available. (This is a not-disguised-at-all plea for more volunteers for things.)

The last year has definitely represented progress for the group as a whole, by any metric. Some of the high points of the last year have been:

- An enormously successful annual conference in Melbourne.
- Formal incorporation of the group.
- Beginning to make more benefits available to the members, both individual and corporate, including software distributions, overseas conference proceedings, and the Nutshell Handbooks.

All of this couldn't have been done without the help of a large number of people. I hate trying to list all of the relevant names, because I will invariably miss out someone. Nonetheless I *will* try, but if I missed your name and you deserve thanks, please accept them from me and the whole group. Explicit thanks to:

Tim Roper	the Secretary, for extremely diligent work from that seat, and an amazing effort on last year's conference.
David Purdue	who has organised some of the importations this year, and for generally helping out.
Labtam	for turning a blind eye to the above efforts.
John Carey	the best newsletter editor for a long time. (John is retiring soon, too.)
Peter Barnes	already doing a good job on the program committee for AUUG '89.
The committee members	who have donated time and effort generally, and to their supporting institutions.
Wael Foda	who does the meeting organisation for the group so well (and gets paid for it, but many thanks anyway).

On the down side this year, a couple of things went awry too. The intended Summer Meetings again failed to get off the ground, for which I must accept some responsibility, and the membership of the group has not grown in the manner which I feel *anything* associated with UNIX should. The committee is learning though, and can pass on the experiences and techniques, so I feel we can look forward to further improvements in the next year.

Best regards, and see you all at AUUG'89.

Greg Rose,
President.

Secretary's Letter

Please give careful consideration to the Call For Papers for AUUG89, our 1989 Conference and Exhibition, which appears elsewhere in this issue. Peter Barnes of the University of Queensland is Chairing the Programme Committee and is responsible for the conference programme only. Please direct all enquiries regarding registration, accommodation, exhibition, sponsorship, etc. to ACMS on (02)3324066 and *not* to Peter.

The recent Inaugural Software Distribution and Nutshell book offers are definitely now closed. However, details of a new offer for proceedings from the recent USENIX conference in San Diego appear elsewhere in this issue. AUUG has flown in stocks for immediate delivery (on receipt of payment) at a lower price than individual copies sea mailed from the USA.

Institutional memberships continue to grow rapidly and currently stand at about sixty. The percentage increase in ordinary members is not so startling but is certainly positive! Since February expiring members have been mailed a reminder and renewal form during the month in or before which they expire. This will continue as standard procedure and should solve the problems with the previous method of highlighting the mailing labels. It already appears to have resulted in less memberships lapsing temporarily thereby missing an issue of AUUGN. Also, a catch-up mailing has recovered some old members whose membership had lapsed. Finally, we are about to mail to over 1000 previous conference and exhibition attendees and other suspected UNIX users with information about AUUG89 and membership of AUUG.

Not all copies of the last issue of AUUGN enclosed the promised flyer from USENIX giving details of its *Computing Systems* journal. Those who missed out last time should have received one with this issue.

Don't forget to keep 8th - 11th August free for Sydney!

Tim Roper

Call For Papers

AUUG '89

Australian Unix systems User Group

Conference and Exhibition 1989

August 8–11 1989, Sydney, Australia

Summary

The 1989 Conference and Exhibition of the Australian UNIX† systems User Group will be held on Tuesday 8th – Friday 11th August 1989 at the Hilton Hotel in Sydney, Australia. Tutorial sessions will be held on Tuesday 8th, and the conference proper from Wednesday the 9th to Friday the 11th. The conference theme is

No one ever got fired for buying UNIX.

AUUG is pleased to announce that the guest speakers will include:

Dennis Ritchie	Bell Laboratories
James Gosling	Sun Microsystems
Sunil Das	City University, London

Papers

Papers on topics related to commercial, government, and non-technical uses of UNIX are now invited. Some suggested topics include but are not restricted to:

- Databases, accounting and UNIX systems
- Reliability, availability, fault tolerance and UNIX
- Transaction processing on UNIX
- Secure UNIX
- UNIX in heterogeneous environments
- Tendering, sizing and comparing UNIX and non-UNIX systems
- The integrated office
- Administration of non-technical UNIX systems
- Look and Feel, graphical and non-shell interfaces

Papers on other (non-commercial or technical) aspects of the UNIX system are also sought.

Authors of papers presented at the conference will receive complimentary admission to the conference and the dinner. AUUG will again hold a competition for the best paper by a full time student at an Australian educational institution. The prize for this competition will be an expense paid return trip from within Australia to the conference to present the winning paper. A cash prize in lieu of this may be paid at the discretion of AUUG. Students should indicate with their abstract whether they wish to enter the competition. AUUG reserves the right to not award the prize if no entries of a suitable standard are forthcoming.

† UNIX is a trademark of Bell Laboratories.

A special issue of the group's newsletter AUUGN containing the conference proceedings will be printed for distribution to attendees at the conference.

Acceptance of papers will be based on an extended abstract and will be subject to receipt of the final paper by the due date. Abstracts and final papers should be submitted to the programme committee chair:

Peter Barnes	Phone:	International	+61 7 3772907
AUUG 89		National	07 3772907
Computer Science	Fax:	International	+61 7 3710783
University of Queensland		National	07 3710783
St. Lucia	Telex:	UNIVQLD AA40315	
Queensland 4067	ACSnet:	pdb@uqcspe.cs.uq.oz	
Australia	UUCP:	uunet!munnari!uqcspe.cs.uq.oz!pdb	
	ARPA:	pdb%uqcspe.cs.uq.oz@uunet.uu.net	

Final papers may be sent via electronic mail and formatted using *troff* and any of the standard UNIX macro and preprocessor packages (*-ms*, *-me*, *-mm*, *pic*, *tbl*, *eqn*) or with TeX or LaTeX. Alternatively, final papers may be submitted as camera ready copy on A4 paper with 35mm margins left at the top and bottom. Intending authors unable to produce either of these forms are requested to contact the programme committee chair.

Timetable

Receipt of Extended Abstracts	Monday 22nd May
Letters of Acceptance Sent	Monday 12th June
Receipt of Final Papers	Monday 3rd July
Conference tutorials	8th August
Conference and Exhibition	9th–11th August

Adelaide UNIX Users Group

The Adelaide UNIX Users Group has been meeting on a formal basis for 12 months. Meetings are held on the third Wednesday of each month. To date, all meetings have been held at the University of Adelaide. However, it was recently decided to change the meeting time from noon to 6pm. This has necessitated a change of venue, and, as from April, meetings will be held at the offices of Olivetti Australia.

In addition to disseminating information about new products and network status, time is allocated at each meeting for the raising of specific UNIX related problems and for a brief (15-20 minute) presentation on an area of interest. Listed below is a sampling of recent talks.

D. Jarvis	"The UNIX Literature"
K. Maciunas	"Security"
R. Lamacraft	"UNIX on Micros"
W. Hosking	"Office Automation"
P. Cheney	"Commercial Applications of UNIX"
J. Jarvis	"troff/ditroff"

The mailing list currently numbers 34, with a healthy representation (40%) from commercial enterprises. For further information, contact Dennis Jarvis (dhj@aegir.dmt.oz) on (08) 268 0156.

Dennis Jarvis,
Secretary, AdUUG.

Dennis Jarvis, CSIRO, PO Box 4, Woodville, S.A. 5011, Australia.

PHONE: +61 8 268 0156 UUCP: {decvax,pesnta,vax135}!mulga!aegir.dmt.oz!dhj
 ARPA: dhj%aegir.dmt.oz!dhj@seismo.arpa
 CSNET: dhj@aegir.dmt.oz

WAUG

Western Australian UNIX systems Group

PO Box 877, WEST PERTH 6005

Western Australian Unix systems Group

The Western Australian UNIX systems Group (WAUG) was formed in late 1984, but floundered until after the 1986 AUUG meeting in Perth. Spurred on by the AUUG publicity and greater commercial interest and acceptability of UNIX systems, the group reformed and has grown to over 70 members, including 16 corporate members.

A major activity of the group are monthly meetings. Invited speakers address the group on topics including new hardware, software packages and technical dissertations. After the meeting, we gather for refreshments, and an opportunity to informally discuss any points of interest. Formal business is kept to a minimum.

Meetings are held on the third Wednesday of each month, at 6pm. The (nominal) venue is "University House" at the University of Western Australia, although this often varies to take advantage of corporate sponsorship and facilities provided by the speakers.

The group also produces a periodic Newsletter, YAUN (Yet Another UNIX Newsletter), containing members contributions and extracts from various UNIX Newsletters and extensive network news services. YAUN provides members with some of the latest news and information available.

For further information contact the Secretary, Skipton Ryper on (09) 222 1438, or Glenn Huxtable (glennt@wacs.vax.uwa.oz) on (09) 380 2878.

Glenn Huxtable,
Membership Secretary, WAUG

AUUG Institutional Members

ACUS / UNISYS
Aldetec Pty Ltd
Australian National University
Australian Nuclear Science & Technology Organisation
Australian Wool Corporation
Autodesk Australia P/L
BHP Melbourne Research Labs
Ballarat Base Hospital
Basser Department of Computer Science
CSIRO DIT
CSIRO Division of Manufacturing Technology
Centre for Information Tech & Comms
Civil Aviation Authority
Comperex (NSW) Pty Ltd
Computer Software Packages
Cybergraphic Systems Pty Ltd
DBA Computer Systems Pty Ltd
Data General
Department of Industry Technology and Commerce
Dept of Industry, Technology and Resources, Victoria
Dept of Lands - Central Mapping Authority
Elxsi Australia Ltd
Flinders University, Discipline of Computer Science
Fujitsu Australia Limited
Gould Electronics Pty Ltd
Great Barrier Reef Marine Park Authority
Harris & Sutherland Pty Ltd
Hewlett Packard Australia Limited
Honeywell Information Systems
Ipec Transport Group
Lands Department, Qld
Lionel Singer Corporation
Macquarie Bank Limited
Macquarie University
Monash University Computer Science
NEC Information Systems Australia Pty Ltd
NSW Parliament
National Engineering Information Services P/L
Nixdorf Computer Pty Limited
Olivetti Australia Pty Ltd
Olympic Amusements P/L
Overseas Telecommunications Corporation
Prentice Computer Centre
Prime Computer Research & Development
Prime Computer of Australia Ltd

AUUG Institutional Members

Q. H. Tours Limited
Qld State Govt Computer Centre
Racecourse Totalizators Pty Ltd
Reark Resources
SEQEB
Sigma Data Corporation Pty Ltd
South Australian Institute of Technology
Sphere Systems Pty Ltd
State Library of Tasmania
Sun Microsystems Australia
Tattersall Sweep Consultation
University of Adelaide
University of Melbourne
University of New England
University of Technology Sydney - Computing Services Division
University of Wollongong
Webster Computer Corporation
Yartout Pty Ltd

USENIX San Diego Proceedings

Timothy Roper

Secretary, AUUG Inc.

Description

AUUG has acquired fifty copies of the proceedings from the *1989 Winter USENIX Technical Conference* (San Diego, January 30 – February 3, 1989). They are available at cost to members only on a first-come-first served basis. We may order a further shipment but that would not be done until sufficient excess orders are received so you should act quickly to avoid delay. The cost per copy is \$39 plus \$6 post and packing (surface mail within Australia).

Ordering Details

Orders must be on the attached order form signed by a member of AUUG. In the case of an Institutional member it should either be signed by the Administrative Contact (the person who signed the current membership form) or stamped and signed by a representative of the institution.

Orders from non-members will be accepted only if they are accompanied by a completed membership application form with payment. Membership information and application forms may be found in a recent copy of the association's newsletter AUUGN or obtained by mailing timr@labtam.oz or the postal address below.

The form with payment should be sent to:

AUUG Inc
PO Box 366
Kensington NSW 2033
Australia

Terms

Payment (cheque or money order) for copies of the proceedings and for membership must be enclosed. Purchase orders will only be accepted from Institutional members.

SAN DIEGO PROCEEDINGS

ORDER FORM

Contact Details:

Name:

Phone:

Fax:

Net Address:

Postal Address:

Shipping Address:

Number of Copies @ \$45 each: Total Cost:

Membership Details:

Name of Member:

Category of Membership:

Ordinary/Student/Institutional/Hon Life

Signature:

Name (please print):

Softway Products . . .

- **ACSnet** - network of unlike machines, and has a purpose and function similar to that provided by the UUCP network in wide use in most of the UNIX world. SUN III is the software package used to build the local or wide area network. This enables the systems to exchange electronic mail, data files, data base queries or any binary information.
- **MacIDRIS** is a fast and compact operating system compatible with the POSIX standard for UNIX-like systems. MacIDRIS has a rich set of UNIX-like utilities for file and text manipulation. The entire system runs as an application on a standard Macintosh SE with hard disk. The implementation allows the MacIDRIS user to alternate between the Macintosh operating system and the UNIX-like environment.
- **52-Backup** provides a "no hassle" backup facility which will make life alot simpler for the UNIX system administrator. 52-Backup includes a unique, innovative, algorithm allowing the "incremental" backup of large files, such as databases. This enables daily backups of (say) 100Mb databases onto smaller (40Mb) tapes. Backups can be scheduled at any frequency desired. Installation, configuration and operation are straight forward as the package was designed for the novice system user.
- **BENCHMARKING** - choosing between a number of computers it is necessary to know whether a computer will provide adequate response under a certain type of workload. Softway will examine the implementation of the computer and its associated system software.
- **COURSES** - "*Beginner's UNIX Workshop*" a three day "hands-on" presentation aimed at first time UNIX users. "*A Fast Start to the UNIX Operating System*" two day course aimed at non-technical personnel. "*UNIX System Administration*" presented over five days, takes an in-depth look at the role of the System Administrator under UNIX.

For further information please contact Elaine Pensabene on (02) 698 2322 or ACSnet elaine@softway.

Softway Pty Limited (Incorporated in NSW) 79 Myrtle St, Chippendale NSW 2008
✉ PO Box 305, Strawberry Hills, NSW 2012, Australia. ☎ +61 2 698 2322 fax: +61 2 699 9174

Structured Query Language (SQL) Shell

Jack Dikian

Q.H. Tours Ltd.
141 Walker St. North Sydney NSW
jack@teti.qhtours.oz

ABSTRACT

The *SQL+sh* is an interactive front-end to Unify's Structured Query Language (SQL). It's main purpose is to add Csh/tenex like functionality to a vanilla query interpreter in the way of *SQL*. A query history stack, ability to recall and edit previous queries as well as an interactive RECORD and FIELD name recognition and completion mechanism are a sample of the sort of enhancements *SQL+sh* supports. This paper presents a brief background to *SQL* before discussing some of the features we included to this package.

Working in an environment where a significant portion of a programmers time is spent writing and maintaining applications software around the Unify relational database; any facility that simplifies database interactions must be an advantage. This database is quickly approaching the 2 G-byte mark with over 300 Mb of supporting software. Like other large database users, the overhead of database related maintenance is a significant consideration. Improvements in database related utilities greatly increases productivity as well as reliability.

One of the most powerful facilities available to the maintenance programmer in our environment is Unify's Structured Query Language *SQL*. This utility is often used to interrogate as well as patch the underlying database. Ad hoc *SQL* queries are often generated to confirm the correctness of application modules as well as serving the more simple day to day user information requirements.

A Quick Look At SQL

SQL is an English keyword orientated query language of great flexibility. It is a language that is easy enough for non-programmers to learn, yet has enough power for data processing professionals. This product was originally defined by Chamberlin and others at the IBM Research Laboratory in San Jose, California, under the name System R. A family of IBM products based on the System R technology was developed. These products are now generally available and are known as DB2, SQL/DS and QMF [1]. A number of other vendors have also produced systems that support *SQL*. *SQL*'s data manipulation statements typically operate on entire sets of records. For example, the **select** and **update** clauses can retrieve and modify a set of values and tables. *SQL*, like all relational data manipulation languages is a set-level language. For this reason, *SQL* is often described as a nonprocedural language. The user specifies "what" data they want and not so much "how" to get it. It is up to *SQL* to decide on how best to execute any particular query. It needs for example to consider which tables are being referenced in any request; the size of the tables; what indexes exists; how selective those indexes are and of course, the form of the **where** clause.

SQL queries consist of clauses, each of which is preceded by a keyword. Examples of keywords include; **select**, **update**, **delete** and **insert**. In fact, the previous four keywords all belong to that part of *SQL* which is commonly referred to as the *DML* or *Data Manipulation Language*. Other optional keywords are used to control, format and operate on the various queries. Some simple examples of queries are given below:-

```
> select Name, Phone  
> from PERSONS  
> where Age > 30/
```

The above example illustrates the selecting or retrieving of the specified fields Name and Phone from a specified table PERSON where some specified condition is true. It is important to note that the result of the query is another table.

```
> select PERSON.* , COMPANY.*  
> from PERSON, COMPANY  
> where PERSON.PName = COMPANY.CName/
```

This example demonstrates the retrieving of data from two tables namely PERSON and COMPANY. We are interested in all instances of the field PName in PERSON matching the field CName in the table COMPANY. This is commonly referred to as "Joining" two or more tables. The availability of the join operation is, almost more than anything else, that distinguishes relational from nonrelational systems.

```
> update COMPANY  
> set CName = 'ACME'  
> where Type = 'ACTIVE'/
```

This example sets the field CName to "ACME" for all COMPANY records that satisfy the condition after the where clause.

The SQL+sh

Our main database currently supports over a 100 tables and close to a 1000 fields. Using *SQL* to interrogate and manipulate data in this environment almost always requires the programmer to first browse through the Database schema listing. This is not only due to the large number of different tables and fields but is also due to UNIFY's record and field naming conventions. The maximum length of a record name is eight characters. It is therefore impossible to create two records with the names "PROGRAMMER" and "PROGRAMME". A compromise may lead to the names "PROGMR" and "PROGME" etc. It is easy to see why the schema listing may be required in such cases.

Creating tables in Unify requires the user to nominate both a short and a long field name. Short field names must begin with a letter and can be up to eight characters long. The long field names begin with a letter and can be up to sixteen characters long. It is the long name that *SQL* requires for carrying out queries. The schema is used to determine or look up this long name. The schema is also used to determine relationships between tables and their corresponding fields.

Editing large queries are handled by - *SQL* writing the last query in /tmp. The edit facility invokes a standard editor such as vi with the last query loaded in the editor buffer. The user modifies and saves the changes before using the restart clause to re-execute the query. Although this facility is useful, it is however often tedious. This is especially true when a simple typo needs to be repaired. Because only the last query is effectively saved, access to previous queries are lost unless the user explicitly saves the editor buffer to a nominated file.

Interestingly, we required in *SQL* a similar transformation in functionality as that provided by say csh and tcsh over the bourne shell. Where tcsh provides file name recognition and completion, we required record and field name recognition and completion. Where csh provides a history and edit facility for commands, we required a history and edit mechanism for queries. In implementing some of the ideas found in csh and tcsh, we were able to address both the above mentioned shortcomings as well provide a much more effective user interface.

Not having access to *SQL* source, the only other alternative in implementing the above changes was to write our own parser sitting on top of *SQL*. This would simply read the input stream, decide if it needs to do anything special like manipulate the history stack, carry through edit commands, expand alias' etc and then write to *SQL* via a pipe. The output of *SQL* is not altered.

SQL+sh reads a schema description file on startup. This file is typically generated by the systems administrator by running a specially written shell script. The description file describes the database tables, their respective fields and other information such as field type and length. The shell script uses *SQL* to dump the relevant table, field types and names.

On startup, *SQL+sh* looks at the environment variable DBPATH and displays the name and address of the working database. After this point, *SQL+sh* enters a for-ever loop waiting for queries, internal commands and or the quit or end clause. A new prompt including the event number is displayed. An environment variable defines the maximum history size. An internal command has been added called " Mod On/Off" which enables and disables the availability of non-passive *SQL* clauses. For example, after entering the command " Mod Off", such clauses as delete, update, insert etc are disabled or ignored. This is useful in cases where support staff use *SQL* to answer quick telephone queries and should not update the database inadvertently or otherwise.

Unlike Unix commands which are newline terminated, *SQL* queries often span over many lines. In fact, users of *SQL* are encouraged to use good formatting procedures when making *SQL* queries. This is in part due to the fact that quite complex *SQL* scripts can be written and saved for regular use. These scripts are also used to feed data to Unify's report generator *RPT*. The "/" character is used to indicate the end of a query. For this reason, *SQL+sh* supports a modified history substitution command in the way of "!event+". This tells *SQL+sh* to re-execute the query beginning with the event number "event" and continue to re-execute events forward in the stack until a "/" character is encountered. All other normal history substitution commands such as "!!", "!.number", "?number" as well as "?pattern" etc have been implemented. Where a query spans many lines, *SQL+sh* collects together the individual clauses to echo a single event in its history stack.

Editing previous queries are handled two ways. The standard *SQL* procedure is to invoke the system editor with the last query loaded into the editor buffer. The edit clause facilitates this procedure. This method is still available and is usually used for editing large query texts. This method allows only the last query to be modified and re-executed. *SQL+sh* introduces the csh like "?event s/pattern1/pattern1" and ^pattern1^pattern2^ mechanisms. These are extremely convenient for repairing typos and or for substituting record or field names while leaving the general structure of the query untouched.

One of the most useful additions to *SQL* was the introduction of record and field name recognition and completion. The idea here was to provide a convenient way to avoid having to look up the record and field names before generating queries. Automatically displaying field types and length was considered useful. Other considerations included providing a means by which key strokes could be reduced and accurately associating relevant field names to their correct parent tables. This mechanism is used in conjunction with the database schema description file. It is no longer necessary to type a complete record or field name. Only a unique abbreviation is necessary. Typing the ESCAPE key after the abbreviation will complete the record or field name, echoing the full name. Unlike tcsh, where there is really only one type of file name completion, *SQL+sh* needs to consider context and determine whether a record, or field name is being sought.

This is achieved by adding some of the *SQL* syntax rules into *SQL+sh*. For example the following grammar extracts define the syntax for the insert and select clauses:-

insert into RECORD [(FIELD,...)]: from filename<constant>| select/

```
select ["unique"] | * | RECORD.* | RECORD.FIELD | FIELD ...,
           | * | RECORD.* | RECORD.FIELD | FIELD ...,
from | RECORD [label] |, ...
where ["not"] | FIELD | RECORD.FIELD | constant ETC.
```

SQL+sh tries to carry out a search of either the appropriate record or field based on the position the ESCAPE key was pressed in the input stream. It is obvious from the above two syntax examples that it is not often possible to determine whether a RECORD or a FIELD needs expanding. In the select clause for example, it is possible to say " select record.field from ..." or " select field from...". Hitting the ESCAPE key just after the select token leaves *SQL+sh* with a choice of searching for appropriate records or fields. In fact, in this particular example, the

system will first search through the record list and then the field list. In general, as each word is read, *SQL+sh* updates a flag indicating whether it is in a "RECORD" or "FIELD" state. This flag is initially set to a "NULL" state thus causing a bell to sound when the ESCAPE key is pressed. A "BOTH" state causes *SQL+sh* to search records and then fields. This state is established by tracking entered words against various syntax rules defined in *SQL+sh*. We have also provided a means of commenting query text. Text found inclosed within the "{" and "}" braces are ignored. This facility was implemented in order to allow a clean method of displaying field types and length in-line. On Hitting ESCAPE in a "FIELD" state, the system will not only display a candidate field name but also place the relevant field type and length already commented.

Besides providing a recognition and completion mechanism, *SQL+sh* also provides a facility where fields belonging to a particular record can be scanned. For example, after having typed in the sub-clause **select * from PERSON where "** it is possible to Hit **Ctrl-f** to echo the first field belonging to the PERSON record. Hitting **Ctrl-f** again will replace the first displayed field name with the next field. When the list of fields are exhausted, the process is repeated. This allows the user to carry out a query on a record even when they had no idea of the field names associated with the given record. The field type and length is once again displayed in comments. Some examples follow:-

```
select * from PE<ESC>"  
                                results in  
select * from PERSON.
```

The cursor sits at the next column position waiting for the rest of the query.

```
select * from PERSON where <Ctrl-f>  
                                results in  
select * from PERSON where PName {STRING 12}  
  
                                Hitting <Ctrl-f> again results in  
  
select * from PERSON where Paddress {STRING 45}
```

The user can now enter the rest of the query

```
select * from PERSON where Paddress {STRING 45} = 'Bag End*' and  
  
                                Hitting <Ctrl-f> here results again
```

```
PAge {NUMERIC 3}  
                                Now we can enter the rest of query
```

```
PAge {NUMERIC 3} <= 111/
```

Often there is the need to carry out repetitive queries involving tests against large text constants such as "0 081 1234678905 0" and "Speak Friend And Enter". An ability to implement a concept of **macros** was also considered a useful enhancement. The same query is often re-executed many times over in the event of a Database maintenance session. One or more parameters in the query may however vary. An ability to expand VMS like "Logical Variables" was added to *SQL+sh*.

The same variable setting and expansion mechanism is used to set and unset simple and complex variables. There is no inherent differences between variable substitution and macro processing. The difference is operational. *SQL+sh* maintains a set of variables each of which has as a value a list of zero or more words. Each word in this list could be a simple constant or another variable. This value may be displayed and changed by using the internal commands **show** and **clear**. After the input line is parsed, and before each query is executed, variable substitution is performed. Variables are keyed by '\$' character. The expansion can be prevented by preceding the

'\$' with a `\' except within `'''s. A Macro with a single argument can be seen as a variable containing another variable in its assignment string. The second variable has to be resolved before the macro can be executed. Newline characters found in the assignment list are ignored. Looping is prevented by checking that the same variable does not appear in the assignment list of that variable. Examples of variables follow:-

```
[1] $new_name = "Bilbo Baggins"  
[2] $my_update = " update PERSON  
[3] set PName = $new_name  
[4] where PName = '*'/  
[5] $my_update
```

Where Is It At.

We have been using this utility on a trial basis for the last few weeks. In general, the added convenience of query recall and edit far exceeds the cost of the extra overhead (minimal anyway). The ability to echo the field length and type results in much less references made to the schema listing. Record and field name completion means less typos in general. Although much of this utility came about after a "wouldn't it be nice..." chat and a couple of very long editing sessions (a few man days), the final thing has proved to be very useful in our environment.

References

- [1] C.J. Date, "An Introduction To Database Systems", Addison-Wesley 1986.

Multi-User Machine Benchmarking

Steven Bodnar

11/10/88

1. Introduction

The role of benchmarking is to facilitate the choice of equipment based on some criteria. It is not to be the sole measure used for selection. Issues such as price, performance, maintenance, software quality, human interface design and quality of support are very important.

The end result of buying a machine is that it should be able to perform all functions required, at the lowest cost. To test this you must have a realistic idea of what your machine is in for. No use buying a machine based on graphics speed if it will be used for general purpose research. Too often people buying machines only include features they would like to see and not take into account the real workload the machine will experience. For example graphics speed is important if its main function is graphics work, but if it is also a general vehicle for research involving AI, database or numerical work then the machine might end up being inadequate.

There are many benchmarking programs available, and many variables that can be tested for. Too often only component hardware and software items are benchmarked resulting in a poor choice in equipment. It is no use testing CPU speed or disk speed, unless it is in the context of a more realistic benchmark.

2. Multi-User Benchmarking

Let us take an example of a machine dedicated to the teaching of students the language Pascal. To have the best machine you would want a choice of machines running the course, and then choose the best machine after experiencing supporting the machine for the year. This might sound extreme but highlights an ideal situation that should at least be approximated.

An approximation would be to simulate the workload expected on the machine. That way we can get an overall picture. No use having high performance components when there are design flaws or bottlenecks, reducing overall performance to that of other machines that have lower performance components.

One freely available benchmark that can simulate a multi-user environment is MUSBUS. It is a multi-user benchmark, which also performs more specific benchmarks to highlight bottlenecks. Another use of MUSBUS which is often overlooked is the testing of the UNIX port. On a good port of UNIX, it should compile and run with no problems. Often a number of C compiler and more general bugs can be picked up and gives a first order test of the port.

MUSBUS allows a user to supply their own workload, the default using a combination of ls, cat, cc, rm, chmod, grep, cp and ed. Although this will give an indication of multi-user performance, it will more than likely not reflect workloads you will expect. In the example of the Pascal course, a workload using the Pascal compiler on assignments that will be encountered in the course, with associated compile errors and runs should be used. Some Pascal compilers have been known to be very memory and cpu intensive compared with the C compiler, so the default workload can give misleading results.

If using large software packages are a necessary point, then use them in the benchmark. No use testing the speed of drawing a horizontal line, when you really want to test a large CAD/CAM package. If you have to port the software then all the better. As many significant packages should be ported to the machines in question when you have them on loan. This will answer useful questions such as:

- 1 Are the main packages runnable on the new machine?

- 2 How buggy is the compiler and UNIX system in general? This is important, as you will have to work around these with other projects. If the vendor cannot fix it straight away the safest approach is assume it will never be fixed.
- 3 Does the interface to the machine, make it easy to use? Windowing software comes in many flavors, allowing the people who will be using them to use them now, will highlight problems.
- 4 How much knowledge and support does the vendor supply in solving a problem? If they actually solve the problem, then you can count yourself as lucky. Any thing less is inadequate. If they are running you around and constantly checking with their overseas parent company, who are slow to reply or do not reply at all, then you should reconsider your position. If they constantly give the impression of incompetence or are caught lying then consider rejecting the vendor. Too often people enter contracts knowing that they or their technical people have grave reservations about the vendor's technical personal, and then get burnt. If you still think it is worth the risk then the majority of your time should be spent in developing an iron clad contract and be prepared for legal action.

When comparing machines it is imperative that the one person actually runs the benchmarks on the machines that are to be chosen from. This gives the one person a very good overview of all the machines. If you cannot obtain a machine from the vendor of the appropriate configuration then this does not reflect well on the commitment and quality of the vendor.

3. Case Study 1

A large number students of about 100 to 150 on a Data General machine running UNIX. Problems with performance, reliability, support and the introduction of the Sun-4 suggested a reappraisal of equipment was needed, so MUSBUS was used to benchmark the systems. Costs of selling DG-2000s and of buying a Sun-4 suggested that we needed to sell four DG-2000s to buy a Sun-4. The configuration included 48 lines, no graphics monitor, 8 megabytes of main memory, and 280 megabytes of disk on a Xylogics 451 controller. This means the Sun-4 with up to 48 users had to be as fast as one DG-2000 with up to 12 users.

The results below were done with a standard workmix supplied with MUSBUS version 5.2 with the machines running in single user mode, not including processes normally used in multi user mode. There are a few problems with this benchmark that should be highlighted:

- 1 Single user mode MUSBUS benchmarks might not give a good indication of how the machine performs in multiuser mode, as you need to run the usual daemons that accompany a multi-user system. If it is running an early version of Berkeley 4.2 or earlier, or running System 5 with some Berkeley enhancements, most likely there will be daemons for telnet, rlogin, talk and many others. They take up resources such as file descriptors, process slots and swap space, often overloading a poorly configured system. Later Berkeley versions have 'inetd' and System 5 Release 3 has 'listen' which alleviates the problem. But you can never be sure what the vendor has done to improve/impede multi-user performance. Also some daemons are very cpu intensive such as rwhod.
- 2 Does not run a Pascal workload mix. In this case study benchmarks were initially not perceived by management as necessary at the time, and there was not an abundant amount of time either. It was deemed a necessary expediency to at least look at general performance.
- 3 The machines were not really matched in disk and memory requirements, but we were looking at an overall swap so it was not considered necessary.
- 4 The DG-2000 is marked crash, which really meant it ran into a kernel limit. It can be argued that better performances could have been obtained by tuning the kernel, but as that task is often a black art and the vendor would be in the best position to determine such parameters, it was deemed not feasible.

Single User Mode MUSBUS tests.

Figures are time taken in seconds.

machine	DG-2000 S4/260		
opt	FPA	FPA	
mem	4	8	
unix	UX3.10	OS3.2	
disks	?	1 SMD	
cpus	1	1	
1 User	Real	376.10	380.43
	CPU	31.97	4.30
4 Users	Real	417.37	381.17
	CPU	129.17	17.60
8 Users	Real	464.67	392.53
	CPU	263.10	35.73
16 Users	Real	794.40	385.67
	CPU	546.77	74.53
24 Users	Real	crashed	418.80
	CPU	crashed	117.30
32 Users	Real	crashed	540.40
	CPU	crashed	171.30
40 Users	Real	crashed	717.87
	CPU	crashed	219.13
48 Users	Real	crashed	1053.80
	CPU	crashed	275.57
56 Users	Real	crashed no swap	
	CPU	crashed no swap	
64 Users	Real	crashed no swap	
	CPU	crashed no swap	

Preliminary results showed that the Sun-4 at 48 users did not perform at the level of a DG-2000 at 12 users. A Sun-4 at 32 users did not perform at a DG-2000 at 8 users. Not very satisfactory. But the drawbacks listed above would suggest that it was not a realistic comparison.

Benchmarks were then run on the machines in multi-user mode with full telnet, NFS and getty daemons, vendor supplied file systems and kernel, plus 1 large print job to run the duration. The print job was perceived by management as necessary, though the effect on a system is almost negligible given the very light disk load it would have and that I/O processors would do most of the work. This of course assumes the printer was placed on an I/O processor port. Often printers and modems are placed on specially supplied ports that are serviced by the main CPU resulting in degradation of performance.

This is a more interesting comparison as it includes a Sun with extra memory disk and controller to determine bottlenecks.

23 terminal gettys with four ethernet gettys for DG (DG-2000 has 24 ports)
 48 terminal gettys with four ethernet gettys for Sun

machine		DG-2000	S4/260	S4/260	S4/260	S4/260	S4/260
		FPA	FPA	FPA	FPA	FPA	FPA
opt		4	8	8	16	16	16
mem		UX3.10	OS3.2	OS3.2	OS3.2	OS3.2	OS3.2
unix		?	1 SMD	2 SMD	1 SMD	2 SMD	2 SMD
disks		1	1	1	1	1	2
controller							
1 User	Real CPU	378.10 33.10	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A
4 Users	Real CPU	400.20 135.50	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A
8 Users	Real CPU	498.30 274.30	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A
10 Users	Real CPU	crashed crashed	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A
16 Users	Real CPU	crashed crashed	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A
24 Users	Real CPU	crashed crashed	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A
32 Users	Real CPU	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A
40 Users	Real CPU	N/A N/A	720.00 216.90	790.40 215.10	485.00 205.10	477.10 201.60	423.80 206.50
48 Users	Real CPU	N/A N/A	1100.80 273.80	1123.60 266.40	533.40 251.90	555.40 244.80	454.20 266.10

Time constraints limited the benchmark as the magic number needed was the Sun-4 at 48 users was performing at a DG-2000 at 12 users.

Conclusions of these results were:

- 1 Splitting the disk traffic over two disks for the Sun made no difference, in fact figures show response slightly worse, in both cases of 8 megabyte and 16 megabyte systems. Suggests a problem with the controller.
- 2 Doubling memory, doubled performance. Shows the benchmark is disk intensive. Suggest bottleneck with disk subsystem.
- 3 Adding another controller, made a reasonable improvement. Suggests the controller is limited in throughput.
- 4 DG cannot handle even ten users in multi-user mode, which was reflected in problems in real life.
- 5 The Sun-4 is a very suitable replacement. Care should be taken with memory, and a closer look at the disk subsystem should be given.

Further investigations showed that the Xylogics 451 was Multibus based and could handle the sustained throughput of one disk, but definitely was swamped with two disks. There were doubts on whether the controller did

overlapping seeks and reads.

These issues of course did not matter as the machine was still superior with these deficiencies compared to the four DG-2000s.

4. Case study 2

Next is a comparison between Sun-4 and a Mips-120 system. Requirement here is a Sun-4 at 48 users be better than a Mips-120 system at 24 users as the offer was two Mips-120 for one Sun-4/260.

Machines ran in multi-user mode with full telnet, NFS and getty daemons, vendor supplied file systems and kernel, only the Sun-4 had large print job to run the duration. It was not applied to the Mips as the benchmarks were done by different people, but the differences should be minimal.

There are a few problems with this benchmark that should be highlighted:

- 1 The required work mix for the machine should have included Pascal and some rather large and highly CPU intensive Fortran floating point statistics packages. The reason for this is the perceived workload had changed from all Pascal to pascal teaching and Fortran number crunching.
- 2 Work load on the Sun had an extra print job, the Mips-120 did not.
- 3 The benchmarks were by different people. This could be the cause of a lot of inaccuracies and should generally be avoided. Conditions could not be held constant if not explicitly mentioned and different people did the work.
- 4 For 8 megabytes of memory the range of overlap is very small. To compare two Mips-120 with a Sun-4, we can only compare the 48 user Sun test with the 24 user Mips-120 test. Really we need to re-do the tests. So that is not a total waste I have added the Sun single user tests. The 40 and 48 user tests show that there is a only a small speed difference between single-user and multi-user tests.
- 5 For 16 megabytes of memory, the overlap is again very small. Need to re-do Sun benchmarks.
- 6 In the Sun tests, cron was given nothing to do. Not sure about the Mips test.
- 7 The 16 megabyte Mips test was done by Mips themselves. This in itself is full of problems. Though later tests showed the results to be accurate.

24 terminal gettys for Mips-120

48 terminal gettys + 4 ethernet gettys for Sun

machine		Mips120	Mips120	S4/260	S4/260	S4/260	S4/260	S4/260	S4/260
		FPA	FPA	FPA	FPA	FPA	FPA	FPA	FPA
opt		8	16?	8	8	16	16	16	8
mem		?	?	OS3.2	OS3.2	OS3.2	OS3.2	OS3.2	OS3.2
unix									
disks		SCSI	SCSI	1 SMD	2 SMD	1 SMD	2 SMD	2 SMD	1 SMD
controller		1	1	1	1	1	1	2	1
Multi-user		yes	yes	yes	yes	yes	yes	yes	no
1 User		Real	375.50	375.07	N/A	N/A	N/A	N/A	376.10
		CPU	4.93	4.43	N/A	N/A	N/A	N/A	4.30
4 Users		Real	375.53	384.43	N/A	N/A	N/A	N/A	381.17
		CPU	19.97	20.10	N/A	N/A	N/A	N/A	17.60
8 Users		Real	379.93	392.27	N/A	N/A	N/A	N/A	392.53
		CPU	43.67	38.20	N/A	N/A	N/A	N/A	35.73
16 Users		Real	460.43	379.10	N/A	N/A	N/A	N/A	385.67
		CPU	100.13	80.33	N/A	N/A	N/A	N/A	74.53

24	Users	Real	1215.00	382.50	N/A	N/A	N/A	N/A	418.80
	CPU		174.20	122.90	N/A	N/A	N/A	N/A	117.30
32	Users	Real	kernel	384.40	N/A	N/A	N/A	N/A	540.40
	CPU		kernel	167.50	N/A	N/A	N/A	N/A	171.30
40	Users	Real	kernel	406.93	720.00	790.40	485.00	477.10	423.80
	CPU		kernel	217.57	216.90	215.10	205.10	201.60	206.50
48	Users	Real	kernel	kernel	1100.80	1123.60	533.40	555.40	454.20
	CPU		kernel	273.80	266.40	251.90	244.80	266.10	275.57

Conclusions of these results were:

- 1 Need to re-do the Sun benchmarks at lower number of simulated users. Results indicate that after 16 users Mips is no longer better than Sun over 32 users for 8 megabytes of memory, but for 16 megabytes seems to out perform the Sun at 40 users each. It would be interesting to see if Sun still has this problem at 16 megabytes with less simulated users.
- 2 The Mips uses a SCSI drive while the Sun used SMD, need specifications on both before a real comparison should be made.
- 3 Need to do Mips test with extra disks or even faster disk. Shows that with only 8 megabytes of memory the Mips quickly runs out of steam, probably because of the disk sub-system.
- 4 Sun-4 seems to handle larger number of users better than two Mips-120 with only 8 megabytes of memory. However one Mips-120 seems to out perform the Sun when both have 16 megabytes.
- 5 Conditions tested for does not represent needs, the heavily statistical component will favor the machine with the higher Floating Point performance, but it is not known how often this is to be done. This should have been part of the workload mix.

5. Discussion on Mips-120

There is a dramatic improvement between Mips-120 with 8 megabytes and with 16 megabytes of main memory. In fact the performance of the Mips-120 with 16 megabytes was so good it warrants further investigation.

The bulk of the improvement seems to be with disk throughput. The Mips-120 uses a synchronous SCSI protocol which has a maximum of 4 Mbytes/sec. Scsi controllers are only just now capable of these speeds and it seems unlikely the the Mips-120 had that technology in it at the time of the benchmark. The drive they used is a Wren-4 which has a peak synchronous transfer rate of 4.7 Mbytes/sec but goes as low as 1.2 Mbytes/sec in sustained transfer. This is with an optimally tuned filesystem, inadequacies can bring this down further. In fact the Wren-4 drives has an internal structure which defeats some of the optimization attempts of the Berkeley Fast File System, which Mips uses, which further compounds the problem.

The bulk of the throughput improvement is through the use of disk buffers. 1.6 megabytes of disk buffers for a 16 megabyte system. The Sun doubled its performance after doubling its memory and probably doubling its disk buffers. Sun's problem is that the large binaries increases swapping, so a large number of buffers can make the problem worse. Mip's dramatic improvement suggests that maybe the 8 megabyte system might have had less than half the buffers, but as it is System 5 it does has shared libraries which gives it another edge. Sun has since brought out shared libraries with SunOS 4.0 which should help it.

6. Conclusions

There is enough information here to suggest the benchmark to be re-run with more specific configurations. For accuracy, they should be done by one person at the one time. Both Mips and Sun seem to take off with the extra memory, and tuning with better drives and controllers seem to be what is needed.

Along with recording hardware configurations, the software configuration such as kernel parameters should also be recorded. A machine can be improved dramatically if the kernel is tuned accordingly.

A more general conclusion will take the form of an observation. Case study 1 showed that the DG benchmarked better than the Sun. The study also showed the Sun benchmarked better than the DG given different circumstances. Case study 2 showed Suns and Mips with a range of performances depending of the configuration. This resulted in the Sun benchmarking better than the Mips-120 in one case and the Mips-120 better than the Sun in the other.

The moral of the story is that you can prove whatever you want with benchmarks.

To make a decision based on this data could result in a poor choice, new releases in the operating system and compilers could change these figures. Faster drives will dramatically change them.

Acknowledgments to the people at La Trobe university in the Computer Center and Computer Science Department for their contributions, and to Sun and Mips for the information they have given.

Sun's Response

Date: Wed, 4 Jan 89 16:31:59 EST
From: richb@sunchat.sun.oz (Rich Burridge)
To: steve@labtam.oz
Subject: Response to MUSBUS paper.

I've received this from Keith Bierman, one of the Tactical Engineers that works in the benchmarking group in Mountain View. Please consider it as our response.

Rich.

-----Forwarded message-----

Date: Tue, 3 Jan 89 18:53:45 PST
From: khb (Keith Bierman - Sun Tactical Engineering)
Message-ID: <8901040253.AA02589@chiba.>
To: sunaus!sunchat!richb
Subject: your benchmark paper
Cc: jeffm

Rich,

Jeff asked me to look over the MUSBUS study, and to comment. Here goes:

- 1) Workload simulations, like MUSBUS, have many implementation niceties which are non-obvious. In a former incarnation I was involved in creating some benchmarks along these lines, and it is indeed very benchmark dependent.
- 2) Machines like the Sun4 are designed with the notion that one has (essentially) one user, and a handful of tasks; and that the user is greedy (i.e. wants huge performance). Machines like a vax780 (i.e. minicomputers) tend to be designed with the idea of having many users with light workloads.

This is reflected in the balance between IO and CPU power. A big Burroughs mainframe from some years back was basically a 1 MIPS machine... but it could handily support 100 users. This is because it had many "channel controllers" (to use the IBM term) which offloaded much of the processing overhead from the CPU.

It is really amazing how well the Sun4 holds up under multi-user workloads. We should not be surprised that a Pyramid or a DG (or a big VAX) can win on "properly configured" (i.e. if I worked at DEC, this would be proper) workload tests.

In general, if one must have one big server, and lots of clients who don't do much work; we should EXPECT the minicomputer to win.

The general concept behind Sun, Apollo, and other workstation vendors, is that one is better off with multiple smaller machines than one big one. Especially favored is at least one CPU per person. To keep human productivity up, it is best to have CONSISTENT response times, so we try to put as much CPU power at the user's fingertips.

Sample workstation oriented solution would be to get many sun3/50's, 60's, diskless 386i's, or 4/110's and a sun3/260 for file handling. The only drawback to the 50's and 60's is their small memories, but students should be running w/o high levels of optimization (student codes are usually small, and usually spend much more time compiling than executing). For f77 type jobs, a dedicated 4/110 or a sun3/fpa equipped machine would be a good choice. Each workstation would handle a handful of students.

A big advantage of this sort of arrangement is a relatively linear growth path (double the students, one can double the number of workstations...also works for 10% increase).

Minicomputer type solution requires a big upgrade (i.e. may pass 2x test, but can't upgrade 10%, then another 10%, etc. for matching purchases to funding this is a big improvement).

If cleverly implemented (i.e. add scsi disks on various machines, don't rely on big server for much) a networked solution is more robust than a uniprocessor solution.

- 3) Kernal configuration. System tuning to the expected workload should be done for this sort of benchmark. Sun's, for example, come with lots of "junk" in the kernal (i.e. support for every device one MIGHT want to add someday). System tuning is a bit of a black art, but it should be done anyway. Otherwise one is not only benchmarking the machines, and the software, but the assumptions of the release control folk at each vendor. Some feel that everything should be there, and users should remove what they don't want (sun) others put in little and force users to figure out how to add it (vax microvms). It is far from clear which is more reasonable, but neither is likely to match the requirements.
- 4) The verbage about quality technical support is on target. Users WILL have problems. How hard it is to get them fixed is very much an issue.

Aside from the vendor, the quality of the user groups should be considered. Sun has one of the most vocal, most sophisticated, and most helpful user communities in the world.

In addition one should consider the quality and number 3rd party supporters, as a strong aftermarket is a sign of a strong vibrant company.

- 5) It is not reasonable to insist that all vendors provide machines for all benchmarking activities...this is simply not economically viable in MOST cases.

Someone who is just learning how to use machine x will not, in general, know how to extract good performance. Everything from compiler switches to system configuration is involved. A newcomer to a machine, no matter how savvy, will be at a considerable disadvantage. The result is that machines which resemble the

preconceptions of the benchmarker will have a competitive edge.

Thus most benchmarking is a cooperative effort between customer and vendor. After the vendor has had a chance to tune the machine (and, explain what is involved), possibly the code (for large applications, which will be run in a production mode (i.e. most big scientific codes)), and possibly the benchmark (i.e. help define what is sensible to be measuring/running) the customer may wish to supervise the final timed runs.

Of course this involves a great deal of effort on the part of vendor and customer. But this allows technical support and competence to be judged.

- 6) When running a mix of production and development runs, one probably should lower the priority of the production runs. Development is very interactive, and users are easily frustrated by delays. Production runs which run for "reasonable" lengths of time, say, 20min can be delayed by as much as 10min w/o seriously impacting performance.

Of course, this illustrates an advantage of networking. On a network the production runs need not impact the development performance.

- 7) Workstation type machines will be very sensitive to RAM deficits, vis a' vis minicomputer class machines. Thus we should not be surprised that the sun (and mips) machines do better with more memory.

Even on minicomputers this can be an important issue. At the '87 Sigmetrics (or the last one in Banff, Canada) there was an interesting paper on putting a .5Gb (or some such) on a Vax780. The increase in deliverable performance was much larger than that obtained by a similar (cost, back before the memory shortage) CPU upgrade.

More memory beats more CPU in many cases (especially multi-user).

- 8) Because these tests are heavily stressing the IO handling there is a lively discussion of the various disk/controller issues. Sun now offers some more powerful controllers, and there is a significant aftermarket. The sun performance could be improved with faster IO components, or via more memory, or via more (cheaper) machines. All of these are sun supported solutions.
- 9) There is mention of large sun4 binaries. It is possible to reduce the size of many binaries via the shared library facility. This can be good for a 10x reduction.
- 10) MIPS makes a fine product, and I am sure a MIPS or Sun based solution can easily handle the departments needs. I am sure that both are better than a DG or DEC solution.

Sun offers a variety of nifty pluses, graphics environments, etc. MIPS has concentrated on CPU performance.

If the sole mission of the department is to support entry level pascal, the "best" solution would probably be 100 Z80 based machines running CP/M (or TurboDOS) running TurboPascal. And a fast PC or workstation for the f77 users.

From the fact that this solution was not considered as part of the benchmark, it can be inferred that the administration advantages of a more centralized system is considered vital; and that there is need to consider other needs (ability to run big problems, advantages of teaching in unix, etc.) Sun offers a wide range of products from modest workstations (sun3/50), to powerful workstations (4/110, 386i) to very powerful (4/260), and a wide range of graphics and imaging products (including the "ultra powerful" TAAC-1).

MIPS offers workstation class servers (i.e. no intergral graphics) which are very good.

Sun offers a wider variety of tools to solve the departments current and future needs, has a larger user base, larger 3rd party support population, and a much larger base of technical know how...

- 11) "The moral of the story is that you can prove whatever you want with benchmarks. "

No. The moral of the story is that different machines are designed, and/or deleivered with different assumptions about what the computer will be used for. If one carefully studies what one really wants to do, one can construct a benchmark which can be used to rank systems.

As admitted in the body of the report, what was being studied changed during the time the research was conducted. If we change the hyptothesis, how can we expect the conclusions to remain constant ?

- 12) "To make a decision based on this data could result in a poor choice, new releases in the operating system and compilers could change these figures. Faster drives will dramatically change them."

True. Also using ONLY benchmark data to decide how to set up the department in question would be suboptimal. Simple pascal programs (typical of students) can be developed on a wide variety of machines, and nearly any vendor can provide a solution. But a vendor like Sun can provide incremental growth.

I hope this is sufficiently responsive.

**EUROPEAN
UNIX® SYSTEMS USER GROUP
NEWSLETTER**

EUUG

*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

Changing the *roff Escape Character

*Markku Sakkinnen
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', '(WVV' (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 'Œ'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 \\\\\*(//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' ('\' 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.

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!

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.

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:

† 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.

```
# 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

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.

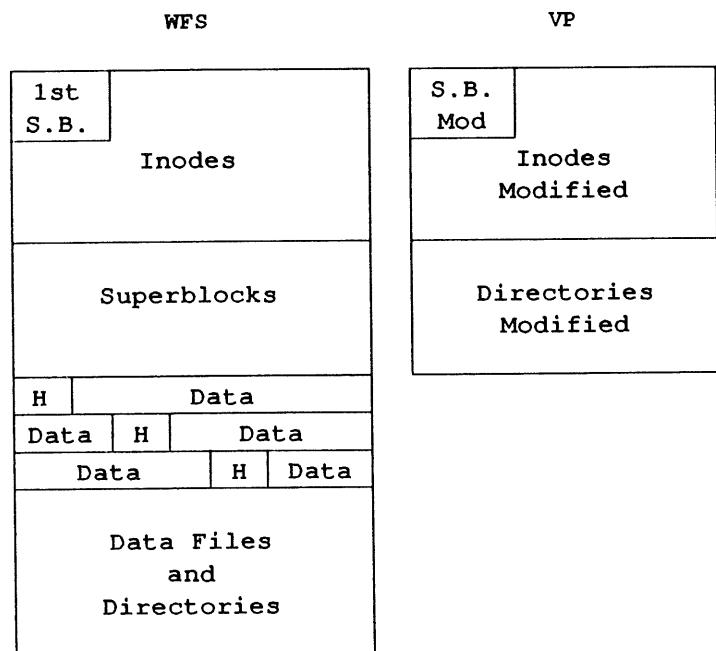


Figure 1

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 than 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 mounted 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 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 rose'. 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?-ls' | 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 worthwhile. 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 "europolian" 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

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 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 Előd Knuth Computer and Automation Institute Hungarian Academy of Sciences H-1502 Budapest 112, P.O. Box 63 Hungary Telex: Phone: Fax:
Secretariat:	Mrs Mária 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:	<i>mpalian@idc.idcyuug.uucp</i>

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*



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, the 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: Eddy 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, e.g. 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
uclfyu	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 many 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 multilingual 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

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

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.

The POSIX Standard and Its Future Development

Cornelia Boldyreff
corn@cs.brunel.ac.uk

UK POSIX Panel Convenor

Recent Developments

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 Administration
 - 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

Future Developments

Operating System Environment Based on UNIX system Model

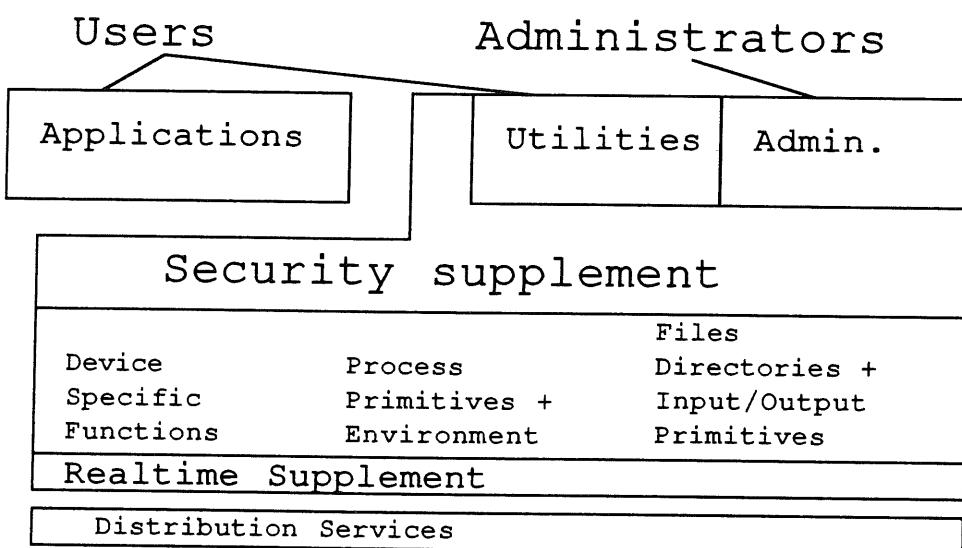


Diagram developed by ISO WG15 to explain their proposed future work.
Tokyo, 20-21 October 1988.

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 SVI 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

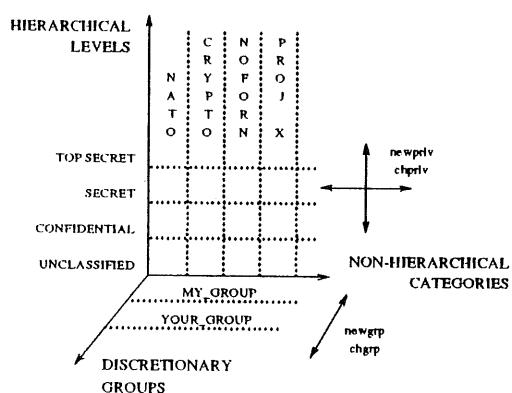
System V/MLS generates an audit record for all security relevant events and all data accesses. There are twenty selectable trace channels (see table), and the records may be stored on disk, tape, printer, or a remote system. A formatter is provided for use in analysing audit trail data in two levels 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

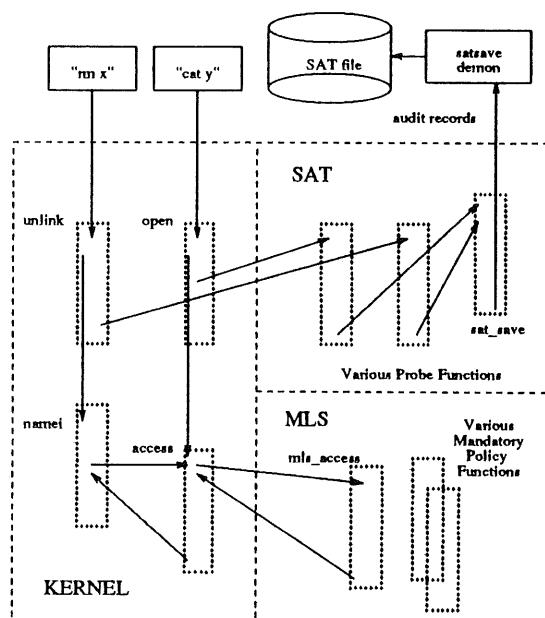
Users initially login with a privilege and are given a shell that operates with that privilege. A new command, *newpriv*, allows them to change to a new privilege in the same way *newgrp* allows them to change to a new discretionary group. Each *newpriv* to a higher level generates a new shell, which 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

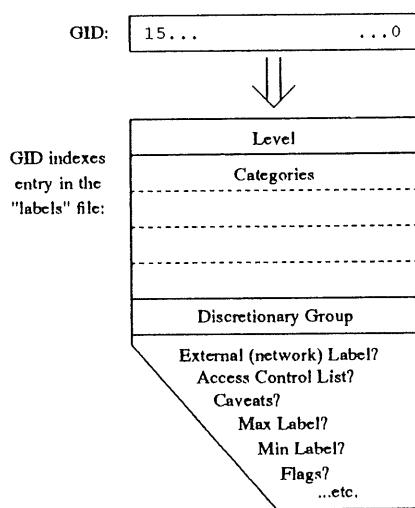
System V/MLS is implemented by two modules, which are called from about fifty "probe points" and "hooks" that have been placed in the UNIX kernel (see diagram).



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

Indirect GID-based Labels

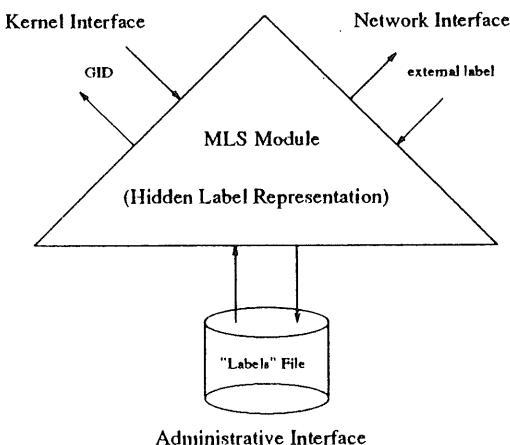
Data labeling is accomplished by using the group identification (GID) as a pointer to the labeling information, which is maintained in the /mls/labels file (see diagram).



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

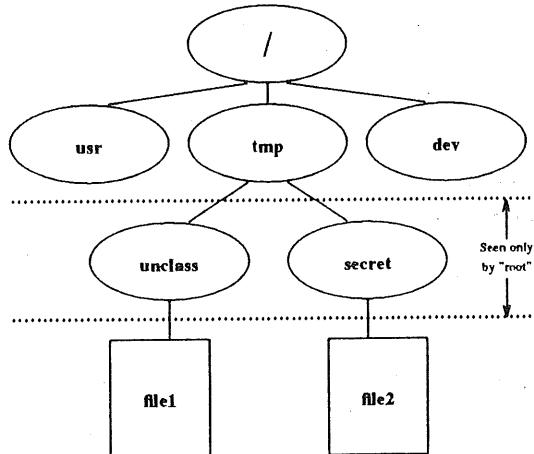
The MLS module "hides" the internal label representation and translates for each external interface (see diagram).



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.

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).

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	PRIME	3B2/600	SUN
NCR	+	+	?	+	-
UNISYS	+	+	?	+	-
PRIME	?	?	+	+	?
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**.

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...

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.gpsi.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-

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

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 caching is explored.

**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 Wheaton

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!ltp!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 Ahamed

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 Ingenierene 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
lby+@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@ln.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. Dudat

A. Freyssinet‡

E. Paire‡

M. Riveill†

X. Rousset de Pinat

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

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

† 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!ltp/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
djk@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.

IMAGES – an approach to an Object Oriented UIMS

Luis Simões

J. Alves Marques

Nuno Guimaraes

Luis Carrizo

Manuel Sequeira

INESC

Lisboa

PORUGAL

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

[†] 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.

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

Bubette 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

† 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.

**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 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

*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.

**EUROPEAN
UNIX® SYSTEMS USER GROUP
NEWSLETTER**

EUUG

*Volume 9, Number 1
Spring 1989*

Editorial	1
A Port of the MINIX Operating System to the Atari ST	2
Book Review	14
AUUG Conference Report	15
Journées UNIX	23
EUUG Executive Report	27
Dutch User Group Report	29
AFUU Diary	30
UKUUG News	32
Calendar of UNIX Events	34
Introduction to Window Systems	37
Puzzle Corner	43
Software Review	44
USENIX Association News for EUUG Members	61
UNIX Clinic	63
ENet	69
UNIX is Chauvinistic	72
Overview of UNIX System V Release 4.0	74
Book Review	79
UKUUG Winter Abstracts	81

A Port of the MINIX Operating System to the Atari ST

Aarron Gull & Sunil K Das

aarron@cs.city.ac.uk

sunil@cs.city.ac.uk

*City University London
Computer Science Department
London EC1V 0HB, United Kingdom*

Aarron Gull is a ("scoff if you like, but it's true; the force surrounds us, holds us, binds everything together") trainee UNIX guru. Currently working on his PhD ("I feel ... a grave disturbance in the force") at City University London. He is interested in distributed processing, films and lemonade.

Sunil K. Das is a founder member of the UNIX Travel Club (at the expense of others).

Abstract

The STIX project consisted of two activities. The first was the creation of a MC68000 cross-development environment on a Gould PN6000 host computer. This would be used for the down-loading of C binaries to Atari ST target machines. The second was the port of the MINIX operating system, designed to run on the IBM PC, to the Atari ST range of micro-computers.

This paper describes elements of the STIX project which were particular interesting or unusual. Since the Atari ST does not have a hardware MMU, emphasis is placed on the process and memory management techniques employed in STIX. The impact of these ideas upon the system call interface, in particular *fork()*, *exec()* and *exit()*, concludes the discussion.

The STIX Project

The MINIX (mini UNIX) operating system runs on the IBM PC range of micro-computers [Tanenbaum 1987a]. Externally, MINIX resembles seventh edition UNIX (V7) by having a similar system call interface and being distributed with many UNIX-like commands. Internally, however, it is very different; MINIX is a modern operating system whose modules communicate via *message rendezvous*. System call requests from user processes are effected by a message pass to the appropriate server module.

For programmers of IBM PC micro-computers MINIX is a low-cost, UNIX-like system which is a

viable alternative to MS-DOS. Teachers and students will find MINIX of particular interest because it is fully documented [Tanenbaum 1987b] and the source code is distributed without licencing restrictions.

The STIX project, a MINIX port to the Atari ST, was conceived so as to allow the operating system to be studied on the 100-or-so Atari ST micro-computers at the City University. The successful outcome of the project would form the basis of future undergraduate or postgraduate courses and projects in computer science. For example:

- operating system design, directly or indirectly based upon MINIX;
- development of commands and tools which run under MINIX; and
- use of the Gould/Atari ST cross-development system.

The remainder of this paper is devoted to an exposition of the Gould/Atari cross-development environment which was constructed and the MINIX port itself. Emphasis is placed on a description of the novel techniques employed to overcome the lack of hardware memory management.

The Host-Target Environment

A host-target environment typically offers several advantages over the traditional method of developing software on the target machine; the host tends to be faster, more stable, and has a greater disk capacity. Moreover, using a host which runs UNIX provides access to a range of tools (e.g., *lint*, *make*, *SCCS* and *RCS*) which are specifically intended for C program development.

It was decided that the creation of an Atari cross development system, running on a Gould PN6000, would greatly increase the speed of the port. This decision had the additional advantage in that, on completion, the system would be available to Atari programmers who require a more hospitable development environment. For this purpose, it was decided to also create a library of function calls to interface to the resident Atari operating system TOS [Gerits 1988].

Version 2.0 of the Stanford UNIX Mac C Development Kit (SUMacC) [Croft 1984]¹ was chosen as the basis for the cross-development system. Packaged for a VAX computer running either BSD UNIX or Eunice, the SUMacC system includes compilers from the Universities of Berkeley and Stanford. These are merged with conditional compilation statements. The Berkeley code is based on Steve Johnson's portable C compiler [Johnson 1978] and was used to form the basis of the host-target system; a detailed description of which is available from the authors

1. SUMacC is distributed in the public domain under the condition that it may be used but not sold because it is subject to a Stanford copyright.

[Gull 1988].

The complete compiler suite comprise:

- System-wide header files;
- C compiler (*cc68*, *cpp68*, *ccom68*, *c268*, *as68* and *ld68*); and
- C library, *nm68* and *lorder68*.

The majority of the STIX-dependent modifications made to the compiler suite (as opposed to porting and bug fixes) were located in the linkage loader *ld68*. Details of these changes are covered by the discussion of the *exec()* system call.

The source code of the C library was not originally available to the authors. As a consequence, the decision was taken to reimplement it. This proved to be a long and rewarding exercise (13,000 lines of C and 2,000 of assembler) but does not merit a full description.

The commands *nm68* and *lorder68* were ported to the Gould. These were used to rebuild the library whenever a module was changed.

From MINIX to STIX

For the purpose of the ensuing discussion, the term STIX will describe the port of MINIX operating system, thus enabling a clear distinction to be drawn between the IBM PC and Atari ST versions of the code.

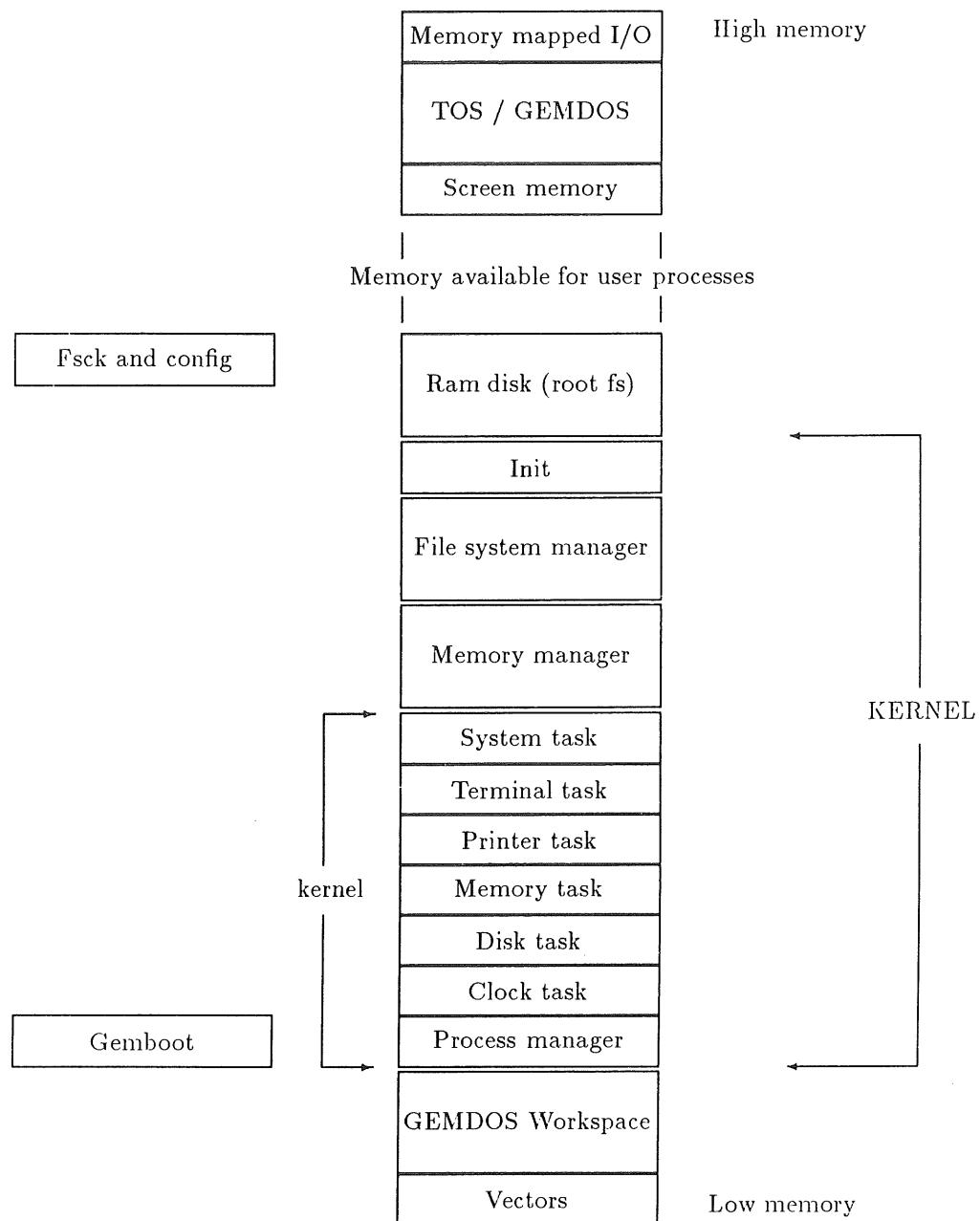
Booting STIX

The File System Check program (*fsck*) is bootstrapped on power-up by the TOS program *Gemboot*. This loads the *fsck* text and data from disk, clears the BSS space and starts running the code.

Fsck loads the modules of the STIX KERNEL into low memory according to the information held in the system configuration file *config.dat*. The BSS and headers are constructed for each KERNEL module. Execution then begins at the main KERNEL entry point.

STIX Memory Layout

The composition of STIX is very similar to that of MINIX in that it consists of four modules which are collectively called the *KERNEL* (n.b. upper case). It differs, however, in that each KERNEL module is contained in a separate *.dat* file on the boot disk: *kernel.dat* (kernel); *mm.dat* (memory manager); *fs.dat* (file system manager) and *init.dat* (init process). The *kernel* (n.b. lower case) is

**Figure 1: The layout of the STIX memory**

further subdivided into a process manager, a system task, and device drivers for the terminal, printer, disks and clock.

Because of the nature of the ST hardware the modules of the STIX KERNEL are linked relative to each other while the modules within MINIX can

be linked independently.

The modules of the STIX KERNEL can make reference to one another though use of headers: arrays of eight longs which precede their text in memory. These contain lengths from which the positions of the other modules may be calculated:

```
#define UNUSED_LENGTH    _begtext [-8]
#define KERNEL_LENGTH     _begtext [-7]
#define MM_LENGTH          _begtext [-6]
#define FS_LENGTH          _begtext [-5]
#define INIT_LENGTH        _begtext [-4]

/* Used for booting off hard disk */
#define BOOT_DEV           _begtext [-3]

/* _begtext[-2] to _begtext[-1] are currently unused */
```

The overall memory layout of STIX is illustrated in figure 1.

An Overview of STIX Processes

A process is the image of a program in execution and includes amongst its associated resources:

physical data such as the text, data, BSS, stack, the invocation arguments and the environment; and the concept of state as defined by the processor registers, the open file descriptors, the current working directory and scheduling information. The memory map of a typical process is shown in figure 2.

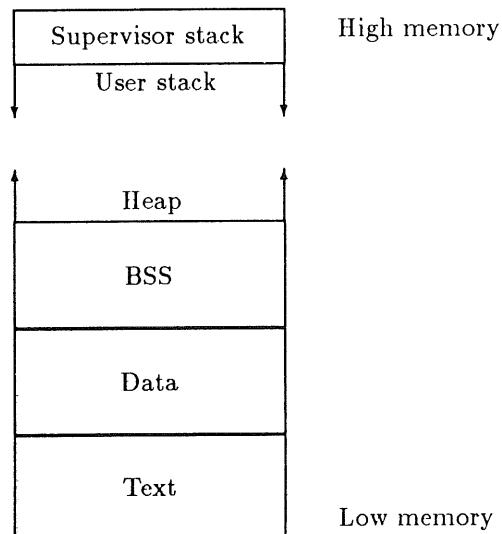


Figure 2: A typical STIX process

STIX processes are very similar to those of V7; each user process is created when another executes a *fork()*.

STIX processes, like those of MINIX, are fixed in size at creation; the *brk()* and *sbrk()* system calls do not change the overall amount of memory consumed by the process: the memory requirement is simply validated.

The code images created by the *fork()* and *exec()* system calls could, in theory, be loaded into any area of free memory². The Atari ST does not have memory management hardware to assist with this. Therefore, three alternative strategies for process creation in the STIX environment were considered:

- Find a compiler which produces totally relocatable code. Although C compilers exist which produce position independent code (PIC) the authors do not know of one which creates code which is relocatable while in execution.

In theory, it would be quite simple to write a compiler which generates such code. The code produced, however, would be very inefficient and system performance would be significantly affected.

- Discard the *fork()* and *exec()* system calls and adopt a *forkexec()* call. This new call would create another process running a specified filename. In order to support the new call it would be necessary to place relocation information in executable files.

This approach would have the advantage of being simple to implement, but the considerable disadvantage of losing V7 compatibility. Since compatibility with V7 was thought to be of primary importance this method was rejected outright.

- Use swapping³ of the data, BSS and stack images of related processes to implement

2. All MINIX processes actually start and end on 16-byte boundaries (*clicks*). This is due to address resolution on the 8088 being indexed through 16-bit segment registers which function as 20-bit registers. STIX processes start on even byte boundaries and are multiples of 4 bytes in length. This enables the *swapper()* to move DATA as quickly as the ST architecture allows.

3. The STIX *swapper* function should not be confused with the UNIX swapper process.

forking. Relocation information would be needed in executable files for *exec()*. Substantial revisions would have to be made to the memory manager code to control such process migration.

This approach has the advantage of retaining the V7 interface, but the disadvantage of adding substantial complexity to the kernel and memory manager.

It was decided that the final method was the most satisfactory.

Fork()

In order to retain the *fork()* system call, STIX supports process swapping. The concept is simple: STIX processes comprise contiguous regions of TEXT (which is static) and DATA (which are dynamic), both of which are of fixed size. When a process forks, the child requires a private copy of the parent's DATA but can share its TEXT. When executing, both parent and child need to occupy the parent's original memory space.

When a process is scheduled its DATA will be moved into the memory it requires to run, swapping it with any DATA already residing there. Related processes will execute by swapping DATA in and out of the memory originally occupied by the parent.

The *fork()* algorithm knows little about swapping: this is handled by the context switching code. The *fork()* code simply creates another entry in the process table containing the new current address. The image of the child is created by the swapper; the child is flagged as *copy_on_swap*. This produces a marked reduction in the overhead of a typical *fork()* *exec()* sequence but only works because the child is scheduled to run before its parent.

The modular nature of the STIX KERNEL means that it is necessary to introduce the concept of the *creation*, *current* and *execution* address of a process. The *execution address* is the memory a process' DATA must be located in while running while the *current address* is its present location. When a process is running its execution and current addresses will be the same. The DATA of processes which are not running, however, may be relocated so that these addresses differ.

Due to the fork mechanism, if two processes share the same execution address, they must be related and, therefore, are the same size. The *creation*

address space is the memory allocated to a process on execution of either the *fork()* or *exec()* system calls. It is also the memory released by *exit()*. On termination process DATA is returned to its creation address space so that its memory can be released.

```
union mem_map {
    struct proc_m_map {
        unsigned mem_exe; /* Process manager only */
        unsigned mem_curr; /* Data execution address */
        /* Current DATA address */

        unsigned text_len; /* Length of text only */
        unsigned data_len; /* Length of data, BSS and stack */

        char copy_on_swap; /* As of yet without DATA */
        unsigned child_creat; /* Address of child DATA storage */
    };
    struct mem_m_map {
        /* Memory manager only */
        unsigned mem_exe; /* Process execution address */
        unsigned mem_creat; /* Process creation address */
        unsigned whole_len; /* Length of process */
    };
};
```

Figure 3: The mem union.

In STIX, the *swapper()* is called whenever a process needs to be restarted. It is the program which ensures that the data, BSS and stack of the process are located in the required region of memory. If necessary, the *swapper()* interchanges the DATA spaces of two processes which are co-resident in user memory. The algorithm of the *swapper()* is shown in figure 4.

Some care must be taken when copying data to and from processes. This will occur as part of message passing or due to kernel system call activities. Processes pass system call addresses

The STIX process manager's tables contain the *execution* and *current address* of each process while the memory manager stores the *execution* and *creation addresses*. The union used to store process memory maps is illustrated in figure 3.

```
algorithm swapper
input: proc table pointer
output: none
{
    if (!currently panicing &&
        process DATA not in execution region) {
        if (KERNEL process)
            panic(attempt to swap in kernel);

        found = false;
        if (!copy_on_swap)
            for (all user processes)
                if (process DATA is in execution region) {
```

which are valid when the process is swapped into its execution region. Such addresses are invalid when the process has been swapped out. Several functions have been modified in order that this is taken into account.

There is a significant flaw in the way processes are implemented on the Atari ST; the hardware provides no mechanism for protecting processes from each other; there is no way of stopping a program, accidentally or maliciously, reading or writing outside its allotted memory. This is attributable to the hardware and not to the implementation of the STIX KERNEL.

```

        found = true;
        break;
    }

    if (found == true) {
        if (DATA are of different lengths)
            panic(cannot swap different length processes);
        swap DATA with that in execution region;
    } else
        if (copy_on_swap)
            copy DATA into child region;
        else
            copy DATA into execution region;

    update the current address fields of the processes;
}
}
}

```

Figure 4: The swapper() algorithm.

Finally, although swapping DATA might seem slow, the actual cost in system performance is small. Most programs immediately follow a *fork()* system call with an *exec()* in the child. The system only needs to make one copy and perform one swap; a copy of the parent is made prior to the child running and a swap is made to release the child's memory after the *exec()*.

The overhead of forking can be further reduced by increasing the timeslice allotted to running processes.

Swapping, perhaps surprisingly, reduces the complexity of the *fork()* system call. It does, however, make process termination and subsequent memory reallocation considerably more complex. This is accomplished during the *exit()* system call.

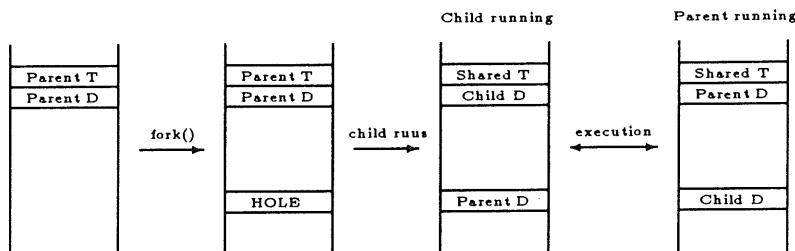
Example 1: This is introduced in order to clarify the usage of the address fields defined in *union*

mem_map().

A process whose execution address and creation address are equal is scheduled to run. The process forks a child. The kernel initiates a search for a process table slot for the child. This is followed by the memory manager searching for a 'hole' in memory large enough to hold a copy of the parent's DATA image. Assuming the *fork()* is successful, a child process is created and scheduled to run before its parent.

When the child begins execution a copy of the parent's DATA is made into child's creation memory space. The DATA of the parent process then becomes that of the child. The parent's TEXT is shared between parent and child.

From then on whenever either process is scheduled, they must be relocated to their execution address space to run. Swapping in this way is only possible because related processes are of the same length.

**Example 1: The fork sequence**

Exit()

The *exit()* system call has the task of clearing up after processes which have terminated. The algorithm of the exit system call is illustrated in figure 5. Points of interest are:

- The memory manager has no understanding of swapping; In order to reduce the number of message passes required only the kernel knows the current address of a process. This makes releasing the memory of a terminated process difficult: the memory manager does not know which region of memory should be released. By calling *sys_swap()* using the terminating process number and creation address as parameters, the process is immediately located.
- A parent process will typically have an

```

algorithm system_call(exit)
input: return code for parent process
output: none
{
    store exit status for next wait( ) system call;

    if (parent is waiting)
        release parent and tell everybody;
    else
        suspend process;

    tell kernel(to swap process into creation region);
    tell kernel(to stop scheduling process);

    if (no other process runs in execution region)
        free(text && execution regions);

    if (execution region != creation region)
        if (no other process runs in creation region)
            free(creation region);

    /* Process is now in zombie state */
}

```

Figure 5: The exit() algorithm.

Exec()

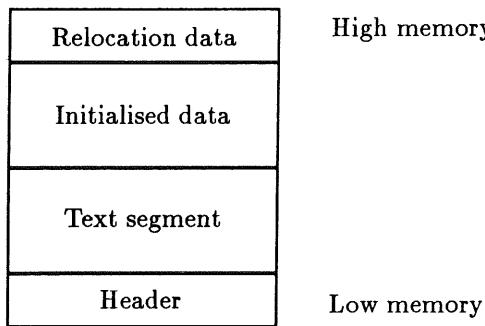
The *exec()* system call has been made more complex due to the lack of hardware memory management on the ST. The main problems which had to be overcome were program

execution address and creation address which are identical. The execution address of a process is shared with any children it created. This memory must not be released until the last child exits, even if the parent terminates. The memory of a process with no children may be released when the process exits.

- MINIX does not release the memory associated with a process until the parent process has executed a *wait()* system call. This could result in a *fork()* or *exec()* system call erroneously failing due to lack of free memory. STIX releases the memory of processes when they *I exit ()*. This leaves entries in the process table which have no memory associated with them. This is similar to the *zombie* process state in UNIX.

relocation and memory reallocation.

The format of a STIX executable file *b.out* is shown in figure 6; there is no space between the header, text, data and relocation segments.

**Figure 6: The b.out file format**

STIX user processes may be any size up to a soft limit of 120K imposed by the superstructure and a hard limit imposed by available memory.

Code executed in user mode differs from that of the STIX KERNEL in the following ways:

- Whilst all executable files in STIX have headers, files intended to run in the KERNEL have a different header format to those intended to run in user mode. The KERNEL headers hold data about the size of the KERNEL modules; user headers contain details which are necessary to create processes.

STIX executables which are intended to run in user mode differ slightly from those of MINIX in that two previously empty header fields are now used to hold the size of the text and data relocation information. The new header

```
struct {
    long fmagic;          /* Executable type */
    long hsize;           /* Header size */
    long tsize;           /* Text size */
    long dsize;           /* Data size */
    long bsize;           /* BSS size */
    long rtsize;          /* Text relocation items */
    long tmem;            /* Total memory required */
    long rdsize;          /* Data relocation items */
} stix_header;
```

Figure 7: The header format

The algorithm of *exec()* is illustrated in figure 8. A process invokes *exec()* by calling one of its front-end functions to build an image of the new process stack from the supplied arguments and environment. The stack image is passed as an argument to the *exec()* system call.

structure is shown in figure 7.

- User code must be able to be loaded anywhere in memory; KERNEL code is loaded at a fixed address in memory. Hence, user code must retain relocation information whilst the KERNEL has no need for it.
- User programs always begin execution at their first instruction; the KERNEL has multiple of entry points.
- User code requires a startup routine to set up the *argc*, *argv* and *envp* parameters; the KERNEL has no need for startup parameters.

To inform the linkage loader *ld68* that code is intended to run as a STIX *user process* the *-M* (MINIX) flag is passed from the compiler front-end.

After a process has executed a successful *exec()* system call and prior to being restarted, its memory map will resemble that in figure 9.

The most important features of *exec()* are:

- new text and data segments have been loaded and relocated;

- the BSS, heap and stacks have been filled with zeros;
- the arguments and environmental information have been copied into the user stack; arrays of pointers to each entry have been set up; argc, argv and envp have been pushed into the user stack; and
- the supervisor stack has been constructed so that the process will restart execution in the C

startup routine.

The C startup routine *crtso* is an assembly language routine which simply calls *main()* as a subroutine. No arguments are passed by *crtso*; these have already have been placed on the stack by the kernel. *Crtso* simply builds the standard parameter passing stack frame around the arguments. The routine *main* can then be treated as a standard C function.

```

algorithm system_call(exec)
input: pathname and stack image
output: only if exec fails
{
    if (invalid stack image || filename is !executable)
        return(error);
    Fetch the new stack from the user;

    Read the file header and extract the header sizes;
    if (!enough free memory for new image)
        return(error);

    /* Last point a failed exec( ) will pass back to user */

    tell_kernel(to swap process into creation region);

    if (no other process runs in execution region)
        free(text && execution regions);

    if (execution region != creation region)
        if (no other process runs in creation region)
            free(creation region);

    Reclaim memory to hold new image;

    Read in the new image from the file;
    if (the header information is invalid)
        exit(process);

    Relocate the image to its new address;
    if (the relocation information is invalid)
        exit(process);

    zero the stacks and BSS of the process;
    Copy the stack image into the stack;
    Relocate the stack image;
    Take care of setuid and setgid bits;
    Reset all caught signals;

    tell_kernel(tidy the stack and reschedule the process);

```

Figure 8: The exec() algorithm.

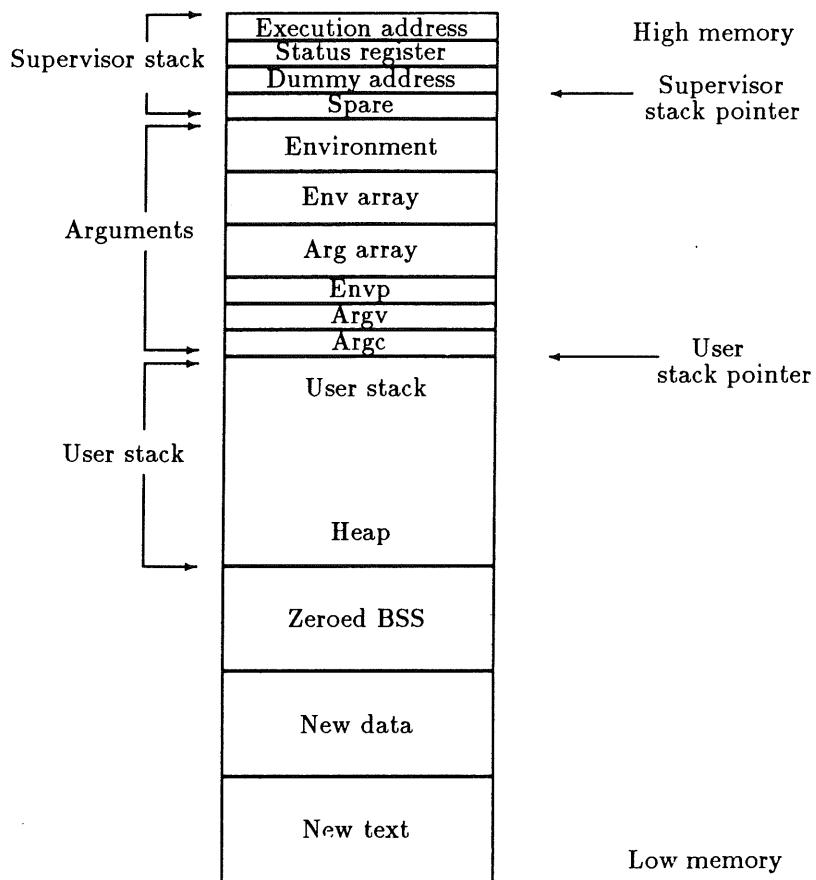
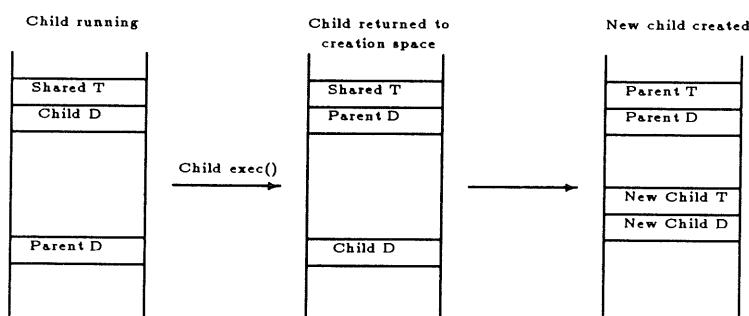


Figure 9: The memory map of a process after `exec()`

Example 2: A process forks a child and the child process executes an `exec()` system call. Once it is certain that the `exec()` can succeed, `sys()` is called to return the child to its creation space. The

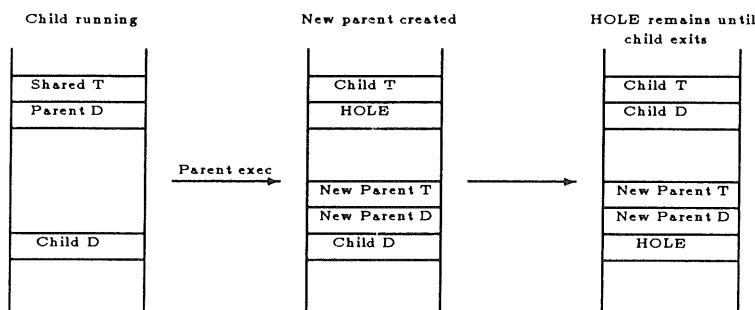
function `free_mem()`, inside the memory manager, is called to release the creation address space. As the parent process still exists, the child's execution space is retained. Finally, memory is allocated to the new child:



Example 2: The child `fork()` sequence

Example 3: This highlights the unusual case when a process forks a child and the parent issues an *exec()* system call. The parent's creation and

execution space is the same, but the space *has* to be retained for child execution. A 'hole' is created in memory which is only removed when the child process terminates:



Example 3: The parental fork() sequence

A complete discussion of the cycle of events during process creation, existence and termination is available from the authors [Das 1988].

Results

In material terms the result of the project is 6 disks: the STIX *boot* disk, a 400K *root* disk and 720K */usr*, */user*, */src1*, */src2* disks. These are all that are required to run STIX as a stand-alone operating system. The KERNEL passes the exhaustive Tanenbaum Validation Suite and is V7

compatible; programs which use V7 calls run successfully in the STIX environment. Moreover, a large number of utilities have been provided.

As a side product of the port an Atari ST development system has been created on the Gould computer. This system can and is used by anyone wishing a more hospitable Atari ST program development environment than that provided by the target machine.

A summary of the changes made to the KERNEL is recorded in the following:

Module	Total lines	Lines changed	changed
File System	3949	491	12
Memory Manager	1445	456	31
Kernel	2450	1669	68
Totals	7844	2616	33

Performance tests indicate that the additional overhead involved in a typical *fork()* *exec()* sequence (in which the child immediately execs) is small. On an 8Mhz Atari 1040 this amounts to around 1/285th second per K of data, BSS and stack in the parent process (1/12th of a second for */bin/sh*).

The C compiler supplied with STIX benchmarks at 762 dhrystones with high register usage and 720 without.

Conclusions

The STIX project has shown that it is possible to run V7 compatible operating systems on machines which have no form of hardware memory management. This particular implementation sacrifices performance when forking in order to

otherwise maintain high throughput.

The authors consider 33% of changed code to be high and attribute it to the vastly different architectures of the IBM PC and Atari ST. Porting MINIX to a computer with hardware memory management would require much less work.

References

[[Croft 1984]] Croft W; *SUMacC: Stanford UNIX Mac C Development Kit*; Stanford University Medical Centre; November 1984.

[[Das 1988]] Das, Sunil K Gull A; *The Process Life Cycle in STIX*; Proceedings of the AUUG Conference: "Networking Linking the UNIX World"; Melbourne, Australia; 13th-15th September 1988.

[[Gerits 1988]] Gerits K, Englisch L Bruckman R; *AtariSTInternals-The authoritative Abacus Software*; 1988.

[[Gull 1988]] Gull A; *STIX: A Port of the MINIX Operating System to the Atari ST*; BSc Final Year Project Report - City University London Computer Science Department; May 1988.

[[Johnson 1978]] Johnson S Ritchie D;

Portability of C Programs and the UNIX System; The Bell System Technical Journal; Vol 57, No 6, Part 2, pp 2021-2048; July-August 1978.

[[Tanenbaum 1987a]] Tanenbaum, A; *MINIX: A UNIX Clone with Source Code*; EUUG Newsletter; Vol 7, No 3, pp 3-11; 1987.

[[Tanenbaum 1987b]] Tanenbaum, A; *Operating Systems: Design and Implementation*; Englewood Cliffs, NJ; Prentice-Hall; 1987.

Book Review

The UNIX Operating System, Second Edition
Kaare Christian, John Wiley, 1988, ISBN 0 471 84781 X. Reviewed by Syngen Brown, <syngen@king.ac.uk>, Kingston Polytechnic

I suspect that in writing this book, the author had no clear perception of his target audience. The overall impression is of an overambitious attempt to include as much material as possible, at the expense of organisation and consistency.

The book begins with yet another minor variation of the folklore surrounding the origins of UNIX, a rudimentary introduction to timesharing systems, and the inevitable "Noddy meets the Bourne shell" dialogue.

The second phase of the book is organised in thematic groups of commands. The selection of commands is creditable, together they convey a good illustration of the tool-oriented concept that is central to UNIX. However, in this coverage; the author oscillates between simplistic explanations of fundamental concepts and tedious descriptions of more technical nuance. On occasions, such material requires the understanding of concepts which have not yet been covered. It is particularly annoying that there is no clear demarcation between basic and advanced material. This is in contrast to the style of many contemporary books, where the need to interleave different levels of material is compensated by imaginative use of typographical devices.

An unusual feature is the inclusion of sections devoted to *sccs*, *yacc* and *lex*. These are important UNIX tools, but they are seldom used by the novice. It is even more unusual to find a

chapter devoted to system benchmarking. This, and the other later chapters, present an overview which misleads through oversimplicity. While such treatment of esoteric and advanced issues does add variety, it does not assist the novice and is no substitute for appropriate specialist materials.

A total omission arises in the case of the UNIX document preparation tools. The reason for this is given in the preface: Christian has another book dealing with that subject alone. Not having seen that work – it must suffice to say that, using a book on computer typesetting by Christian is rather like having Norman Tebbit as your vocational guidance counsellor.

As a reference work, the book fails for lack of coherent organisation. A prime example being the treatment of shell metacharacters. This material is distributed throughout the book, with little effort to present a unified concept. Although there is a section headed *Metacharacters*, this is restricted to those effecting filename expansion, leaving the novice with an erroneous impression of the concept.

This is a verbose work: a feature which will deter the novice by sheer information overload, and the experienced user by the effort to extract signal from noise.

To be fair, I did not find any major factual inaccuracies in the book. It is also the case that there are worse UNIX primers available. Criticism aside, it is inevitable that this book will appeal to some people, in particular those with no clear idea of what they are doing.

AUUG Conference Report

Jim Reid

*jim@cs.strath.ac.uk
...!mcvax!ukc!strath-cs!jim*

*Dept of Computer Science
University of Strathclyde
Glasgow*



Jim Reid has been the Systems Manager in the Computer Science Department at Strathclyde University for over seven years. Away from work, he enjoys playing the trombone and listening to music – not always the same thing! Other interests are Dundee United Football Club, beer and curries.

Getting There

Back in March last year, the Australian UNIX User's Group posted a news article advertising their forthcoming conference in Melbourne. The conference theme was to be "Networking – Linking the UNIX World". I decided to chance my arm by submitting a revised copy of my "Installing NFS on a 4.2 BSD VAX" paper that I had prepared for the Manchester EUUG conference in 1986. Nothing then happened for a few months and I duly forgot all about it, not wanting to build my hopes up.

Much to my surprise, I received a letter inviting me to the AUUG conference. Even more to my surprise, I somehow managed to get the money to go!¹

1. How the money was raised was a story of drama, intrigue, brinkmanship and eleventh hour grovelling. It doesn't bear repeating here. Special thanks are due to the UKUUG. Their support made all the difference when, at the last moment, it looked like my trip would have to be cancelled.

Day 1 – Tuesday 13th September

After two weeks of sightseeing in Australia, I arrived in Melbourne late on the Monday evening. This meant I missed the pre-conference dinner the AUUG had arranged for the speakers. After an early night, I was up bright and early next morning (honest!) to meet the AUUG executive and get to the registration desk. It turned out that the AUUG executive were all wearing suits, a fact that was the target of some derision by many of the delegates.

I picked up my conference folder, which had been sponsored by Nixdorf. Inside was a copy of the AUUG newsletter (the conference proceedings), my name badge, a notepad, some glossy advertising material and a rather tasteful AUUG lapel badge. It was green with a gold border, shaped like Australia containing "AUUG" in gold letters.

Opening Remarks

Peter Poole, Chairman AUUG

The meeting was formally opened by Professor Peter Poole of Melbourne University. He said that the AUUG had come a long way since it was

founded four years before. The current conference was by far the biggest and was the first to be held in a high-quality venue, the Southern Cross Hotel. He also announced that the AUUG had just been incorporated that week, giving the group a formal status in law.

*UNIX Networks:
Why Bottom-Up Design Beats Top-Down*
Michael Lesk, Bellcore

Mike Lesk's paper was the keynote address of the conference. He gave a historical perspective on the development of *uucp*. Essentially, it started as a means for him to automatically distribute software at Bell Labs. Electronic mail was an afterthought!

He then went on to analyse current networks such as the Internet and CSnet. He said that the early lessons had not been learnt and that current trends were opposite to the early days at Bell Labs. For rapid growth, computer networks should have no central administration. (This would obviously upset the *uucp* backbone sites, the NIC and the PTTs to name but a few.) Local administration should be minimal – one reason *uucp* has proliferated because it does not require modifications to the kernel or to most system software.

Looking into the future, Mike expected that data communications will get faster – i.e., higher bandwidth – but not much cheaper, even though processors will continue to get quicker and cheaper. As the PTTs provide newer technologies to the home, networking should become as easy as the telephone or fax. If our experiences of other mass communications – netnews, television, videotex – were anything to go by, electronic mail would be used for truly awful things when it becomes commonplace.

Coffee

Future Telecommunications Products
Steve Jenkin, Softway Pty.

Steve's presentation discussed some of the goodies that Telecom – the Australian PTT – were promising. "Just around the corner" were ISDN – Integrated Services Digital Network - on 64-Kbit digital lines, 34-150 Metropolitan Area Networks (MANs), Broadcast ISDN on 150-Mbit optical fibres, not to mention X.400. He doubted the ability of Telecom to deliver these services at the prices and in the timescale they envisage. Customers of British Telecom would probably

think "so what's new?".

*Overview of the Sydney University Network
Version IV*

Piers Dick-Lauder, University of Sydney

If you get confused between operating system versions and CPUs when someone says Sun-N (N = 2, 3 or 4), think of the Australian UNIX community. They also have a set of network protocols with the same name! Piers gave an overview of the changes made to their protocols to make it better – quicker, more reliable, cheaper – yet retaining backwards compatibility with older versions of the protocol.

Lunch

At lunch, I was joined by Greg Rose and Chris Maltby, President and committee member of the AUUG. Amongst other things, we discussed the organisation of our various national (or is it continental?) user groups. It turns out that the AUUG don't have anything like the backup provided to the EUUG and UKUUG by Owles Hall. The executive have to deal with all the AUUG's administration – memberships, subscriptions, conference arrangements, etc. – themselves. Until recently, the group's affairs were chaotic as it was difficult for the committee members to spare the time to do these tasks. I suppose this shows how much work is being done almost unnoticed by the people at Owles Hall. It certainly made me more appreciative of their efforts and of the work done by the AUUG committee for the Melbourne conference.

NeWS Programming

Tim Long, Sun Microsystems Australia Pty.

This talk was more of an overview of NeWS, Sun's PostScript-based window system. It covered the usual ground of event handling, lightweight processes and multiple drawing surfaces. In a sense Tim was trying to sell NeWS as the window system that would be all things to most users.

The X window system: An Overview

Tim Segall, Hewlett-Packard Australia Limited

An overview of the X window system was given by Tim Segall. In a fairly routine presentation, he mentioned the main X concepts – the X server, a bit-mapped display, the Xlib library, the interface to the X server protocol and so on.

*HOWS: A Window System**Greg Rose, Softway Pty.*

This was the comedy presentation of the conference. Greg showed a few slides of window systems: "X" (a glass skyscraper), "Colour" (some pretty stained glass windows) and "Open Look" (an unfinished house with no glass in the window frames). After the jokes, Greg told the meeting about HOWS (Horrible Old Window System). This was actually a small and rather nice window system that Peter Collinson of UKC had developed for the Atari ST. It used humble and cheap hardware to make a quite useful window system.

Coffee*An Implementation of FTAM**Andrew Worsley, CSIRO*

Andrew described the work he had been doing on implementing FTAM for UNIX systems at CSIRO. He made some comparisons between his implementation, the ISO/IEC 80002 FTAM and the 4.3 BSD *ftplib*. His version was about twice as big as the 4.3 *ftplib*, mainly because the FTAM protocol is more complex than *ftplib*. I was surprised that he was able to get data transfer rates for both FTAM implementations that were about 80% of those for the 4.3 BSD *ftplib*. FTAM had always struck me as a big, slow beast.

He commented that it was difficult to fully test his code, though it did work OK when tested against itself. Access to the ISO/IEC 80002 implementation had greatly helped testing and bug finding in both versions. He would have liked to be able to test his code against commercial FTAM software, but he found that vendors were very reluctant to let him have access to even beta-release versions of their products.

*IOS/OSI Under Berkeley UNIX**Mike Karels, University of California, Berkeley*

Mike's talk was on the work being done at Berkeley and elsewhere to integrate an ISO protocol stack into a future Berkeley UNIX release. It appears that most of the necessary bits are either ready or being worked on: X.25 sockets and ISO transport protocol software from the University of Wisconsin, X.400 from University College London and Nottingham University, the virtual terminal protocol from the Mitre Corporation and X.500 directory services from the National Bureau of Standards and perhaps UCL. Some of this software would be available in a

later ISO/IEC 80002 release, but all should appear in a subsequent Berkeley release.

Berkeley's work in this project will primarily be the kernel-level integration and updating of the socket framework to support ISO addressing. Some of this work would entail "kludging the clean design of the existing socket mechanism to cope with the warts in the ISO protocol stack". (I paraphrase Mike's words since the detail is too much to mention here.) Mike also touched on the work Berkeley had done towards giving BSD a POSIX interface.

Cocktail Reception

The first day closed with a cocktail reception in the exhibition area. I stuck to the Foster's Lager and Victoria Bitter since I've never been fond of drinks with fruit salad or wee umbrellas in them. I asked Mike Karels how the Berkeley internals book was coming along. Apparently it had gone to Jaap Akkerhuis for typesetting and that proof reading the first draft was completed. This meant that the book would be due to appear in late December or early 1989, which is how it eventually turned out.

Day 2 – Wednesday 14th September*Futures for UNIX Software**Larry Crume, AT&T Unix Pacific Co. Ltd.*

Larry is responsible for UNIX in the Pacific, Asia and Australia. He outlined AT&T's plans for the future of UNIX as "the platform for the 1990s". He said that Xenix, System V and Berkeley UNIX versions would be merged into one version called System V Release 4.0 by 1989. Application Binary Interfaces would be defined for the common processors – Intel 80386, Motorola 68000 and 88000, SPARC, the Clipper and the MIPS R3000.² It was not the case that AT&T were restricting UNIX to the SPARC, simply that the SPARC processor ABI was the first that AT&T had defined. He gave details of when to expect completed definitions of the ABIs for other processors.

He went on to explain what would be in System V Release 4.0 (SysVR4). The answer was just about everything: real time, POSIX conformance,

2. I assume it was an oversight that there was no mention of an ABI for the VAX or the 3B2.

internationalisation, X/Open capabilities and improved operation, maintenance and administration facilities. As part of the unification, the release would support BSD-style signalling, and it would take NFS (running on top of TLI) and Sun RPC from SunOS, as well as Sun's merged X/NeWS window system. SysVR4 would still have sockets, although they'd probably be implemented on top of streams and the TLI. NFS and RFS would both be present. A new version of SVID will be developed.

Larry spoke a little about the AT&T/Sun collaboration. SunOS releases from version 4.1 onwards would be SVID-compliant. (He didn't say if that was the existing SVID or the new one.) He also said that AT&T and Sun were already looking at System V Release 5.0. This would entail a kernel restructure and ideas such as using C++ and Mach were under consideration.

Larry said that AT&T would *participate* with OSF, but would probably not become a member. He underlined AT&T's stated commitment to open systems.

Open Look was mentioned and a videotape of a marketing presentation was available for people to see what it would look like. Open Look will be based initially on X11 with a NeWS version later. It will have a "standard look and feel" – and will be an open standard, with the source code being licensed. An application style guide was to be published by the end of 1988.

In the question and answer session, Larry was asked how much of SysVR4 would be unbundled. His answer was "you get what you pay for", so I suppose this means that things like troff will continue to be unbundled. Personally speaking, this is probably a good thing since I doubt if we'll have the disk space to take everything that would be on offer in System V Release 4.0. He would not be drawn on the issue of source licensing, saying the details had still to be worked out. However, I think he did say that licences would continue to be available to academics at "reasonable sums".

Future Berkeley UNIX Developments

Mike Karels, University of California, Berkeley

Mike's second presentation started with a brief description of the recent 4.3-tahoe release. He then went on to list the current projects being worked on at Berkeley. These are:

- routing protocols
- the Internet nameserver

- ISO/OSI networking
- a POSIX-compliant interface
- system interface updates
- the generic file system interface
- rearrangement of the UNIX filesystem

A new virtual memory system is under development. This will permit shared memory, file mapping, device memory mapping, copy on write and lightweight processes amongst other things. There will be one large sparse address space using multi-level paged data structures. Swap space won't be pre-allocated, but will be done through the file system. All of this will be integrated with the kernel memory allocator and the network and filesystem buffers. This might strike the reader as being similar, if not identical to Mach. This is because Mach is being used as a model for the VM system and Berkeley may even use some Mach code.

The next Berkeley release will have a POSIX-style *tty* handler and may go some way to providing a streams-like framework for the *tty* and network protocol handlers. There will also be some kernel clean-ups – the merging of the proc and user structures and a new sleep/wakeup mechanism.

On licensing, Mike said that Berkeley are considering using System V Release 2 as their base for future distributions so that they can use AT&T updated utilities such as *make*, *sccs* and *sh*. Earlier versions of BSD relied on the now defunct 32V UNIX licence. System V Release 3 won't be chosen because of its demands for SVID compliance.

Unified UNIX

John Young, Sun Microsystems Australia Pty.

This talk contained more or less the same information as Larry Crume's presentation. Sun and AT&T were working on a new merged UNIX that would have *everything* – NFS, RFS, sockets, streams, X, NeWS, POSIX compliance, SVID conformance (was that the old SVID or Larry's new one?), a nod in the direction of X/Open, all possible flavours of signals, System V this, Berkeley that and something else again from SunOS.

Coffee

Open Software

Tom Daniel, Hewlett-Packard Australia Limited

Tom described the Open Software Foundation that had been formed by Apollo, DEC, H-P, IBM and others. He explained (not at all convincingly to

me) how these companies would produce a common UNIX version that could somehow be proprietary to each member company. He said that OSF were using IBM's AIX as a starting point. Each company would submit its own technologies to a panel for assessment and incorporation into the OSF standard. Input from academic institutions would also be welcomed. The OSF members could then add their own features to the common standard to satisfy their customer's demands.

*ABI and OSF:
Technology and Future Impact on UNIX
Ross*

Ross began his presentation by showing a slide he had used at a USENIX conference three or four years ago. His slide (meant as a joke then) made ridiculous and incredible predictions. Ross was very surprised to see them come uncannily true. The ones I remember were:

- Bill Joy leaves Berkeley to join a start-up company
- DEC market UNIX
- AT&T buy up Bill Joy's company
- AT&T pay Bill Joy to produce a merged UNIX
- IBM adopt UNIX

He then tried to explain the chaotic marketing and standardisation efforts currently surrounding UNIX, showing the viewpoints and concerns of users, software developers and vendors.

Panel Session

The five speakers from this morning's session lined up for a panel session that turned out to be rather subdued. I had been expecting fireworks, but there were none. Robert Elz was lurking around at the back of the hall, but he was keeping silent.³

Most of the questions got the same answers. Who said what (I've paraphrased and perhaps slightly exaggerated their statements) is left as an exercise to the reader:

3. I gathered from other AUUG delegates that Robert was usually outspoken and always ready to make blunt, direct statements. I suppose he'd be the AUUG conference equivalent of Dave Tilbrook.

"OSF will win in the end as their standard is not proprietary"

"System V Release 4 will win as it will have everything in it"

"We're not really interested in standards?"

"Just wait for a year or two until the dust settles"

Lunch

I skipped lunch to take a long look round the exhibition. The biggest surprise of all was that IBM were there. They had a bunch of PS/2s and a couple of RTs (6150s) on show. There were also some glossy handouts on AIX/370. IBM now offer their UNIX — a System V/BSD hybrid — on top of VM on their mainframes.

Another surprise was the Sony stand. They were exhibiting their NEWS workstations which looked very nice. For me, the glossies had a lot of the right buzzwords in them — TCP/IP, 4.3 BSD, NFS, X-windows — to name a few. The machines had bitmapped screens and had two 68020 or 68030 processors — one for I/O and one for your application. The colour workstation had some very pretty demos and the quality of both the colour and monochrome displays was very good. Apparently, Sony went for 4.3 BSD instead of System V because it simplified the legal and licensing problems. The Sony rep. told me that System V would not be important to their marketing as they were going after the academic/scientific market. [They had some office automation software on show, so that could be another target for them.]

The Comperex stand proved popular. This was probably because they served free ice-cream. I liked the strawberry best.

Also on the Comperex stand was a video camera and frame buffer. The AUUG were taking video pictures for their face server project. By the time I got home and was back at work, an icon of my ugly face was waiting to stare at me from my mailbox.

*Distributed Trouble:
The University of Sydney Experience
with Networked Workstations
Rex Di Bona, University of Sydney*

This paper won a prize from the AUUG for the best student paper at the conference. Rex explained some of the security problems that he has encountered with a number of networked

workstations at the University of Sydney. Some of the attacks were mentioned were astonishingly simple.

Installing and Operating NFS on a 4.2 BSD VAX
Jim Reid, Strathclyde University

This may well have been the best paper at the conference, but then I'm biased! Actually I was thrilled and a little scared to make a UNIX presentation when Ken Thompson⁴ was in the front row of the audience.

Networks: Legal and Social Implications
James Watt, Griffith University

This was a thought-provoking talk that explored the legal issues involved in using computers and networks. James posed some interesting questions like:

"Can someone be sued over faulty software posted to comp.sources.unix?"

"How could a court prove that a user did something illegal on a computer?"

"Are printouts admissible as evidence?"

"Who, if anyone, can be prosecuted over obnoxious electronic mail?"

Sadly, the answer appears to be that the questions will only be resolved when someone is taken to court and that the legal profession does not seem to understand the problem.

Coffee

Human Interface Issues
Michael Lesk, Bellcore

Mike closed the second day with two talks that should be familiar to those who attended the UKUUG meeting at Newcastle in July 1987. His jokes and anecdotes were the same as the ones he used then. He talked about his experiments with an on-line library index at Bell Labs and then discussed his work on computerised direction finding.

When Mike worked at Bell Labs., he experimented with two on-line index systems for the library. The first used a simple keyword lookup taking as input a string that was either part of a title or an author's name. The second used a

4. If you're going to namedrop, you might as well make sure the name is one worth dropping into the conversation.....

top-down menu system based on the Dewey classification system. His results suggested that the keyword system was more popular. However, when he repeated the experiment with keyword and menu interfaces to an Associated Press news wire service, he found that there the menu-style interface was more popular.

The computerised direction finding work was interesting. It is a straightforward enough problem, but for me it was memorable for Mike's story of how a US car-hire firm wanted to exploit the idea and how it was killed by the bureaucracy in AT&T.

The Conference Dinner

Most of the delegates went off to the bar to kill the hour or so before the conference dinner. I got into conversation with Ken Thompson⁵ who started to talk about Plan 9, his new operating system. This was named after the notoriously bad 1950's science fiction B-movie "Plan 9 From Outer Space". We spent the rest of the time in the bar discussing this movie and other films in that genre. Apparently, they have quite a cult following at Bell Labs.

As I wandered into the dining room, I met George Michaelson of Yorkbox fame. Some of you may remember the acrimonious mail I used to send him when he worked at York University on the dreaded Yorkbox. Now he works in Australia (surely not because of my mail?) and this was the first time we'd met in person. I felt it was rather strange to travel 12,000 miles to meet for the first time someone I sort of knew who used to work only 200 miles away.

The dinner was sponsored by Pyramid, who gave everyone a small plastic pyramid with their logo on it. The pyramid was magnetic and came with some paper clips, so I reckoned it had a useful purpose apart from being a welcome advertising freebie.

John Mashey tried to give his talk about comparing software development to a military mission. Sadly, hardly anyone seemed to be paying attention to him. There was too much background noise for him to be heard and most folk looked to be more interested in the food and drink.

5. There I go, namedropping again....

The dinner became quite boisterous. There were lots of paper aeroplanes flying about and some of the helium-filled balloons ended up on the ceiling. The high spot came when one bloke collected all the balloons and tied them to a heavy piece of silverware that held a card with a table number on it. This thing must have weighed close to one kilogram and it ended up on the ceiling about 5 metres above us, suspended from about 50 balloons! Clearly, AUUG dinners can be as interesting as those held at EUUG conferences.

Day 3 – Thursday 15th September

AUUG ACSnetSIG Meeting

Thursday started with a Special Interest Group meeting on ACSnet – the Australian UNIX network that uses the SUN protocols. There was some concern about a plan by the Australian Universities to set up their own network. This would be a bit like JANET, though it would use high bandwidth (> 1 Mbit) connections. People were worried about how this would be funded, perhaps by stopping the money spent on ACSnet. There was also concern over the choice of protocols that may be used – TCP/IP, SUN, DECnet, SNA, ISO, even Coloured Books were being suggested! Some people felt the lines were too fast until it was pointed out that there was talk of running telephone and video circuits on the proposed network.

Link Management and Link Protocol in SUN IV

Bob Kummerfeld, University of Sydney

This talk was a followup to the one given by Piers Dick-Lauder the previous day. Bob went into much more detail on the new link level protocol used in version IV of the SUN protocol. The new protocol had been designed to be adaptable enough to work efficiently and cheaply over a variety of links including TCP/IP, telephone and X.25.

A New C Compiler

Ken Thompson, Bell Labs.

Ken had taken some sort of study leave and was working at the University of Sydney. He talked about a new C compiler he had been working on. The main design goal was that the compiler had to be *fast* – people tend to compile code more often than they execute it! The new compiler was also to be fairly portable, generate “not terrible” code and had to support ANSI C. The compiler was targeted for the NS32032, M68000 and CRISP

processors, though people were working on other targets such as the VAX.

The compiler takes 8 passes to convert C code into a .o file. The last pass is an assembler. Ken described some fairly quick but effective optimisation techniques that the compiler uses: for instance, a simple peep-hole optimiser, register allocation based on loop depth, assessing the costs of saving registers for subroutine calls (arguments are not popped after a call) and so on. There was no distinction made between automatic and external variables in what Ken called the compiler’s “registerisation”.

I managed to note down the times he gave for comparison between the Sun C compiler (*cc*) and his new compiler (*2c*). The times were for compiling *dc* – roughly 2,100 lines of C and 2,400 lines of *#include* files – on a Sun 3/60.

	<i>cc</i>	<i>cc -O</i>	<i>ld</i>
<i>cc</i>	34.8	57.0	0.3
<i>2c</i>	6.5	10.9	2.9

Interestingly, a version of *dc* that was compiled with the new compiler was about 10% quicker than one using the standard Sun-supplied compiler.

Coffee

Distributed Processing in the Queensland Government *Peter Waller, CITEC Queensland*

This paper discussed the development of computerisation of the Queensland State Government. The presentation explained how, in consultation with the University of Queensland, the state is making increasing use of UNIX throughout its departments. Peter discussed the particular problems of educating naive users and of networking and data communication, especially with sensitive installations such as the police computer.

Time Synchronisation on a Local Area Network

Frank Crawford, Q.H. Tours

Frank described the real problem of keeping time synchronised on the 4 Pyramids that Q.H. Tours maintain their database on. This was solved by using the 4.3 timed utility, though it had to be modified for his circumstances. Time could not be allowed to go backwards or stop for more than a second as this would break the database. A

mechanism was also needed to make the systems fault-tolerant. This was achieved by making all the computers keep time rather than by selecting a single master.

On MICROLAN 2 - An Office Network

Richard Lai, ICL Australia Pty.

This described a simple and cheap network developed by ICL for use in office environments.

Lunch

I had a lengthy chat with John Carey, the editor of the AUUG Newsletter, over lunch. The AUUGN (pronounced 'organ'), comes out every two or three months, much like the EUUG newsletter. He usually had difficulty getting contributions for the newsletter, much like the editor of the EUUG newsletter. I sympathised with him and, to salve my conscience, that's why I've prepared this article.

Writing Portable Programs

Stephen Frede, Softway Pty.

Stephen was unable to present his paper, so Greg Rose did it for him. The paper outlines the differences between the main UNIX variants – System V and BSD. The differences were discussed at the level of system call names and parameters, library routines (i.e., When do you use *index()* instead of *strchr()*, and *#include* files.

The paper was incomplete, but if Stephen does complete the job (if this is even possible), a finished version would be an invaluable aid to every UNIX programmer.

Real-time Disk I/O Scheduling on UNIX Systems

Jeonbae Lee, ETRI, Korea

The Electronics and Telecommunications Research Institute in Korea have been working with Samsung to develop a multiprocessor UNIX machine using M68000s. This machine has a similar design to the Sequent and Encore boxes. Lee described how the group at ETRI had modified the UNIX kernel (called KONIX – Korean UNIX) to give precedence to disk I/O requests from real-time processes. He produced a table that showed how little the response time for a real-time process degraded as the load on the computer increased.

The Process Life Cycle in STIX

Sunil Das, City University

Sunil wasn't as lucky as I had been in getting to Australia. Yours truly gave a presentation on the work that Sunil and Aaron Gull had done to port Minix to the Atari ST. I had also been asked to tell the Australians about the UKUUG, so I did that as well.

Coffee

RISC and the Motion of Complexity

John Mashey, MIPS Computer Systems

Sorry, but my notes for this talk have disappeared and there wasn't a paper in the conference proceedings. As I remember, John's talk was about the evolution of the MIPS RISC processor. He went into great detail about various trade-offs that the hardware and software engineers made to make the chip run fast. It turned out that the hardware people were able to do clever tricks that made things easier for the compiler writers. They in turn were able to do clever tricks that helped the hardware design people.

I also remember that John brought a few of the latest MIPS chips along for the audience to look at. He counted them all out and counted them all back in again, which was a pity. I would have liked another souvenir to bring home along with the AUUG badge.

Close

At some point over the three days, the AUUG held their business meeting. It was brief, even by UKUUG and EUUG standards. The committee was quickly re-elected and details were given of the 1989 AUUG Conference. This will be held in the Hilton Hotel, Sydney between the 8th and 11th August, 1989. The conference theme will be "Nobody Ever Got Fired for Using UNIX".

Final Impressions

I had thoroughly enjoyed the conference. The presentations had been of a high standard – at least as good as at any EUUG conference I've been to – and most of the speakers had something interesting and informative to say. I'm glad to say that UNIX is very much alive and well "down under". I hope that it won't be long before I get the chance to return there.

Journées UNIX

Rosalie Hurtado

Introduction

Rosalie Hurtado is a journalist specialising in UNIX. She has been the driving force behind several UNIX-oriented publications in France, the best known being */usr/info/sm90* which was dedicated to the SM90 workstation, and */usr/info* which is a more general UNIX publication.

Rosalie has also been active in promoting Grenoble as a centre of UNIX competence in France, and was heavily involved in the organisation of both the 'UNIX club of Grenoble' and the recent combined exhibition, conference and AFUU AGM, known as the 'journées UNIX' held at Grenoble.

This is her account of that meeting.

UNIX: Grenoble dans la course à la standardisation

Grenoble sort renforcée des journées UNIX organisée pour la première fois en province. Une avalanche d'informations a peine stabilisées et intéressant la capitale des Alpes ont émergées:

Le centre de recherche Européen d'OSF;

Le centre de recherche du constructeur de supercalculateurs, Alliant;

Et enfin la première adhésion universitaire française à OSF est Grenobloise (ENSIEG).

Une accumulation de petits succès, signe de l'attraction qu'exerce la ville sur le plan scientifique à un niveau mondial.

Il n'est pas nécessaire de dire que l'IMAG (Institut d'Informatique et de Mathématiques Appliquées), le Centre de recherche Bull, son groupe de compétence UNIX, les développements menés par Hewlett-Packard dans le domaine des réseaux, l'implication UNIX de Cap Gemini Innovation, Sema group et Cisi Informatique contribuent à ce dynamisme et à cet engouement pour UNIX.

Selon une étude publiée par le Comité d'Expansion Economique de l'Isère (Grenoble, pôle informatique européen), 80 sociétés grenobloises utilisent UNIX.

En augmentant son potentiel, Grenoble espère participer aux grands mouvements stratégiques qui se jouent aujourd'hui pour accélérer la standardisation d'UNIX.

Grenoble, la capitale des Alpes, a émergé des journées UNIX, renforcée par l'expérience. Il y a eu une avalanche d'informations, mais elles étaient encore instables et intéressantes pour la capitale des Alpes.

The establishment of the European research centre of OSF;

The centre of research for the manufacturer of the super-computer, Alliant;

And, at last, the first French university to join OSF is a Grenoble university – ENSIEG.

An accumulation of small successes, which are a sign of the attractions of the town of Grenoble for the scientific world.

It goes without saying that IMAG (the Institute of Computing and Applied Mathematics), the Bull research centre, with its UNIX competence centre, developments brought by Hewlett-Packard to the domain of networking, the implication of Cap Gemini Innovation in the UNIX world; the Sema group and Cisi Informatique all contributed to the dynamic feel of this event.

According to a study published by the Committee for the Economic Expansion of Isère (Grenoble axis of European Computing), 80 companies in Grenoble actually use UNIX.

In augmenting its potential, Grenoble hopes to participate in the strategic developments taking place today pushing forwards the standardisation of UNIX.

Il semblerait que chacun y aille du sien pour accelerer ce processus.

Objet de standardisation: les interfaces utilisateurs qui aujourd'hui sont loin d'êtres stabilisées.

Les resultats de l'appel d'offre d'OSF a ce sujet devraient permettre une avancee, selectionnant le meilleur produit du marche.

Gilbert Eloy, Directeur regional d'SM OSF (France, Benelux, Pays-Bas) souligne que le nombre des membres associes ne cesse d'augmenter:

"Nous sommes aujourd'hui une soixantaine, un grand utilisateur, Shell, nous a rejoint."

Jean-Claude Frobert, Directeur Marketing des systemes ouverts d'Unisys annonce l'ouverture du premier 'Open System Center Européen' a Zurich.

"Nous allons ouvrir 16 centres de competences UNIX d'obedience X/Open en Europe. Notre objectif est de tester des logiciels et du materiel et de decerner des labels officiels X/Open"

indique-t-il.

Un budget de 3,5 millions de dollars est consacré a cette operation qui montre qu'Unisys a fait d'UNIX son cheval de bataille.

Ce systeme represente 5%, soit 500 millions de dollars du chiffre d'affaires consolide d'Unisys. Selon une enquete Infocorp Unisys couvre 15 % du marche UNIX.

Faire partie du groupe de travail X/Open devient un argument commercial:

Top Log annonce que LPI (Language Processors Inc.), dont les produits sont diffusés par la société francaise, rejoint le groupe de travail X/Open, la norme globale, internationalement reconnue et supportée par CAE (Common Application Environment) devient un atout.

Synergie entre la recherche et l'industrie

Le salon, de petite taille, mais ou les plus grands fournisseurs de materiel UNIX etaient representes (IBM, Unisys, Bull, DEC, HP, Apollo, Alliant, Convex, Apple), a fait l'objet d'une grande premiere.

En effet, nos gourous UNIX francais:

Pierre Laforgue de l'IMAG, Eric Paire du Centre de Recherche Bull et Yves Devillers, responsable du backbone FNET a l'Inria,

It seems that everyone has to do his best to speed this process up.

One object of the standardisation process: user interfaces, which today are far from being stable.

The results of the 'Requests for Technology' made by OSF on this subject should permit an advance in this area, by selecting the best existing products on the market.

Gilbert Eloy, the regional director of OSF (France, Benelux and Holland), underlines that the number of members associated has not stopped growing:

"Today, we are sixty, a large user, Shell, has joined us."

Jean-Claude Frobert, marketing director of Unisys open systems, announces the opening of the first 'European Open Systems Centre' in Zurich.

We are going to open 16 centres of UNIX competence, following the X/Open philosophy, in Europe. Our objective is to test software and hardware to give an official X/Open label'

he said.

A budget of 3.5 million (US) dollars has been created for this operation, which shows that Unisys has made UNIX 'its battle cry'.

This system represents 5%, that is 500 million dollars, of turnover of the Unisys group. According to a survey by Infocorp, Unisys actually takes 15% of the UNIX market.

Being a part of an X/Open working group becomes a commercial necessity:

Top Log (a subsidiary of Metrologie) announces that LPI (Language Processors Inc.), whose products are distributed in France by Top Log, has joined the X/Open working group. The global standard internationally recognised and supported by the CAE (Common Applications Environment).

Synergy between research and industry

The exhibition, was small in area, but had representatives from the biggest UNIX suppliers (IBM, Unisys, Bull, DEC, HP, Apollo, Alliant, Convex, Apple), this is perceived as being a good 'premiere'.

Some French Gurus:

Pierre Laforgue of IMAG, Eric Paire of the Bull research centre and Yves Devillers, who is responsible for the EUnet backbone at INRIA,

ont decidé, non seulement de relier les ordinateurs entre eux grace a un cable Ethernet et a TCP/IP, mais également de creer une connexion entre ce reseau local heterogene et Internet (via Transpac). Ainsi le DPX2000 de Bull, sur le stand ANL (Agence Nationale pour le Logiciel) a servi de relai aux autres ordinateurs, qui, sur l'espace de deux jours ont appris a connaitre le monde specifique 'des News et des mails'.

Internet donnant ainsi acces aux deux plus grands reseaux heterogenes francais bases sur UNIX (l'Imag et l'Inria) et aux noeuds de communication avec les Etats Unis.

Grace à la premiere experience ont a pu voir un ecran Bull sur le stand de DEC par exemple, ou une fenêtre de Vax Station dessinait une image transmise par un DPX 2000 par X Window (X11).

De même X11 a marche entre le DPX 2000 du stand de l'ANL et le Stand Apple.

Le Mac II supportait le nouvel AUX (vu en France pour la premiere fois, mais non encore commercialise aux Etats Unis!). Cet AUX supporte non seulement X11, mais également un UNIX system V fort honorable, avec meme une caracteristique rare (car tres difficile a implementer sous SV): le 'job control' sous system V (control-Z sous csh).

Devant la camera de FR3, Eric Paire et Yves Devillers ont pu faire une demonstration en reel de l'interet de communiquer par les News:

"Nous avons recuperé le fichier de patch officiel de sendmail pour protection anti-virus en 21 secondes par un 'anonymous ftp' sur un serveur americain, via la liaison satellite entre Sophia et Princeton"

expliquent-ils.

A cette occasion, l'AFUU decentralisait pour la premiere fois sa conv les resultats de leurs travaux. Pour le grand public des plaquettes ont ete editees:

Vivre avec UNIX, réussir avec UNIX.

"La bande 'benchmark' SSBA est aujourd'hui connue de tous les constructeurs"

indique-t-il.

Un sous-groupe 'Benchmark-SGBD' s'est donne pour objectif

decided to not only to interconnect their machines via Ethernet and TCP/IP, but also to make a connection this heterogeneous network and the Internet (via TRANSPAC). A DPX2000 from Bull, on the stand of ANL served as a relay to the other machines, and during the two days of operation introduced everyone present to the world of News and mail.

The Internet connection thus gave access to two large heterogeneous French networks based upon UNIX (IMAG & INRIA), and to communication nodes in the USA.

Thanks to this effort, it was possible to see Bull software via a machine on the DEC stand, or watch a window on a Vax Station drawing an image transmitted by a (Bull) DPX 2000 running X Windows.

The same X11 setup worked between the DPX 2000 on the ANL stand, and the Apple stand.

The Mac II was running AUX (seen in public for the first time in France but not yet commercially available in the USA). AUX supports not only X11, but also a complete System V, with a rare characteristic (because it is difficult to implement on SV): 'job control'.

Before the TV cameras of FR3, Eric Paire and Yves Devillers were able to give a real time demonstration of the Internet being used to read News:

"We have transferred the official patch for sendmail, to protect against 'virus infection' in 21 seconds by anonymous ftp, from an American server, via satellite, between Sophia (in the south of France) and Princeton"

they explained.

On this occasion, the AFUU decentralised, for the first time, its 'Convention AFUU', during which the results of their various working groups were made public. They had produced some literature for the public:

Living with UNIX, and succeeding with UNIX.

"The benchmark tape (SSBA) is now recognised by all of the manufacturers"

they said.

A sub-group 'Data Base Benchmarking' has set itself the following objectives:

"d'offrir le plus tot possible une version 0, d'une suite de benchmarks SGBD-independant et UNIX-portables significatives des performances essentielles des SGBD sous UNIX",

dixit J.M. Saglio, responsable de ce groupe.

Dans le groupe de travail Securite, CHAOS (pour Controle Hermetique et Automatique des Objets Sensibles sous UNIX) a été mis au point pour debusquer les pirates.

Selon les propos de Philippe Dax:

"Face a la multiplicité des attaques sur les systemes informatiques accessibles a distance, les outils d'audit, preconises dans le Trusted computer System Evaluation Criteria (TCSEC) dit 'Livre Orange', sont devenus necessaires. Sous UNIX, il n'existe pas, de façon satisfaisante, d'outil standard qui permette d'auditer un système. CHAOS est un logiciel qui essaie d'y remedier."

Les systemes informatiques se complexifiant de plus en plus, ils entraînent une fragilisation globale de l'organisation. UNIX ou pas UNIX, tous les systemes communicants sont loges a la même enseigne!

"To offer, as soon as possible, a 'version 0', of a benchmark suite, which will be data-base independent, and UNIX portable, which will give an indication of the essential performance of a data base system under UNIX",

said J.M. Saglio, the head of this group.

In the working group on security, CHAOS, there was an explanation of steps to take to deter pirates.

According to Philippe Dax:

"Faced with the multiplicity of attacks on computing systems accessible via networks, the auditing tools recommended in the Trusted Computer System Evaluation Criteria (TCSEC), also known as 'The Orange Book', have become necessary. UNIX doesn't have, at least in a satisfactory way, any standard tools to perform a system audit. CHAOS is a system which will try to remedy that problem."

Computing systems become more and more complex, and they cause a certain fragility in the global organisation. UNIX or not, all systems which communicate are sailing under the same flag!

EUUG Executive Report

Helen Gibbons
euug@inset.co.uk

EUUG
Owles Hall
Buntingford, UK

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

The EUUG Executive Committee held its last meeting of 1988 on the 12th December. The meeting was hosted by AFUU at its offices in the Kremlin-Bicetre in Paris, which incidentally must be in one of the quietest locations ever, as it overlooks a vast graveyard! The AFUU Secretary, Anne Garnery, whom many will remember from the Paris Workshop, made everyone very welcome and helped greatly with transport, as the date fell in the middle of the Paris strike. Teus Hagen chaired the meeting which was attended by Vice Chairman, Michel Gien; the Treasurer, Nigel Martin, who was suffering from a very severe bout of influenza; Philip Peake, Publications Executive; Kim Biel-Nielsen, P.R. Executive; Ernst Janich, Conference Executive; Daniel Karrenberg, Network Executive; and the Business Manager. Unfortunately Neil Todd, the second Conference Executive, was unable to attend this meeting. This did not however prevent him from doing a great deal of work behind the scenes.

As always, a great deal of time was spent on examining the financial position of EUUG, and financial sheets were presented which showed that the present situation was well up to budget and that a healthy operating surplus had been achieved giving reserves almost equal to one year's

operating costs.

In order to be truly European a decision had been taken to send out the 1989 subscription invoices in ECUS. These went out to all the National Groups in January but it remains to be seen whether they will find this a convenient currency to use as it is subject to fluctuation and to bank charges. The Executive Committee will be interested to hear the Groups' views on this.

At the last Governing Board Meeting in Portugal the Executive was mandated to implement a system of debtor control that would result in the fair treatment of all groups, since from experience it is known that some pay their subscription invoices promptly, while some pay only after several months. The following scheme was therefore designed and implemented at the beginning of the year.

1. Invoices are issued in accordance with current policy requesting payment within 30 days of date of invoice.
2. Groups will attract a credit of 2.5% of the invoice value if the money is received within this time, the credit being offset against future invoices.

3. For payment between 30 and 60 days the invoice remains at the amount stated.
4. After 60 days 2.5% per month interest will be charged from the date of the initial invoice on the outstanding balance.

As the aim is fairness to all Groups and not the general benefit of the EUUG, any interest received will be offset against discounts and paid back to the Groups in the form of a credit note against their October invoices.

It was noted that some Groups had still not paid their 1988 October invoices, but on the whole the Secretariat was congratulated for having done an excellent job on debt collection.

The Committee also deliberated on new financial control measures which would make each officer responsible for the expenditure of the allocated budget under his control with a purchase order system being operated from Owles Hall.

The Committee greatly welcomed the return of the German Group, GUUG, to the EUUG and agreed that, as a gesture of goodwill, the last issue of the 1988 EUUG Newsletter should be sent free of charge to all the members of the German Group along with a welcoming letter.

The Portugal Conference was reviewed and it was thought to be a great success with approximately 240 delegates at the Conference and some 270 members attending the Tutorials. Although not all the bills had yet been received it was forecast that the event will have generated a sizeable surplus.

Thereafter the Conference Officers reported on the Brussels Conference on the 3-7th April 1989, and the Vienna Conference on 18-22nd September 1989, both of which were well advanced. Full details of the Brussels Conference appear elsewhere in this newsletter. The conference Officers have also already started work on the Munich Conference on 23-27th April 1990 and the French Conference in Nice on 15-19th October 1990. Plans are also afoot for a Conference on Norway in Spring 1991.

The executive Committee then turned its attention to publications and learned that the Publication Executive, Philip Peake, was looking into the possibilities of increasing the Newsletter to six issues per year. Exchanges with other publications were discussed, but it was noted with regret that the sale of the 4.3 BSD Manuals, now stocked by the EUUG Secretariat was disappointingly low. These well produced glossy manuals sell at £60 per set of seven including postage and packing and are available on order from Owles Hall for immediate delivery.

The main item under Public Relations was the decision to order an EUUG portable stand which could be used at various exhibitions and events. A budget of £3000 was set aside for the provision of a suitable stand which could also be loaned to the National Groups for display at their events. A slide show describing the EUUG had already been compiled by Kim Beil-Nielsen.

Daniel Karrenberg impressed the meeting by tabling a draft of the EUnet Directory, a document which had taken a great deal of work but will surely prove very worthwhile and already a great many forward orders have been received for it. It is now going into production and will be available soon at a cost of £18.

The final session of this very busy meeting, which went on all day with only the briefest break for a light lunch, was devoted to the planning of the EUUG Workshop to be held in Dublin on 5-7 May this year. The purpose of the Workshop is to discuss objectives, strategies and policies to go forward to the Governing Board. It will look at existing services to see how they can be improved and consider what new ones might be beneficial.

The next meeting of the Executive Committee will be held on 13th February in Amsterdam. The policy continues to try to meet each time in a different country, but all possible efforts are made to keep meeting costs to a minimum.

Dutch User Group Report

Frances Brazier
frances@psy.vu.nl

*Department of Cognitive Psychology,
Vrije Universiteit,
Amsterdam,
The Netherlands*



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

News from the Netherlands

Conferences

Our last conference was a success! Never before have we been confronted with the problem of just not having enough room. The programme included tutorials, technical presentations and an exhibition, all open to all participants. Approximately 300 people attended the conference, including members of the BUUG, DKUUG, FUUG, UKUUG and GUUG. Our first experience with the publication of proceedings was well received.

At the moment the NLUUG is busy preparing for our next conference. May 9th, on human-machine interfaces. As the programme is still not definite, interested readers should either read news or contact the NLUUG at a later date. Our experience with 'foreign' (non-Dutch) speakers at our last conference on standards, was promising. Not because the speakers were not able to speak Dutch but because they were able to provide first-hand information. This does not imply that the Dutch speakers were any less qualified or less interesting - in contrary - but additional contributions from 'abroad' were well

appreciated. Finding good speakers willing to give a presentation at a level which can be appreciated by most of the audience, is difficult. But I know we're not the only group or organisation with this problem.

The backbone

Since the first of January *mcvax* is free of the burdon of being both the Dutch and the European backbone. The new Dutch backbone, *hp4nl* is now responsible for all Dutch mail and news. Although still stationed at the CWI, the NLUUG soon hopes to take over its maintenance and support, at least financially. We are currently in the process of looking for candidates.

Rumour

Rumour has it that the crypt competition will be of a more impressive calibre than originally planned. Time will tell.

AFUU Diary

*Philip Peake
philip@axis.fr*

Axis Digital, Paris



November 1988

Last week (18th of November) was the AFUU AGM. Since the AFUU constitution states that governing board members are elected for a period of three years, and that they cannot re-present themselves for election for at least one year, I thought that I had written my last article for the AFUU slot in the EUUG newsletter.

Alas, it seems that I was mistaken. Due to a postal strike in France, virtually no-one had received their voting papers, and so an EGM (because we didn't attain the quorum for the AGM – it gets difficult with 700 members!) decided that the existing governing board should rest in place until the postal strike is over, and we can re-start the election process.

So, once again, I have to face the prospect of Alain Williams printing that *horrible* photograph that he keeps just to annoy me ...

This election is more interesting than those of the past, because, for the first time we have more candidates than posts available. 17 candidates for 7 posts.

In some ways it is unfortunate that our constitution limits the number of members of the governing board, because we certainly have work for as many people as we can find.

Apart from the problems of voting, the AFUU meeting seems to have been successful. There were presentations from each of the working groups, explaining what the group does, and what results they have generated so far. It was unfortunate that all had to fit into one day, and so some of the presentations were in parallel, meaning that people had to decide which presentation to miss.

The AGM took place in Grenoble, taking advantage of the UNIX exhibition and conference organised by 'The UNIX club of Grenoble'. This conference took place on the 17th and 18th of November.

Now we have to start preparations for the *Convention UNIX '89*, which takes place in Paris at the end of February. As well as this, we have a certain amount of work to do to help the EUUG with preparations for the conference which is due to take place in the south of France in autumn 1990.

When the elections are over, and the new members of the governing board are installed, I wish them good luck, they have lots of work already lined up for them ...

January 1989

On the 5th of January, we have (at last!) held our EGM, and I can safely say that I am no longer responsible for writing the AFUU page for the newsletter; but since I had already started ...

The ballot was carried out by post, and for the first time, we had a computer program to perform the counting, ordering and production of statistics –

thanks to Michel Sutter for that.

Being seasoned computer users, there was also a parallel manual count ... Surprising as it may seem, both systems agreed!

The governing board of the AFUU (Conseil d'Administration or CA as it is called in French) is as follows:

Name	Organisation	Responsibility	Duration
Jean-Luc Bellet	AT&T		3 years
Jean-Louis Bernard	Consultant		1 year
Christophe Binot	Hewlett-Packard	Vice President	3 years
Nicole Blaine	Sun Microsystems		3 years
Matthew Capril	Bull		3 years
Anne Francois	Hommes & Communications		1 year
Jaques Guidon	Agence National du Logiciel		3 years
Dominique Maisonneuve	Sligos	President	3 years
Veronique Mansart	Dataware		2 years
Pierre-Louis Neumann	Inria	Treasurer	2 years
Jean-Christophe Petithory	University (Paris VIII)		3 years
Jean-Jaques Rousset	Inforama		2 years
Alain Saint-Patrice	Comtec		2 years
Michel Sutter	Matra Datasystems	Secretary	2 years
Philippe Vaudou	Individual (user)		1 year

As you can see from the *duration* column, the scheme of retiring one third of the board each year is now fully in effect (the *duration* column indicating how much time the various people have left to serve – those with 3 years are the people just elected).

The 'executive' (or Bureau) is elected by the board each year, congratulations to Dominique Maisonneuve on being elected President for a second time!

UKUUG News

Mick Farmer
<mick@cs.bbk.ac.uk>

*Department of Computer Science
Birkbeck College*



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

Reorganisation

The UKUUG Committee was recently restructured into an Executive Committee and an Advisory Committee. The Executive Committee consists of the elected officers (Chair, Secretary, and Treasurer) and the Advisory Committee consists of those helping in various ventures such as Meetings, Workshops, and Networking. Collectively, we can be reached as ukuug@ukc.ac.uk.

UKUUG and UKnet Winter Technical Meeting

This was held over two days on December 19/20, 1988 at the University of Kent. The annual UKnet meeting needed some quick scene shifts, as we had a lot of good things to get into one afternoon and one morning. The reason for this was the lack of a summer UKUUG meeting, because of the EUUG meeting in London. We therefore decided to start with a UKUUG afternoon featuring several talks previously given at EUUG and USENIX meetings, along with a couple of new talks.

The annual UKUUG business meeting followed. Sunil Das (Chair) and Mick Farmer (Secretary) were re-elected. Zdrav Podolski (Treasurer) reported on last year's accounts and this year's projected expenditure. His report was approved.

The evening session followed the usual *Bar, Dinner, Bar*, meeting schedule.

Tuesday started early as we had a lot of network talks to get in followed by a *UKUUCP* tutorial in the afternoon.

The general feedback suggested that most people enjoyed themselves and found the talks interesting. The tutorial was particularly popular with Lee McLoughlin even making UUCP intelligible to us, which shows the high level of presentation. All in all it was an enjoyable and, we hope, worthwhile two days.

UNIX Security Workshop

We are holding a Security Workshop on 16 February 1989 at the Institute of Education, London WC1. It provides UNIX System Administrators with an insight into the security problems relating to UNIX systems, and will act as a forum to share knowledge and experience.

UKUUG Summer Meeting

The UKUUG Summer meeting will be held at Strathclyde University, Glasgow on June 27/28. Details are being finalised, but we hope that the meeting will reflect some of the interesting things being done with UNIX by people in the West of

Scotland. There may be a small exhibition.

Following the success of the UKUUCP tutorial at the Winter Technical meeting, the UKUUG is considering running tutorials at this meeting. Suggestions for tutorials and speakers can be made to the UKUUG(ukuug@ukc) or to the local organiser.

Local arrangements are being made by:

Jim Reid
 Dept of Computer Science
 Strathclyde University
 Glasgow G1 1XH
 United Kingdom
 Phone: (+44) 41 552 4400 x3319
 Email: jim@cs.strath.ac.uk

London UNIX User Group (LUUG)

The LUUG lecture series is now well-established. The first two lectures by Steve Kille on *ISODE*, and Eddie Bleasdale on *CUE* were very popular.

The 1989 programme starts with a security discussion on 26 January. February's topic, *Making software installation easier*, is another discussion meeting, which has the potential to turn into a major project. Dominic Dunlop will be talking about *Technological red herrings* on 30 March. Ray Anderson will de-mystify *X-windows* on 27 April. Jim Crammond is the speaker on May 25, when he will describe the new *UK-Sendmail* configuration package. Beyond that? Well, watch **uk.events** for details!

Glasgow UNIX User Group (GLUG)

This local user group is hoping to have its inaugural meeting sometime in February 1988. We wish them well.

That's all for now folks!

Mick Farmer
 Secretary, UKUUG

Calendar of UNIX Events

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

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

Abbreviations:

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

year mon days	conference	(sponsor,) (hotel,) location
1989 Feb 28-Mar 2	UniForum	Moscone Center, San Francisco, CA
1989 Feb 28-Mar 3	U Convention	AFUU, Paris, France
1989 Mar 1-2	GOSIP Users' W	NIST, G, MD
1989 Mar 2-3	GUUG W	Saarbrueken, Germany
1989 Mar 30	LUUG (UKUUG)	18 Bedford Square, London
1989 Apr 3-7	EUUG	Palais des Congres, Brussels, Belgium
1989 Apr 10-11	ANSI X3J11	Phoenix, AZ
1989 Apr 18-20	IETF	IAB, (NASA, Kennedy Space Center, FL)
1989 Apr 18-20	IETF	IAB, (PSC, Pittsburgh, PA)
1989 Apr 24-28	IEEE 1003	Minneapolis-St. Paul, MN
1989 Apr 25-27	ISDN in Europe	IFIP-TC6, ICCC, The Hague, Netherlands
1989 Apr 26-29	Networking Forum '89	Sendai/Tokyo, Japan
1989 Apr 27	LUUG (UKUUG)	18 Bedford Square, London
1989 Apr	Soft. Management W	USENIX, New Orleans, LA
1989 May	U 8x/etc C&T	/usr/group/cdn, Toronto, ON
1989 May 8-12	DECUS S	Atlanta, Georgia
1989 May 9	NLUUG	The Netherlands
1989 May 14-16	AMIX	Kfar Hamaacabia, Israel
1989 May 16	POSIX Appl. W	NBS, G, MD
1989 May 25-28	ENA conference	Allentown, PA
1989 May	U 8x/etc C&T	/usr/group/cdn, Toronto, ON
1989 Jun	NZSUGI	New Zealand
1989 Jun 7-9	Italian U C	i2u, Milan, Italy
1989 Jun 7-9	COMUNIX	/usr/group/UK,
1989 Jun 7-9	Eur. U User Show	EMAP Int. Exh., Alexandra Palace, London
1989 Jun 12-16	USENIX	Hyatt Regency, Baltimore, MD
1989 Jun 19-23	ICX89	USM, IEEE, Valparaiso, Chile
1989 Jun 27-28	UKUUG	Glasgow
1989 Jul 10-12	13th JUS UNIX S	Tokyo
1989 Jul 10-14	IEEE 1003	San Francisco, CA

1989 Jul 26-28	IETF	IAB, Stanford, Stanford CA
1989 Aug 8-11	AUUG	Hilton Hotel, Sydney
1989 Sep 18-22	EUUG	WU Vien, Vienna, Austria
1989 Sep 19-22	ACM SIGCOMM	Austin, TX
1989 Oct 16-20	IEEE 1003	Brussels, Belgium
1989 Oct	UNIX Expo	New York, NY
1989 Oct 31-Nov	2 IETF	IAB, U. Hawaii, Honolulu, HI
1989 Nov 1-3	UNIX EXPO	Javits Conv. C, New York, NY
1989 Nov 6-10	DECUS S	Anaheim, California
1989 Nov 9	NLUUG C	The Netherlands
1989 Nov 9-10	JUS 14th S	Osaka
1989 Nov 24	AFUU C	Paris, France
1989 Dec	UNIX Fair	JUS, Tokyo
1990 Jan	U in Gov. C&T	Ottawa, ON
1990 Jan 22-26	USENIX	Washington, DC
1990 Jan 23-26	UniForum	Washington Hilton, Washington, DC
1990 Jan 29	IEEE 1003	New Orleans, LA
1990 Feb 6-8	IETF	IAB, (FSU, Tallahassee, FL)
1990 Mar 27-30	AFUU	Paris, France
1990 Apr	IEEE 1003	Montreal, Quebec
1990 Apr 23-27	EUUG	Munich, Germany (tentative)
1990 May 2-4	IETF	IAB, (U. Washington, Seattle, WA)
1990 May 7-11	DECUS S	New Orleans, Louisiana
1990 May	U 8x/etc C&T	/usr/group/cdn, Toronto, ON
1990 Jun 11-15	USENIX	Marriott, Anaheim, CA
1990 Jul 31-Aug	2 IETF	IAB, ?, not in North America
1990 Autumn	EUUG	south of France
1991 Jan 21-25	USENIX	Dallas, TX
1991 Jan 22-25	UniForum	Infomart, Dallas, TX
1991 Feb	U in Gov. C&T	Ottawa, ON
1991 Spring	EUUG	Tromso?, Norway (tentative)
1991 May	U 8x/etc C&T	Toronto, ON
1991 Jun 10-14	USENIX	Opryland, Nashville, TN
1992 Jan 20-24	USENIX	Hilton Square, San Francisco, CA
1992 Jan 21-24	UniForum	Moscone Center, San Francisco, CA
1992 Jun 8-12	USENIX	Marriott, San Antonio, TX
1993 Jan	USENIX	northeast North America
1993 Mar 2-4	UniForum	Washington, D.C.

Organising Bodies

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

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

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

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

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

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

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

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

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

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

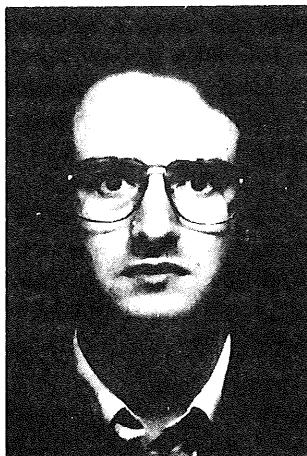
USENIX Association Office
 P.O. Box 2299
 Berkeley, CA 94710
 USA
 +1 415 528 8649
 office@usenix.uucp

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

Introduction to Window Systems

*William Roberts
liam@qmc-cs.UUCP*

*Department of Computer Science
Queen Mary College
London, UK*



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

Introduction

Hello! This is the first article in a regular column on window systems: I hope to cover more or less everything that combines UNIX with on-screen graphics. If there are any questions you have, any products or systems you'd like more information about, let me know and I'll try to get an article from someone who knows. If there is anything you'd like to write an article about, mail me and I'll be happy to oblige.

Lined up for the near future are articles on; X11 and the Atari ST, Presentation Manager and OS/2, and Open Look (the new interface specification from Sun/AT&T). To start the column off, I'll summarise the current major systems and where the UNIX+graphics world seems to be going at the moment.

Proprietary Systems

Most manufacturers of 'UNIX workstations' supply a fairly powerful machine intended for use by a single person sat at a graphics screen; such systems always include a *window system* which allows ways of sharing the screen between several applications and has a programming interface for writing graphical programs. Familiar examples

include the Sun Microsystems workstations with the SunWindows and SunView systems, Apollo workstations with the Domain Graphics Manager, and Hewlett-Packard with Windows 9000. The main feature of such systems is that they are proprietary: the programming interface is only available on one manufacturer's machines, and a supplier writing graphics programs for several machines has to have a different version of the source code for each machine. This can be extremely annoying, especially when the graphical techniques are actually the same in two systems and all that changes is the *typedefs* and the library routine names.

Standards

The usual weapon against proprietary systems is non-proprietary standards, and the graphics world is no exception. Early graphics standards which are available on some machines include the ACM SIGGRAPH standards CORE and CGI, the ISO GKS (Graphics Kernel System) standard and the most recent, PHIGS (Programmers Hierarchical Interface to Graphics Systems). The distinguishing feature of these systems is that they are all derived from the world of Computer Aided Design, and the use of *vector graphics* which

describe the display in terms of lines and filled areas. These are usually aimed at producing realistic pictures of three dimensional objects or two dimensional graphs with labelled axes, etc. Neither of these is appropriate to the typical use of a UNIX workstation, where the one graphics display is used to provide multiple terminal windows, each of which behaves like a separate VDU screen. To tackle these *raster graphics* displays where the final image is composed of a rectangular array of dots (called *pixels*), a new collection of standards and/or proprietary systems has been formed.

Raster Graphics Standards

There are a number of systems which have some claim to be called 'standards', though none of them have been accepted by any of the national or international standards-making organisations. The main ones are listed below, and summarised in the following sections.

Presentation Manager

This is important because it is the standard window system with the IBM/Microsoft OS/2 operating system. It is proprietary and runs only on OS/2, which implies an Intel 80286 or Intel 80386 processor and a machine that is very like one of the flavours of IBM PC or PS/2. However, anything that IBM does will affect most of us whether we like it or not, so this one counts as being "important".

Macintosh

The Macintosh window system is another proprietary one, but is of some importance in that it is very successful and has inspired almost all of the other systems: so much so that Apple are currently suing Microsoft and Hewlett Packard for stealing the "look and feel" of the system by copying it too closely. Don't expect this to appear on any other machines.

X

This is the window system developed originally by MIT, but now actively promoted by the X Consortium, a large group of computer companies including all of the graphics workstation manufacturers you can think of (e.g., Sun, DEC, Apollo, Hewlett Packard, IBM) and some surprises (Kodak-Eastman, Cray Research). It is the window system chosen by the Open Software Foundation, and also features in the plans of the AT&T/Sun/Unisys grouping now known as

'UNIX International' (a ghastly name!), so it is likely to feature in all new UNIX systems for the next few years.

NeWS

This is a development by Sun Microsystems, but is available under licence and has been ported to other systems. Its main claim to fame is that a combination of NeWS and X will be the window system distributed as standard with AT&T System V release 4, as part of the Sun/AT&T commercial collaboration.

Having listed the important systems (if your product isn't in this list, I'm afraid that you aren't currently important...), what are the strengths and weakness of each, and which ones are in direct competition with each other?

Macintosh

This system has no direct competition because it only available for the Macintosh, but its pride of place in the Macintosh world is being slightly eroded by Apple's decision to supply the X system with their A/UX version of UNIX System V.2.2 in addition to supporting parts of the Macintosh system. The problems stem from the architecture assumptions used in the original Macintosh, which didn't have a multiprocess system or memory management hardware. Its only advantage is the ability to run 'well-behaved' Macintosh applications.

The Macintosh QuickDraw library is fairly typical of current graphics libraries. The facilities are more complex than traditional pen plotters and fall into three groups.

The first group are operations on *rasters* (also known as *bitmaps*), which are rectangular arrays of dots. Each dot is called a *pixel* and is a number indicating the colour to be displayed on the screen; these are normally packed into memory as tightly as possible, so a *monochrome* bitmap with just the colours white or black will be stored as one bit for each pixel, 8 pixels to a byte and so on. Operations on bitmaps combine each pixel in one *source* bitmap with the corresponding pixels in another *target* bitmap (sometimes two others), replacing the pixel in the target bitmap with the result. The combination of pixels is usually done as a bitwise logical operation (AND, OR, XOR, etc.) on the numeric representation of each pixel, and the whole process is called *RasterOp* or *BitBlt*. If the source bitmap is smaller than the target, it will be *tiled* by placing (imaginary) copies of the source raster next to the original

access to mouse and keyboard events. The purpose of a window system is to share the physical screen between several applications, without those applications having to know in advance about each other; this is similar to the task of the kernel in sharing the available CPU time and memory between applications. The usual technique is to let each application create one or more *windows* which are treated like the whole screen, but to combine these 'virtual screens' together on the real screen. This involves distancing the application from its window and requiring that all window operations are carried out via the window system: in particular, the part of the screen used for one window may be partly hidden by another window which has been placed 'in front', so applications cannot write directly into screen memory. For this reason, applications have to deal with the problem of redrawing the content of their window whenever necessary, because the window system may not guarantee to remember everything that was drawn before.

Turning back to the specifics of Presentation Manager, its main advantages are the notion of presentation and device spaces, the lightweight process mechanism in OS/2, and its strategic role in IBM's announced plans. A *presentation space* is a data structure which filters application graphical commands before passing them to the main part of the window system. This allows an application to change coordinate systems and other such graphical transformations, but also includes two possible ways of avoiding the 'redraw your window' problem; first, an off screen copy of the screen window, so missing portions can automatically be transferred to the screen from the copy, or second, a *display list* which stores a list of all the graphics commands and can replay them when necessary. A *device space* is another filter which deals with device dependent issues and copes with the multitude of different graphics hardware available for IBM machines; a closer UNIX analogy might be that of a module in the System V.3 STREAMS system. The OS/2 lightweight process mechanism is used in Presentation Manager to allow multiple event loops, so that large parts of the application can operate in parallel. This of course increases the kernel-like complexity of applications, but it is useful and does make some things much simpler. Finally, Presentation Manager is part of a grand IBM vision called SAA, so someone in IBM believes in it.

Presentation Manager is occasionally mentioned on the *comp.windows.misc* newsgroup.

X

As its supporters keep having to point out, this is "a window system called X, not a system called X-Windows". X is really the main window system to watch in the UNIX world because all of the graphics system manufacturers are supporting it, because it offers device-independent graphics that can operate over network connections, and because versions of it are available free (including source code).

X is a *distributed window system*. Recall that an application has to manipulate its windows indirectly by asking the window system to carry out graphics operations; X takes this notion a step further and allows the window system to be on a separate computer with the application requests passed through a network connection (or even a simple serial line). The part of the system which actually carries out the work is called a *graphics server* and has to run on a machine with a graphics screen (and usually a mouse and keyboard as well); the application end is called a *client* of the server. It is entirely possible for several clients to use the same server, and for those clients to be on different machines. Pushing this idea further, X defines an *abstract graphics device* with a fixed set of operations, and any particular graphics server will implement those operations on one particular type of hardware, emulating the abstract device. X is carefully designed (as the result of experience with using earlier versions, we are now at X version 11 release 3) so that it can be implemented on almost any raster graphics display currently in existence, and so that almost any existing window system could be built on top of X: for this reason X is known as a *platform* and is said to "implement mechanism, not policy". In principle, proprietary systems such as Presentation Manager and the Macintosh system could be built as libraries on top of X, allowing Macintosh applications to run using an IBM graphics server or Presentation Manager applications to appear on your Macintosh screen.

The most important things about X are that it covers the things which people know how to do well (overlapping rectangles), it leaves open issues which aren't settled, e.g. (how to handle colour) so that existing systems fit into the model, and it does nothing else. It is a rationalisation exercise intended to remove once and for all the

repeatedly until the target is completely covered; this is used to do cover large areas with a regular pattern. If three bitmaps are used, the third one is called a *mask* and controls which pixels in the target will be modified and which will be left alone. To put a complex shape onto the screen, a mask is prepared with all of the pixels in the shape set to one, and all the other pixels set to zero: using this as a mask when copying the source into the target ensures that only the pixels which form part of the shape are changed. Combining the use of a mask with tiling the source raster gives the ability to *fill* complex areas with a colour or pattern.

The second group of operations involve drawing lines and curves: they describe areas to be filled using the raster operations from the first group. Lines are drawn using a *pen* which will have a shape (given by a mask raster) and a colour (given by a source raster). Lines are drawn by working out the pixels 'brushed' by the pen as it traces the line, and filling the result with the pen colour. Line-drawing operations include straight lines, arcs of circles and a general form of wavy line called a *Bézier curve*. There are also assorted convenience routines including arrow heads, rectangles and rectangles with rounded corners.

The final group deals with text. Text is placed on the screen by specifying the character string to be printed, the pixel position of the first character and the *font* which specifies the character shapes for each code. Every character code has a corresponding bitmap (sometimes called *glyphs*), and the text primitives copy the character bitmaps into the target raster, computing the start fo each character based on the width to the preceding characters. This allows for proportionally spaced text in various sizes, but the problems of rotating bitmaps mean that such fonts have a fixed orientation (usually going across the screen).

There is some discussion of QuickDraw on the *comp.sys.mac* newsgroup.

Presentation Manager

Presentation Manager is part of the OS/2 scheme of things and again has little direct competition in the OS/2 market. Microsoft offer a look-alike for MS-DOS systems, called Windows 2.03, and the current litigation from Apple over the "look and feel" of Windows 2.03 is clearly intended to affect Presentation Manager as well (Windows 2.03 is sold as looking identical to Presentation Manager as far as the user is concerned): suing

Microsoft is perhaps a way of getting a useful precedent without taking on IBM immediately. Regardless of these legal troubles (which really can't succeed in the long run, not against the IBM money mountain), Presentation Manager is a fairly typical example of how a window system works on a multiprocess operating system.

Each application is written using a large library of graphics routines for handling line drawing, multifont text and so on, and also using the *window system toolkit*, another large library of routines which provide the now-familiar buttons, menus, scroll bars, etc used by graphical user interfaces. The toolkit in particular uses object-oriented programming techniques: all buttons are very similar and share the same code but applied to a separate data structure for each button. The *instance data* for a given button will include things like the label written on the button (e.g., 'Save File') and a pointer to the *callback routine* to be executed whenever the button is pressed. The typical structure of a program has an initialisation section in which all the necessary buttons etc are created and placed onto a single window belonging to the application, followed by an *event loop* which involves waiting for something to happen (e.g., a key press, a mouse motion, perhaps an input from some external sensor in a process control environment) and then responding to that *event* by calling the appropriate routines. The toolkit routines will normally be handled directly within the library, though Presentation Manager itself doesn't work like this. This sort of program is difficult to analyse because it has a beginning, a middle and lots of other bits, and the flow of control cannot be determined without knowing exactly how the toolkit works. Tools such as lint, adb and dbx won't be much help because so much of the object-oriented machinery involves pointers to functions and the state of the program resides much more in the values stored in all those small data structures than in the current stack backtrace. The closest equivalent in UNIX experience is trying to debug the UNIX kernel, though window systems are usually slightly easier.

The graphics and toolkit libraries interact with a central part of the window system which is normally embedded in the operating system kernel: this deals with the handling of mouse and keyboard interrupts, passing them as events to the application programs, and with managing the screen. Some systems run outside the kernel, in which case the kernel must supply fairly direct

nuisance of different source code for different vendors hardware: the vendor now supplies an X server and the existing program still works.

The distributed nature of X offers further advantages, especially to multi-user systems such as UNIX. Manipulating large graphics displays is a processor-intensive activity and makes big performance demands on memory (a Sun 3/50 runs benchmarks 20% faster if the screen is disabled!); this makes it impractical to have lots of graphics displays on even the fastest of minicomputers. However, with a distributed window system, you can add graphics displays simply by buying small computers to run the graphics server, and running the applications on the multi-user computer as before. We are beginning to see companies producing 'X terminals', simple diskless graphics workstations whose 'operating system' is X and which connect to Ethernet rather than RS232 ports. The cost of these is currently around \$2000 US, which is cheaper than most 'add-on graphics displays', and cheaper still are X servers based around small micro computers such as the Atari-ST (of which more in a future issue). At the other end of the scale, Cray Research is a member of the X Consortium (which now controls changes to the X system) because systems like X offer the only rational way to add interactive graphics to supercomputers.

The advantages of X are many; it provides manufacturer independence, it simplifies the production of graphics software over a wide range of machines, and it raises the capabilities of many existing systems. The drawbacks are mostly to do with its flexibility. You are expected to make your own policy decisions and so you are faced with a choice of toolkits, a choice of window managers etc, all of which make life harder: "I wanted an answer, but you have given me just a new set of questions".

X is already widely available: it is available on Hewlett-Packard machines under HP-UX, it will be available on all DEC machines shortly (under MS-DOS, VMS and Ultrix), it will be available on Sun systems shortly (and SunView2 will be built on top of X) and is available from Apollo and Apple (for A/UX). The free source code mentioned earlier is available from MIT for the cost of the magnetic tapes, but is also available from various sites in Europe.

One of these is from Jamie Watson. He offers a distribution on QIC-24 (Sun, IBM RT/PC, Arix,

NCR others) or TK-50/TK-70 (DEC MicroVax and Vaxstation) cartridges. He makes the distributions free; all you have to do is agree to send back his tape, or an equivalent one. By the way, someone sent back a tape *FREIGHT COLLECT*. Pretty tight. Please don't do that.

The tapes that he sends contain the following:

- The complete MIT X.V11R3 distribution, including core, and user contributed 1 & 2.
- All 'official' patches issued by MIT up to the day that he makes the tape.
- The Purdue Speedups.
- Everything that has been posted to comp.sources.x

Contact him at:

Jamie Watson
Adasoft AG
Nesselerweg 104
CH-3084 Wabern
Switzerland

EUNET: *jw@pan.uucp*
Tel: +41 31 54 35 70
Fax: +41 65 42 10 71

The difference between the sample server on the tape and the one you buy from Hewlett-Packard (for example), is that Hewlett-Packard have put effort into tuning their version to run as fast as possible on their hardware; the sample server is intended to be easy to port so that you get something working, and doesn't claim to be particularly fast. There are some very active USENET newsgroups discussing X, in particular comp.windows.x and comp.sources.x, plus there are other mailing lists on specialised sub-topics.

Finally, X has no serious competition. Both the Open Software Foundation and UNIX International are supporting X, it is the window system recommended by X/Open, and best of all, none of these groups has control of X so it will remain an open standard.

NeWS

NeWS (the Network extensible Window System) was developed by Sun Microsystems and was viewed as direct competition for X. It is now more clearly understood to be different from X and represents one possible direction for future window systems.

NeWS is a device-independent distributed window system, but it carries device independence much further than X by using the PostScript language developed for use with laser printers. In systems like X, everything is described in terms of the pixels on the screen, so applications are not independent of things like the shape of a pixel (not all dots are square?!). NeWS however takes its descriptions in absolute terms using real number coordinates and real number proportions of red, green and blue for its colours; you can draw anything in any colour at any size. The NeWS graphics server then has the task of reproducing the ideal description as best it can with the hardware available, and it can use any and every piece of trick hardware available. This makes the NeWS server 'future-proof' because it can incorporate new developments in graphics hardware without changes to the application programs, and also make it a much better environment for applications which produce on-screen an approximation to the output you will produce on a printer. All of this is much harder work computationally, so a NeWS server is likely to be more expensive than an X server and might run slower.

NeWS has facility which gives a definite advantage over X: the language used by clients to control the server is an interpreted programming language rather than a sophisticated but fixed set of commands, so subroutines can be downloaded to the server and avoid some of the overheads of network communication between the server and client. NeWS extends the PostScript language to add lightweight processes, so it is possible to create processes which run entirely within the NeWS server (for instance, the notorious 'eyeball' demonstration where a cartoon-style eye watches the mouse as it moves around the screen). This sort of technique allows parts of the toolkit to live in the graphics server, which makes the response time of pop-up menus much better than in a system such as X, where everything is delayed by the round trip times of the network communication.

Despite its advanced features, NeWS is not currently strong competition for X because it is about two years younger and is not so widely available. There is also resistance to NeWS because it doesn't allow the traditional hands-on access to specialised hardware and so cannot be made to perform some of the tricks of high-speed animated graphics. Faced with this problem, Sun have produced a solution which combines all of

both worlds (not just the best!): a combined X and NeWS server, called X11/NeWS. The underlying 'pixel shuffling' is common to both servers, so it is relatively easy to build a server which works as both an X server and a NeWS server on the same screen with the same mouse and keyboard: problems only arise in areas of conflict such as control of the screen background, and in dealing with how the two systems should be visible to one another. Sun claim to have tackled these problems and will shortly be Beta-testing the combined server, which will be the X server available for Sun machines, and which will be part of the standard distribution of AT&T System V release 4. This will enable NeWS to filter out into the marketplace inside the X server, where it will be waiting as applications come along which require its special capabilities.

At present NeWS is available without the X server on Suns (naturally) and a number of other machines including the Macintosh II under A/UX and some specialised graphics hardware from Parallax and Silicon Graphics. Sun's source code is reasonably easy to port, and Sun themselves did a 'reference port' to the DEC VaxStation II under Ultrix, and to an unnamed 80386-based machine running System V.3; both of these ports come as part of the NeWS source code. They also did a 'joke port' to the Atari ST; the ST uses the same 68020 processor as the Sun 3, and apparently one Atari owner at Sun noticed that the Sun load file ('dot-o') format could be mangled into something that the Atari loader could read.... The Atari ST port indicates a possible further development in distributed window systems, the use of a graphics-only server which is connected simply (perhaps even via a modem) to another machine which deals with the local area network connections and anything which doesn't directly affect the pixels on the screen.

Newsgroups relevant to NeWS include *comp.windows.news* and *comp.lang.postscript*.

Glossary

The above discussion has tried to introduce the jargon of window systems at the same time as discussing the important systems currently around. You'll have to refer back to find them, but the important words are:

pixel, raster, bitmap, RasterOp, source, target, tiling, mask, font, glyph, event, event loop, toolkit, callback routine, graphics server, abstract graphics device.

There are no good general books on this subject (though I'm helping to write one!), but there are reasonable books on the Macintosh system and probably Presentation Manager (there was a good book by Petzold on Microsoft Windows, the ancestor of Presentation Manager). The computing press have details of courses on specific window systems and/or overviews, though for proprietary systems such as NeWS it is easiest to approach the manufacturer for information on courses.

Conclusions

The above mixture of survey and introduction should have given you some idea about what is going on in window systems at present. This is currently an active area, and one in which UNIX is very much involved with window system standards being pulled along by the UNIX standards activity. Fortunately there is an open window system (X) which has been gathering momentum in recent years and is well placed and well suited to being the standard window system for UNIX; an open window system for an open operating system.

Puzzle Corner

Mick Farmer
mick@cs.bbk.ac.uk

Birkbeck College
University of London

Hello,

One day last year I casually submitted something amusing (to me :-) to Alain for inclusion in the Newsletter and now look! A regular feature.

Anyway, I hope you find these puzzles interesting and stimulating. First, the solution to Puzzle Number 1 in the last issue:

1. Thompson's book is dedicated to Mike, because the other four are accounted for.
2. Thompson's christian name cannot be Steve, Mike, or Brian.
3. Thompson's christian name cannot be Ken, because Mike Ritchie dedicated his book to Ken.
4. Thompson's christian name is therefore Dennis.
5. Bourne's christian name is therefore Brian.
6. Ken's surname is Kernighan.

Second, Puzzle Number 2 which is dead easy:

Last night I dreamt that the EUUG had overcharged some of the national groups and that

they were going to refund 1,000,000 ECUs. With no guideline available they decided that the amount received by each group was going to be some power of seven, e.g., 1, 7, 49, 343, ... They also decided that no more than six groups would receive the same amount. How did they distribute the 1,000,000 ECUs?

Third, Puzzle Number 3 which is slightly more difficult:

It was a windy day and Brian was flying his kite from the Hill using a new wire developed at the Labs. This wire had a diameter of only one-hundredth of an inch yet it was extremely light and strong. The wire had been provided in the form of perfect sphere just two feet in diameter. What is the length of the wire? You can assume that when the wire is wound up the ball is solid. Answers to the nearest mile!

For those not used to British measurement of distance: 1 foot == 12 inches, 1 mile == 5280 feet.

That's it then. Mail me if you want some clues otherwise read the solutions in the next issue.

Mick

Software | Review

Donal Daly
daly@cs.tcd.ie

*Distributed Systems Group
Department of Computer Science
Trinity College Dublin.*



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

Introduction

Welcome to *Software | Review*. This new column will review software that appears on the net. This is software that is posted to the source news groups and is in the Public Domain. In the future it may also deal with commercial software as well. For an example of what sort of software get posted to news groups, I have included a listing, at the end of this column, of all the software that has been posted to the *comp.sources.unix* group. I hope you will also find information here about soon to be released software, and what's the latest and greatest new utility is.

I welcome contributions from you. In fact I actively encourage it!! This is the reason for the pipe in the title. I will filter ('pipe') your contributions and gather them together into topics for publication. You can send contributions to me at the above email address.

Introduction over – let's get down to it! This first column includes information on a new upcoming news software releases. A short note on a new window manager package just released. A report on the newish source group *comp.sources.x* and reviews on *wanderer* (a game), *xshar* (a handy utility). We round off this month column with a

brief report on archive sites. I have 'bullied' a few friends into helping me write this month's column. You will see their names at the start of the section they wrote, everything else was by me and I accept responsibility for the article as a whole! (If you knew me this would bring a smile to your face :-)). I must thank also, our newsletter editor, Alain Williams for his gentle prodding ensuring that this column saw the light of day. Jamie Watson read an earlier version of this column and provided some useful comments.

MGR – A Window Management System

During the last 2 weeks of January and the first week of February a new window management package was posted to *comp.sources.unix* (Volume 17). The package was split into 61 parts and was the largest package ever for *comp.sources.unix*.

MGR was written by Steve Uhler at the Computer Science Research Division at Bellcore. The code itself is quite small according to an introductory news message: "For the basic window manager the MGR code is one tenth the size of the

comparable X code''. MGR is supposedly not tied to a given operating system or hardware. Has anybody used this 'beast', and would like to share their experiences?? Below is an abstract from the documentation which serves to best describe MGR.

MGR - C Language Application Interface
Stephen A. Uhler
Bell Communications Research

MGR (ManaGeR) is a window system for UNIX that currently runs on Sun Workstations. MGR manages asynchronous updates of overlapping windows and provides application support for a heterogeneous network environment, i.e., many different types of computers connected by various communications media. The application interface enables applications (called client programs) to be written in a variety of programming languages, and run on different operating systems. The client program can take full advantage of the windowing capabilities regardless of the type of connection to the workstation running MGR. This document describes the C interface library for MGR which provides a set of macros and functions that implement the stream protocol. This library provides client programs written in C with a function call interface to MGR. The library provides the lowest level access to MGR functions and represents a one to one mapping to the underlying stream protocol.

ShareWare Software

The concept of Sharware software, is where you normally receive the package for free and then if decide you like it and want to use it you pay a fee. This places the trust on the user to register his package. Jamie Watson informed me about a specific Shareware package which he now uses, and is quite impressed with, and I thought you might like to hear about it as well. The package in question is *Jetroff* which was posted to *comp.sources.misc* a few months ago. Complete sources were posted as an incentive. This text processing package had a number of good features, including complete *ditroff* functionality, and the ability to handle bit-image files created by various PC drawing utilities.

To register your copy would cost you \$50 as an individual, a \$100 for a company. This seems a reasonable price when you compare it to a package like XROFF which costs

Like Jamie, I feel that something of this quality and usefulness deserves a little encouragement. If you have had any good/bad experiences with Shareware software let me know.

WANDERER

(Hugh Grant <hugh@maths.tcd.ie>)

This game, billed as a "mini rogue-like adventure game" first appeared back in issues 2 and 3 of volume 5 of *comp.sources.games*. I didn't look at the game then, as I was under pressure to do some work of an academic nature, and to stop "playing stupid computer games". However, when version 2 of the game appeared recently, curiosity won, and the 's' key was pressed and the source lodged in a secure place. It was only when Donal Daly started badgering me to review some software from the net that I actually got around to compiling and playing the game and I must say I was quite surprised at what it turned out to be. The title does not accurately describe the game, it being neither an adventure, or for that matter like rogue.

The game is by Stephen Shipway from Warwick University. It comes in two shell archives, which unpacked without any difficulty. There are *makefiles* for UNIX and MS-DOS, though I have only tried it on UNIX. I just typed 'make' and the game compiled perfectly first time. It's nice to see this, as it gave me hope for the next stage: actually running the thing. 'make install' copied everything to the right place, and so I decided to sit back and have a go at playing the thing. Here one problem cropped up: the high score table was incorrectly created as 'hiscores' by the *makefile*, and the game looks for a file called 'hiscore', but this was the only problem I ran into. I'd imagine the game will run on almost anything, as I tried it on fairly minimal hardware (a PDP-11/73 running 2.10BSD).

The game is quite fun to play, although it was a bit confusing at first, until I figured out exactly what I was supposed to do! (Yes, I really should have read the *README* files first, I know, I know). The basic idea is that the player has to move around the screen collecting treasure without being killed by falling rocks, arrows, bombs, or on later screens, monsters. There are some other goodies such as teleport traps, and slides for the rocks to bounce down. This is definitely not a hack 'n slay game, as each screen must be completed carefully, or one may make

some treasure inaccessible, hidden under a pile of fallen rocks. The game uses *curses* to show a window onto a larger screen that scrolls automatically as the player moves, although a facility to zoom out and see the whole arena is available. The game also allows the editing of new screens, and though I tried this, I always seemed to make the screens far too easy or totally impossible.

The game is good fun, and makes a change from the usual *nethack/rogue/moria* sort of game. The scrolling window is a nice touch and is probably as good as it gets on an ASCII terminal (though I'm not sure it would be much fun at 1200 baud). A manual page would have been nice, for although nearly everything is in the *README* files, I got the impression that one was assumed to have played version 1 of the game. All in all a worthwhile addition to any UNIX games collection, and full marks to Stephen Shipway.

XSHAR

(Hugh Grant <hugh@maths.tcd.ie>)

This posting, a replacement for the usual *shar* and *unshar* programs, appeared in *comp.sources.misc* in early May. It was labelled as 'shar2' in the headers, but is in fact called 'xshar' in the documentation and *makefile*, so I'll refer to it as that from now on. It really only caught my notice recently when I had to transfer a large number of files to a new VAX which had not yet got any file transfer protocol up and running, so I decided to mail the files to my account. Our existing *shar* program was fine as far as it went, but it had no option for setting file sizes, or for automatically dealing with binary files. Hazily remembering this article, I scurried back to my 'News' directory and was overjoyed to see it had all the features I needed! Whew! I was afraid I'd have to send around five megabytes of data doing the encoding and splitting by hand!

Xshar was written by Bill Davidsen in the States, and was posted on the 10th of May, 1988. The programs come bundled as one *shar* file which contains the source and documentation for *xshar* and *unshar* as well as a *makefile*. The documentation is fairly complete, consisting of two manual pages and a *README* file. Everything compiled without a hitch and seems to work very well.

Xshar seems to have all the bells and whistles that anybody could ever want. It has an option for

either treating everything as a plain text file, or as a binary file, *uuencoding* it before writing it to the output file. It also offers a mode where *xshar* decides for itself whether a file is text or binary and acts accordingly. This option uses 'file' to inspect each input file, so it should work fairly reliably, at the cost of forking an extra shell each time, but then that's a minimal cost compared to *uuencode* anyway. Another option allows *xshar* to split its output into several files, each with a maximum size in kilobytes. This is ideal for mailing through systems which have a 64k limit on mail messages. *Xshar* is quite intelligent about this, splitting input files over more than one output file, unlike a similar program I used which just left files larger than the limit as they were. The output files recombine the input automatically when run.

The *unshar* program is designed for stripping off the headers in mail or news messages and passing the remainder straight through to the shell for decoding. Funnily enough, the *unshar* program that comes with *xshar* seems crude compared with the one residing normally on this system, which offers a few extra options such as specifying the output directory.

In summary, *xshar* seems to be a useful tool, especially for those involved in posting files over the net. It certainly wins over the ordinary *shar* in its detection of binary files, and its ability to split input files in an intelligent way.

comp.sources.x

(Jamie Watson <jw@pan.uucp>)

As most EUUG members are aware, interest in the X Window System has been increasing tremendously in the past year or so. Of course, as the net often reflects the interests of its readers, activity related to X has been increasing as well. Likewise in the tradition of the net, sources for X programs began to appear in various places on the net. This produced something of a problem, since there was no single place to watch for X sources. Recently, a new group was created specifically for posting of X sources. The name, obviously, is *comp.sources.x*.

This group is moderated, so the postings are strictly limited to source code for X programs, and occasional informational postings by the moderator. No discussions, flame wars, or other non-source postings are tolerated; we have *comp.windows.x* for that. The general quality of the programs posted has been very high; in fact,

the only shortcoming so far is that there have not been nearly enough games posted. Time should remedy this problem, though...

The following is a brief summary of what has been posted so far. More detailed reviews of some of these programs will be given in a future column.

- *xfig* – A MacDraw style line editor
- *awm* – The Ardent Window Manager
- *twm* – Tom's Window Manager
- *xwatch* – A replacement for the (awful) *xbiff* mail notifier
- *xipr* – A utility to dump a window to an Imagen printer
- *xmille* – The Mille Bourne game
- *xtools* – An easy way to start X programs
- *menupane* – Popup menus for the X toolkit
- *qix* – An arcade game
- *dclock* – A *real* digital clock
- *xplaces* – toolplaces for X
- *xphoon* – Show phase of the moon on the root window

Obviously, these form a very nice core of utilities and games for X. The window managers are particularly nice, since both of them do a fair number of the new window manager functions introduced with X11, such as title bars, which *uwm* does not. Also, anyone who is still suffering with *xbiff* really should get the *xwatch* program as soon as possible. The prize winner for pure aesthetic beauty in this group, though, is *xphoon*.

Archive Sites.

There are a number of archive sites around Europe. It would be useful to build up a complete picture of what archive sites there are, what they keep and how one may access them. So if you are an archive site or if you know of archive sites mail me with the details and I will publish a list in a forthcoming newsletter. For now here are details of 3.

Denmark

The Danish User Group (DKUUG) runs a mail-based service at diku. It is only available to EUnet users in Denmark because of accounting. It features access to the latest EUUG tape

distribution which includes sources from *comp.sources.unix* and *comp.sources.games*. Also some specially collected items like GNU EMACS is available. To get in contact with this archive service, do:

```
mail diku!archive
Subject: help.
```

Archive mail enjoys a 100% surcharge compared to ordinary mail.

England

In England there is an FTP- and mail-based server at the Imperial College London. It is for UK sites only. It is run by Lee McLoghlin and Stuart Mc Roberts. All volumes of *comp.sources.unix* are online, though some maybe in compressed form. For information about the mail based server mail to mail info-server@doc.ic.ac.uk, with a message body of:

```
request catalogue
topic comp.sources.unix
request end.
```

They also store most of the GNU software, X Windows, MINIX updates, *uucp* and most other software deemed useful by the management.

Switzerland

Jamie Watson (*jw@pan.uucp*) Service Anonymous UUCP, SnailMail of tapes and diskettes. *Comp.sources.x games,misc,sources*.

NewsFlash!!

For those running the USENET Software (B News 2.11 patchlevel 14) there are two new News software packages on the way.

One Called C News by Henry Spencer, Geoff Collyer (see USENIX Winter 1987 "News Need Not be Slow", pages 181-190) and News 3.0 by Eric B. Raymond. In the next issue I will include reviews of these packages.

For those who tend to resist change, or hold News 2.11 dear in their hearts (:-), three patches have been posted recently to *news.software.b*. These patches are mainly small bug fixes, and speedups.

Next Issue

Well that's the first column over. I am interesting in comments (by e-mail) if you have found anything interesting or useful. All 'flames' passed through my special mail filter to */dev/null*, this helps keep me optimistic about life, the

universe..... Remember any contributions are gratefully received.

If you use any of the GNU software, I am interested in your experiences using them.

I will gather together responses and publish them in the next issue. Also in the next column there should be a more complete list of archive sites.

Comp.sources.unix Archive Listing

Subject: Volume 16 (Ends January 17, 1989)

index16.1	Introduction to comp.sources.unix
index16.2	List of sources in the archives
conf2	(4 parts) Multi-user conference system
index16.3	Corrections to first two INFO postings
colm	A columnation program
deliver	(3 parts) Mail delivery program
dist2	(7 parts, 2 patches) Larry Wall's Configure generator, etc.
fep	(5 parts) Front end editor program
fido	Watchdog for users and machines
hgrep	Highlighting grep filter
ida2	(8 parts) IDA Sendmail kit
identlist	List identifiers and declarations for C sources
less5	(4 parts) Less, a pager that's more than more
lintproto.pch	Turn 4.2BSD lint into a prototype generator
lumberjack	Logfile monitor tool for Suns
month8.7	(6 parts) A visual calendar and appointment system
narc	Archive net sources groups
obvious-pw	Tell if a password is "obvious"
pcomm2/part01	(8 parts) Modem communications package
plp	(16 parts) Public lineprinter spooler package
psterm	(4 parts) Terminal emulator for NeWS window system
sao	(49 parts, 2 patches) Smithsonian Astronomical Observatory
spiff	(4 parts) Spiff, find approximate differences in files
texi2roff	Convert GNU Texinfo files to nroff/troff files
uniqbib	Remove duplicates from a "bib" database
xfmt	A simple formatter

Subject: Volume 15 (Ends August 12, 1988)

index15.1	Introduction to comp.sources.unix
index15.2	List of sources in the archives
abcd	Automatic Backup Copy Daemon
arc5.21	(5 parts) ARC (PC compression program), v5.21
ck	Check mailboxes for new mail
cshar	(3 parts, 3 patches) Tools to create and unpack shell archives
cu-shell	A "shell" for CU, Kermit, etc.
cu-shell.note	Ignore the copyright on the cu-shell posting
ddd	Fast, multi-process dd(1) clone
delta-times	Delta time routines for alarm(2) manipulation
dis6502	6502 disassembler
dis88	(2 parts) Symbolic disassembler for PC/IX
dvipage	(4 parts) Sun previewer for TeX DVI files
emitc	Routine to process ctime(3) output
gbench	(2 parts) Graphics benchmark toolkit for X

hash8	Hash long identifiers into unique short ones
ien116	Updated IEN-116 namesever
inc-elim	Filter to eliminate file inclusion commands
index15.3	Update on "killer" and error in "stevie" subject line
ioccc/part07	(7 parts) International Obfuscated C Code Contest
lp-onionskin	Wrapper for System V lp, bug work-around
lwf	ASCII to PostScript filter
mcat.new	A cat(1) for mmap'able devices
monthtool	(2 parts) Monthly appointment calendar, for Suns
moontool	The moon on a Sun
mp	Mail pretty printer v1.4 (aka mail->postscript)
mush6.2.pch	Upgrade kit for Mush release 6.2
mush6.3kit	(4 parts) Mush (mail user's shell) upgrade kit, version 6.3
net-notify	Network message system, sort of like wall
newgetty	Alternate getty front-end, with speed detection
nroffgraphics	(2 parts) Tools for nroff graphics on dot-matrix printers
pages	Page accounting aide for SysVrel3.1
perl2	(15 parts) Perl, version 2
ps.sun.pch	Module to make postscript interpreter work under Suntools
rmtlib	Remote magtape library for BSD
rot	Rotate text
ru	A users(1)-style rwho
siod	Scheme in one defun
stevie	(2 parts, 1 patch)i Stevie, an "aspiring" VI clone for Unix...
surun	Run commands as another (or super) user
touchup	(6 parts, 2 patches) A bitmap editor for Suns
tpscript	(5 parts) Ditroff to PostScript translator
twm	(4 parts) A window manager for X
ultrix-modem	UUCP/CU access on one modem
unfsd	(2 parts) User-level NFS server
uumailclean	Clean-up backlogged UUCP mail
vn.april.pch	(2 parts) VN, April, 1988, upgrade kit
vtree	Visual display of directory tree
whichtape	Tools to help find files on backup tapes
window-srch	Windowing search (not unlike context grep)
xmodem3.6	(5 parts) Xmodem release 3.6
yp-quote	Building custom Yellow Page Maps

Subject: Volume 14 (Ends May 20, 1988)

index14.1	Introduction to comp.sources.unix
index14.2	List of sources in the archives
ioccc	(5 parts) International Obfuscated C Code Contest
vplot	Device-independant graphics system, with drivers
mush6.0	(14 parts) Mail User's Shell
nntp1.5	(9 parts) Network News Transfer Protocol, version 1.5
vn.nntp.pch	VN NNTP conversion kit
mush6.0/patch1	Mush updates for SystemV, etc., Patch1
jove4.9	(21 parts) Jove, an emacs variant, version 4.9
flex	(5 parts) Flex, a lex replacement
flex/patch1	Flex, a lex replacement, Patch1
3bconnect	3B2 Ethernet Connection and File Transfer Utility
rast	Sun rasterfile I/O library

splay-tree	Splay tree library
bsd-dyna-link	Dynamic linking package for BSD
cdecl2	(2 parts) New version of Cdecl, parse C declarations
shellforms	(2 parts) Forms interface for shell scripts
sharedmem	(4 parts) Shared memory emulation for 4.2BSD
calc	A trig/multi-base calculator
pcomm	(6 parts) Dial out and terminal emulator
pcomm/patch1	Dial out and terminal emulator, Patch1

Subject: Volume 13 (Ends early March, 1988)

4.3autobaud	Baud rate detection for 4.3BSD
atl	List jobs in at queue for 4.3BSD
attpc.renice	Change process priority on ATT PC
autoadd	Program to add users to system
backups	Tools to help automate backups
bool-eval	Boolean expression array evaluator
bpatch.2	Binary file editor
bpatch2	Binary patch program, ported to 80286 etc.
budpak	Utilities to monitor usage on system
cassette-lbl	Cassette label formatting program
cfc	New version of .cf compiler
check	Check for mistakes in C programs
derez	(2 parts) Derez, remove stale files from system
e	Friendly front-end to vi
ease.pch	Patches to EASE sendmail.cf language
file	(2 parts) Replacement for the file(1) command
funcproglang	(2 parts) Functional programming language
iface	(2 parts) Generic user interface kit
korner	Convert (some) csh scripts to ksh scripts
labels	Program to make mailing labels
lit	Lit, a "better" echo
little-st2	(5 parts) New release of little smalltalk
m4	(2 parts) Public domain M4 macro processor
mcc	Merge C code with compiler error messages
measures	Brute force measurement selection
modemcap	Hardware-independant modem routines
nroff-driver	Nroff driver table utility
pas2c.pch	Patches for Pascal-to-C translator
perl	(10 parts+sample) Perl, a "replacement" for awk and sed
perl	(2 parts) Perl patches 1 through 10
printacct	Print BSD accounting file
process-uucp	Tools for pathalias with MMDF
pwget	Programs to retrieve /etc/passwd info
ratfor	Public domain RATFOR in C
rf	Rolodex-like filing system
rolodex	(4 parts) Screen-oriented rolodex program
rpc3.9	(15 parts) Sun RPC, release 3.9
rstat	Remote statistics server
sbbs	(2 parts) A BBS written in /bin/sh
sc5.1	(3 parts) SC spreadsheet program, version 5.1
sets	Perform "set" operations on command line arguments
slice	Split file based on patterns or line numbers

starchart	(2 parts) Starchart package patches
ups	BSD File delivery programs
vn.jan.88	(5 parts) VN newsreader, 1/88 version
vt220fontedit	Font edit program for VT220 terminals
with	Resource allocation program
xmodem	(3 parts) Full featured xmodem program, v3.4

Subject: Volume 12 (Ends February, 1988)

afio	(2 parts) Manipulate CPIO-format archive and files
cake	(9 parts) Cake, a make replacement
cnews	(14 parts) C News alpha release
crc.pch	CRC Graphics Package Patch#1
fuser	Who's using that file? (For Unix-PC)
hershtools	(5 parts) Hershey font manipulation tools and data
index12.1	Introduction to comp.sources.unix
index12.2	List of sources in the archives
ln03-plot	New version of LN03 plot(3) package
ops5	(5 parts) OPS5 in Common Lisp
musbus5.2	(3 parts) Monash benchmark update
pathalias9	(2 parts) Pathalias, version 9
pdtar	(3 parts) Public domain TAR
postscript	(18 parts) A PostScript interpreter
qterm.alt	Query terminal for its type
starcharts	(7 parts) StarChart program and Yale star data
strings.coff	Find printable strings in COFF files
vmail	(3 parts) vmail - screen-based mail handler
zmodem	(3 parts) Zmodem file transfer programs

Subject: Volume 11 (August 11, 1987 to October 6, 1987)

3bnet	(2 parts) 3Bnet utilities and printer spooler
avl-subs	AVL Tree subroutines
bsd.2.10.note	BSD2.10 available from Usenix
bsmtp	Batch SMTP program
bundle	Buffered copy to/from physical devices
comobj.pch	Patch for Common objects sources
cpmod	Copy modes/ownerships/times
getty-enable	Getty on/off programs for 4.[23] BSD
graphedit	(2 parts) Graphcs editor for Suns
hum.pch	Hum concordance package update kit
id	(3 parts) C cross-reference database system
inline	(4 parts) Inline code expander for C
inline/patch1	Inline code expander for C, Patch1
jove.pch	(4 parts) Jove upgrade kit
jove.pch/patch1	Missing file from Jove update, Patch1
lemming	(2 parts) Update kit for lemming editor
less3	(3 parts) The 'less' pager
little-st	(3 parts) Little Smalltalk interpreter
musbus	(4 parts) MUSBUS 5.0 -- Monash University Benchmark
monthtool	Sunview visual calendar
mtools	(2 parts) MS-DOS disk tools for Unix
mush5.7	(12 parts) Mail user's shell

netdata	Transfer data (and mail) between SYSV and CMS
number	Arabic numerals to multi-lingual natural language
psfig	(5 parts) Including PostScript figures in ditroff
qsubst	A query-replace program
reader.poll	Poll on copyrights
saver	Small SUN screen-saver
sc4.1	(3 parts) Spread sheet program, sc 4.1
se.pch.2	Second update for 'SE' editor
smail3	(3 parts) Smail, UUCP domain mailer
syslog	Development version of syslog(3), for ATT, too
syslog.sysv	SystemV version of syslog
tcsh.4.3	(2 parts) Tcsh for 4.3 CSH
tcsh	(6 parts) New version of T-shell
tek2ps	Tektronix4014 to PostScript filter
templates	(6 parts) Template-mode for GNU Emacs
test.el	(3 parts) Test system for GNU Emacs
vitals	Word counts, checksums, etc.
watcher	(2 parts) Watcher system monitor program
zoo	(7 parts) File archiver programme

Subject: Volume 10 (June 17, 1987 to August 10, 1987)

agef	Show disk usage by file age
cbar	Another changebar program
cbw	(5 parts) Crypt Breaker's Workbench
cfc	"Compile" sendmail.cf files into EASE language
comobj.lisp	(13 parts) Common Objects, Common Loops, Common Lisp
complex-lib	Complex arithmetic library
copytape	Copytape, a magtape xerox (tm) machine
copytape2	NEW version of magtape copy program
crc_plot	(6 parts) CRC Plotting Package
derez	Find and remove stale files from a disk
des	DES encryption routines and a login front-end
dev.fd	A /dev/fd device driver for 4.3 and NFS systems
ease	(4 parts) Ease translator repost
fastgrep	(3 parts) Reposting of world's fastest grep
hum	(3 parts) Bull Tuthill's "hum" text concordance package
ida	(7 parts) the IDA sendmail kit
ien116-server	IEN116 Nameserver
ifp	(7 parts) Interpreted Functional Programming lanuage
lc	An "ls" program
lemming	(4 parts) A graphics editor
logo	(6 parts) Logo interpreter for Unix
magtapetools	(2 parts) Magtape handling package
mx-macros	REPOST of Troff macros for "ACM Transactions"
notes-mod.pch	Patches for NOTESFILES for moderated groups
nrchbar	A "changebar" interface for *roff
ptoc	(12 parts) Pascal to C translator
qterm	Query terminal for its type
regexp.pch	Bug-fix for regexp() library
screen	(2 parts) BSD multi-screen manager
sps	(3 parts) SPS for BSD, Ultrix1.2, Sun3.x, NFS
sxt-sh-jobs	(2 parts) Diffs for SystemV /bin/sh job control with sxt's

top_s375	(2 parts) Top users display, 2.1 with Symmetric changes
tr2latex	Translate troff to LaTex
x10r4.sunpch	(3 parts) X10R4 patches for Sun3/110C

Subject: Volume 9 (March 3, 1987 to June 16, 1987 <great renaming>)

assem2	(2 parts) Generic assembler for micro's
bitstring	"Bitstring" package
elm2	(19 parts) ELM Mail System
fastgrep	(2 parts) Fastest grep around
gwyn-dir-lib	New directory-access library
index9.1	Introduction to mod.sources
index9.2	Index of mod.sources archives
index9.4	Change in archive sites, recent errors
localtime	Public Domain (Table Driven) "localtime" patch
month	(2 parts) REPOST of Visual calendar program
mx-macros	Troff macros for "ACM Transactions"
old.bad.code	Previous "obfuscated C" winners
printf	Printf(1), for shell scripts
teco	(4 parts) A TECO text editor
uemacs3.8b	(14 parts) MicroEMACS, version 3.8b
uumail.pch	UUMail 4.X patch
xscreen	(2 parts) Screensaver for X window system
xterm	(7 parts) Terminal emulator for X window system
zmac	(2 parts) Z80 macro cross-assembler

Subject: Volume 8 (January 26, 1987 to March 3, 1987)

ansitape	(2 parts) ANSI tape program
cut+paste	Public-domain implementations of cut(1) and paste(1)
dca2troff	Convert IBM DCA documents to troff input
display	Execute command repeatedly, display output
ease	(4 parts) Ease, a language for writing sendmail.cf files
fixcpio	Repair damaged "cpio -c" archives
foogol	A (vax) compiler for a tiny ALGOL-like language
getpw	Public-domain getpw*(3) routines
graph+	(3 parts) A Graph Plotting Program
her2vfnt	Hershey fonts to 'vfnt' rasterizer
hier	Directory hierarchy scanner
index.1	Accessing the archives
index.2	Index of volumes one to seven
jove	(13 parts) The JOVE text editor
kurses	A program to call curses(3) functions
mcp	(8 parts) Account creation/manipulation program
micrognu	(11 parts) A Micro-Emacs variant that resembles GNU Emacs
multi_seed.c++	Simultaneous multi-site news feeder in C++
multivol.pch	Multivol, Patch #1 (see Volume 7)
pd-localtime	(3 parts) Public Domain (Table Driven) "localtime"
phoon	Phase of the moon, date routines
prep	(2 parts) A pre-processor for FORTRAN source
psfig-tex	(3 parts) Including PostScript/Mac figures in TeX documents
qterm	Query terminal for its type
se	(7 parts + 1 Patch) Georgia Tech 'se' screen editor

shrink_names	Shrink VeryLong+File.names to shorter names
smail2	(5 parts) Smail (UUCP domain mailer), release 2.3
soelim	A .so/.nx/.PS filter for *roff files
sp	(2 parts) Soundex spelling-checker
tabs	A tab/space conversion program
textool2	(2 parts) A collection of tools for TeX users
trmatch	Syntax-checker for *roff
uk-1.4.pch	Patch for UK-1.4 mail configuration
unaxcess2	(4 parts) UNaXcess Conferencing, version 1.00.02
uucp.x25pad	UUCP X.25 'f' protocol and PAD dialer
uumail4	(4 parts) Uumail release 4.2
uutty	Bidirectional getty/login for System V
vn	(3 parts) The VN news reader
vtrm	(2 parts) A Unix/PC virtual terminal package

Subject: Volume 7 (Ends January 20, 1987)

2.11news	(20 parts) 2.11 News Release
4.3cpp.patch	#elif patch to 4.3BSD cpp
aaakeys	Ann Arbor XL key uploader
append	Allow additions to 'protected' directories
basic	(6 parts) A BASIC Interpreter
bpatch	Binary (file) patcher/viewer
cmstape	Read and write IBM VM/SP CMS dump tapes
csh.patch	Two CSH patches
des	Purported DES program in C
determcap	Decomposing termcaps
dirstack.csh	CSH tools for directory stacks
elm_update	(3 parts) ELM Update Kit
forktest	Find security holes in shell-escapes
getmetrics	PostScript program to generate .afm files
getoptprog	Getopt program for scripts
hostup	An alternative to the BSD ruptime command
idle.users	A simple BSD idle-users daemon
image	(5 parts) Image manipulation routines in C++
index.1	Index and Archives
index.2	Complete Listing of Mod.Sources Archive
index.3	Archive access and listing
index.4	Index for Volume 7 and other info
less3	(3 parts) New release of LESS
make	Public-domain MAKE
micro.asm	(2 parts) Generic assembler for micros
msdos_mk.patch	Patch to msdos_mk for Microsoft C
multivol	(2 parts) Multivol V1.00 - multivolume backup utility
nag	(2 parts) Nag reminder service
new_archives	Additional UUCP Access to Mod.Sources
patch2	(3 parts) Release 2.0 of patch
paths.mk	Makefile to build UUCP paths
pdtar	Public-domain TAR program
read-vms-backs	Read VMS backup tapes
regex	Ed(1)/regex(3)-compatible reg. exp. package
remtape	Remote magtape library for 4.3BSD
rvi	(4 parts) Vi front-end for remote editing

safe	Limit a program's execution time
smail	(2 parts) Domain mailer and rmail replacement
sop	A .so filter for n/t/*roff files
sunmailwatch	A mail watcher for SUNwindows
tar_aids	Tools to read damaged tar tapes (tar_aids)
texdvi2tty	TeX DVI driver for TTY's, etc.
textools	(2 parts) A collection of tools for TeX users
tinytcp	A tiny set of TCP routines (tinytcp)
top2	(2 parts) Top users display for 4.2BSD, Version 2.0
tput	Public-domain tput(1) program
tput2	Public-domain TPUT (corrected implementation)
untamo2	Log out idle users
untamo3	Log out idle users (untamo revised)
uucp+nuz.tulz	Erik Fair's UUCP & Usenet toolbox
uuencode	Uuencode and uudecode
vms_tools	(2 parts) Unix-like tools for VMS systems
vttest	(2 parts) Test VT100 Features
xlisp.patch	Patch to Xlisp1.6 for Pyramid machines
xmodem	(2 parts) Full-featured XMODEM
yacc.notes:	Tools to restart YACC parses
yacchacks	Tools to restart YACC parses
yearlength	Compute length of any year

Subject: Volume 6 (Ends mid-July, 1986)

intro	Introduction to mod.sources
untamo	Untamo, another idle daemon
calls.new	New calls; shows function call flow
vol	Create volume headers for tar
makekits2	Makekits revisited
maildigest	Mail digest utilities
gr_scripts	Shell Scripts for game regulator
pacman.p	Apollo Pacman-like game
datediffs	patches for date to use elsie!ado's localtime
getpaths	Tools for analyzing netnews paths
sysVtalkA	A talk for system V.2
sysVtalkB	A talk for System V
texdvi2lj	(3 Parts) TeX DVI driver for LaserJet+
halign	Halign - line up columns
context	Context - generalized context printer
pacman.p.h	Missing files from Apollo pacman
less.patch	Patches for more/less interoperability
qterm	Query Terminal for terminal type
printfck2	New printfck and manpage
context.1	Manual page for context program
compress.xenix	Xenix patches to compress4.0
fmtr.patch	Patches to fmtr
unrm.rm	Rm and unrm programs
elm	(14 Parts) Elm mail system
cvs	(2 Parts) CVS, an RCS front-end
ditrev	Page reverser for ditroff
stringlib	X3J11/SVID/4BSD/etc string library
cpp.patch	Patches to 4.2BSD cpp for #elif, // comments

help	(2 Parts) Help programs
glob	'Globbing' library routine
cdecl	English<->C translator for C declarations
sh.ulimit	Add ksh-style 'ulimit' to 4.2BSD /bin/sh
bsearchstr	Binary search for strings in a file
yyref	Cross-reference for Yacc
newbatchA	Usenet news batcher control program
newbatchB	Usenet news batcher control program
malloc	A "smarter" malloc
S3uuque	Uuque for System III/V in C
lbl	Lbl preprocessor for [nt*]roff
malloc.mk	Missing makefile for "malloc" posting
elm/Patches1	Elm fixes for BSD, et. al.
Misc.Patches1	Changes to calls, compress, ditrev, getpaths, nbatcher
vt100tool	(10 Parts) VT100TOOL for Sun's
settz.patch	Updates to "settz" data files
uEmacs3.7	(12 Parts) MicroEmacs, Version 3.7
bsd.ps.patch	Speed, etc., patches for BSD ps
watch	A multiple "tail -f" program
reminders	A Personal Reminder system
sysVdial	(3 Parts) System V generic dial routines
rpc2	(11 Parts) Sun RPC Source
malloc.patch	Bug fix for "smarter malloc"
newsctn	Count unread news articles
less2	(2 Parts) New version of less
msdos_mk	A Make for MS-DOS and VAX/VMS
att_which	A "which" for non-BSD systems
lj_filter	Filter for HP Laserjet
xlisp1.6	(6 Parts) Xlisp version 1.6

Subject: Volume 5 (Ends late May, 1986)

uEmacs30fix	MicroEMACS version 30 updates.
uumap	Automated UUCP maps
dither	Color Dither (ver 1.1)
retouch	Retouch(1): force changed date
backup	Front end for BSD dump
junkmail	Delete outdated mail automatically
smallc	(3 parts) Small C compiler version C3.0R1.1
moon_sun	Sun and Moon rise/set program
par	More patches to par/unpar
smtp_send	SMTP SEND command for Sendmail
bmgsubs	Boyer-Moore-Gosper fast search subroutines
untic	Decompile terminfo description file.
bmfix	Fix to B/M/G for odd address optimization
rcsit	Prepare files for RCS (new version)

Subject: Volume 4 (Ends early May, 1986)

bm1.2	Bm version 1.2 (blindingly fast "fgrep")
simplex	Simplex Curve Fitting Algorithm in C
chuni	Change a user's default universe (Pyramid Specific)
Msg	(8 parts) Screen-oriented "User Agent" mail program

sim2	Update to "sim" (volume 3) similarity tester
shortc	C program to map flexnames into short identifiers
setz	Time conversion / time zone system
TVX	(10 parts) Portable editor, with "emacs" and "vi" modes
hershey.f77	(2 parts) Hershey Fonts in Fortran 77
rolodex	(3 parts) Rolodex database program
68kdisassem	(2 parts) 68000 disassembler
bm1.2speedup	Speedup for bm on some machines
regexp3	2nd bug fix for regexp (volume 3)
tm_to_time	Convert broken-down time into time_t.
68kdisss.fix	Patches to make MC68000 disassembler work on SUN UNIX
amiga	Amiga file browser
rcsit	New version of rcsit(1) - prepare files for RCS
hershey	(5 parts) Hershey fonts
egrep	More Pep for Boyer-Moore Grep
tc	Compile/decompile nroff driver tables (USG only)
regexpfix	Regexp(3) improvement
shortc	Shortc: sed output, and standard input
match1.2	Fast grep for Vaxen
rlogin	4.2bsd rlogin enhancements
list	List-of-numbers generator
client	Generic client and server commands for 4.2BSD
client_man	Client/server context diffs to 4.2BSD man.c
UK-1.4	(5 parts) Sendmail UK-1.4
ISO_Pascal	Yacc and Lex for ISO Level 0 Pascal
TVX	1st batch of TVX Bug fixes
rpt	A program called 'rpt'
subnetARP	4.3BSD IP subnet ARP hack
UNaXcess	(3 parts) UNaXcess (unix bulletin board)
uEmacs	(6 parts) MicroEmacs, v. 30
travel	Travel-itinerary macros for nroff
aaa	The amazing awk assembler
sources	Two tools for organizing sources from USENET
load	Routines to check the load average
uEmacs_tc	Termcap support for MicroEmacs v. 30 sources
archx	Archx: suggested replacement for shar
ar	Portable ar: suggested replacement for shar
se	(8 parts) Georgia Tech 'se' screen editor
telnetd	Telnetd in the kernel
uumail3	(2 parts) Uumail 3.0
lplot	(2 parts) Lplot and quickplot
mail	Patches to BSD4.2 mail (SysV mailx?)
sticky	PostScript sticky label program
uEmacs3.6	(8 parts) MicroEMACS 3.6
texindex	Make an index from a LaTeX .idx file
chown	Improved and expanded chown/chgrp
calendar	(2 parts) Calendar generation program
strings	(3 parts) String routines
gr	A Game Regulator
printfck	Have lint check (most) printf calls
unparfix	Unpar compatibility with Sys V (patch)
texindex2	AAAAARRRRGGGGHHHH!!!! Bugs in texindex!!!

UnaXcessfix	UNaXcess update #1
xmodem	4.2BSD XMODEM programs
icon	Tools for editing Sun icons
fmtr	Simple text formatter

Subject: Volume 3 (Ends February, 1986)

G-format	(4 parts) VAX BSD4.2 compiler modifications to use G-format fp.
GaTech	(3 parts) Sendmail patches/configuration files from Georgia Tech
GaTech.upd	Updates to GaTech sendmail package
Hey	Hey(1) [from Unix/World, Oct. 85]
LA50	Convert Nroff underlines to LA50 and VTxxx sequences
LaserJet	(2 parts) Ditroff HP LaserJet driver
MSdir	MSDOS directory access routine
RFS	(7 parts) Public domain, kernel-resident distributed file system
SPS	(3 parts) Show process status - BSD only - replacement for "ps"
TCToTI	Termcap to terminfo conversion program
TRC	(8 parts) Expert system building tool
agelog	Trim log files while retaining recent entries
att_getopt	AT&T's public domain distribution of getopt(3)
badm	BSD4.2 MASSBUS disk formatter utility
bm	Ken Yap's changes to bm (in volume 2)
calendar	A calendar generator program - replaces UNIX "cal"
calls	C program function call cross referencer
calls_4.2	Patches to calls for BSD4.2
chsh	Chsh,chfn for SV (password file programs)
chsh2	Chsh,chfn - Original contained security bugs.
clr.queue	Script to clean-up the sendmail queue
command	Replacement for system(3).
ctags	Ctags source code from Ken Arnold
date	Formatted date program
decus_grep	Public domain version of grep.
dial	State transition controlled communications program
dial.sample	Example dial script.
dialout	BSD4.3 Kernel changes for dial in/out on modem lines
dtree	Directory heirarchy display program for 4.2
ff	(2 parts) Simple text formatter for flexible uniform formatting
give	Give away ownership of files (System III/V specific)
hdifff	Source file compare program
head	Public implementations of head(1) and ctags(1)
help	VMS-style help facility
hyphen	Program to enhance troff's hyphenation capability
idledaemon	Yet another idle login checker (BSD 4.2 only)
ieee	(6 parts) IEEE Floating Point Calculator (in Pascal)
infer	Inference engine + demo
laserjet	BSD 4.2+ lpd printcap/spooler for LaserJet printer
lcat	Troff->laserjet filter package (uses vfont files)
lcat2	Troff->HP Laserjet filter - newfonts.c
less	Similar to more(1) but better
lib_term	Datum entry using termcap
libc_term	Datum entry using curses
llib-dbm	Lint library for the DBM routines (BSD systems)
man	Compiled version of the 'man' program for System V

match	Faster than bm (VAX only!)
mdump2	Revised mdump, the multiple dump per tape utility
modgen	Extract usenet moderator list from postings
modnotes	Notes (1.7 or later) updates for moderated groups
modula_pp	Pretty printer for Modula-2 written in Modula-2
newspace	Determine newsgroup disk usage
nwho	Enhanced "who" program (uses termcap)
okstate	Kermit archive on OKSTATE; uucp access information
pathalias2	(2 parts) Pathalias, the mod.map database path optimizer
pretty	Pretty printer in lisp + columnator in CLU
prune	Prune tops of line-oriented log files
rcsit	A program to prepare files for RCS.
regexp	Regular expression routines (like System V regexp(3))
regexp2	Bug in regexp, and fix
rename	A companion to restor (automated inode mapping)
rmsecure	Source for a safe "rm" (csh, BSD only)
rsend	BSD network communications program (like write & talk)
scpp	(2 parts) A selective C preprocessor - clean up your C files.
sim	Software similarity tester for C programs
sndml.mods	Mods to sendmail to provide translation tables
suntools	Improved version of Sun's window manager (suntools)
swho	Screen based who (uses curses - continuous update)
tc	Control your terminal via termcap in shell scripts
telno	Permute telephone numbers into letter equivalents
texchk	(2 parts) Syntax checker for the LaTeX TeX macro package.
times.awk	Uucp info from LOGFILE (awk script)
ttype	Typing tutor - BSD specific
ttyuse	Creates a Summary of daily Terminal usage
turbo_patch	Fix to turbo_tools, SHELL.PAS transmitted with error
turbo_tools	(2 parts) Turbo Pascal version of "Software Tools in Pascal"
uuhosts4	Grab mod.map data for later use version1.69
uumail2	Pathalias-based uucp mailer, release 2
uumail2.fix	Small fix to uumail release 2
vtem	A VT100 emulator based on termcap
wm.new	Window manager built on top of curses
xargs	Execute a command with many arguments

Subject: Volume 2 (End roughly August, 1985)

Smail1	Update to smail (in volume 1)
access	Kernal Hacks for access control lists
basic	(4 parts) A BASIC interpreter in C (needs work)
bgrep	Boyer-Moore based fgrep like program
bm	Much faster Boyer-Moore
bm2	Various bm updates
choose	A program to select lines at random
compress	(2 parts) Compression 4.0 program better than pack or compact
cshar3	Update to C shar (volume 1)
cpg+mdep3	Cpg revisited (C formatter - original in volume1)
makekits	Software "kit" generation script
mdump	Multiple dump per tape utility (see update in volume 3)
remote	Remote mag tape routines
remote2	Small patch to remote tape library

rtar	Diffs to tar to use a remote system's tape drive
runtime	Runtime memory allocation for multi-dimensional arrays
tools	(6 parts) Software Tools in Pascal
uroff	Nroff underlining
window	(4 parts) BSD 4.2 window manager + Patches to Curses
wire	(2 parts) Wirewrap program.

Subject: Volume 1 (Ends June 1985)

ANSI.C	Yacc and Lex for 11/12/84 draft of ANSI C
Smail	A smart net mailer - a front end using pathalias data
UK-1.1	(3 parts) UK-1.1 Sendmail Configuration Package
Xlisp1.4	(4 parts) Lisp written in C with object oriented extensions
bed	Editor for binary files. Front end for ascii editors
bourne	(9 parts) Bourne shell enhancements (history,tilde,job control)
cforth	(3 parts) Forth Interpreter written in C
checkin	Editor interface for RCS logs
cpg+mdep	Cpg - C formatter, mdep - make dependency generator
cpp	(3 parts) C preprocessor suitable for use with Decus C
cshar	Shell archive builder (shar) written in C
cxref	C cross referencer
diffc	Contextual diff (diff -c) for Bell systems
dynamic	Dynamic loading code for 4.2bsd
getopt	Public domain getopt(3)
lbgm	Newsgroup archiving (Little Bird Gave Me)
newshar	The Connoisseur's Shar, version 2
newsweed	A program to delete unwanted news articles
patch	A program to apply diff format output to update files (1.3)
pcurses	(11 parts) Public domain Terminfo/Curses (needs a little work)
rfc_882	RFC 882 - Domain Names - Concepts and Facilities
m	(9 parts) Rn news reading program, version 4.3
rpc	(10 parts) Sun "Remote Procedure Call" source code
sendmail.cf	GaTech Sendmail configuration
uucpanz.V7	A uucp status program (V7, BSD version)
uucpanz.S5	Uucpanz for System V
uuque	A uuwizard's utility for uucp queue snooping
vnews	(7 parts) New reading program for 2.10.2 news
vstr	Dynamic string package
xfernews	Uucp traffic batching system
xref	A general purpose cross reference utility
vnews.1	Manual page for 2.10.2 vnews(1)
readnews.1	Manual page for 2.10.2 readnews(1)
expire.8	Manual page for 2.10.2 expire(8)

USENIX Association News for EUUG Members

Donnalyn Frey
donnalyn@uunet.UU.NET

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.



1989 Winter USENIX Association Conference

The USENIX Association's 1989 Winter Conference Conference, held January 30 - February 3 in San Diego, California, was a success. Over 2,000 people attended the conference. The first two days were devoted to tutorials, with the next three days for technical sessions. William T. O'Shea, Vice-President of Product Development at AT&T, presented the Keynote Address.

Tutorials were 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.

Technical papers were presented on distributed systems, file systems, operating systems, window systems, internetworking, objects and memory, processes, security, and two special interest sessions. A Work in Progress session was held on Thursday afternoon to preview new work not yet ready for publication.

Viruses, Worms, and Other UNIX Pests

A special session entitled "Worms, Viruses, and Other UNIX Pests" was held at the conference. The session focused on recent UNIX worm and virus papers and included discussion of the recent problems. The session contained a paper by Donn Seely entitled "A Tour of the Worm" and a paper by Eugene Spafford on "Some Musings on Ethics and Computer Break-ins".

Earlier in the conference, Tom Duff of AT&T had presented a paper entitled "Viral Attacks on UNIX System Security", accepted as a conference paper long before the recent worm attacks.

Open Software Foundation & UNIX International

The USENIX Association hosted information sessions by both the Open Software Foundation and UNIX International, Inc., formerly the Archer Group, at the conference. The OSF session was held Tuesday evening, January 31 and the UNIX International session was held on Thursday evening, February 2.

This was the second time the OSF held an information session with USENIX Association conference attendees. The OSF panel included Alex Morrow, director of strategic planning, Ira Goldstein, director of the research institute, Ellis

Cohen, member of the User Environment Component team, and Eric Shienbrood, operating systems project leader. They provided conference attendees with an OSF technology update, including the initial User Environment Component offering, an overview of the selection process, a review of the delivery schedule and development period evaluation, and a summary of the licensing options.

This was the first time UNIX International held an open meeting with UNIX users. The UNIX International panel included senior technical representatives from UNIX International and member companies. They discussed the role, objectives, and purpose of the new organization in guiding UNIX development at AT&T, how members can participate in the process, and answered questions from conference attendees.

To request copies of the Winter 1989 USENIX Association Conference Proceedings, send email to *{uunet,ucbvax}!usenix!office*.

1989 Summer USENIX Conference and Exhibition

The 1989 Summer USENIX Conference and Exhibition will be held in Baltimore, Maryland on June 12 – 16, 1989. The first two days will be devoted to tutorials, with the next three days for technical sessions. For further information on the conference, contact the USENIX conference office.

Workshop on Software Management

The USENIX Association will be holding a Workshop on Software Management on April 3 – 4, 1989 at the New Orleans Hilton and Towers in New Orleans, Louisiana.

The workshop will be concerned with the management and processing of source; the discipline of managing, maintaining, and distributing software. The workshop is planning presentations and discussions of software management including release engineering, configuration management, installation tools and techniques, construction tools and techniques, source code control systems, and testing.

For further information on attending this workshop, contact the USENIX Conference Office.

Workshop on UNIX Transaction Processing

The USENIX Association will be holding its first Workshop on UNIX Transaction Processing on May 1 – 2, 1989 at the Pittsburgh Hilton Hotel in Pittsburgh, Pennsylvania.

The workshop plans to include papers and discussions on:

- Transaction Integrity
- Two-Phase Commit
- Distributed Transactions
- Client-Server Transaction Models
- Transaction Queuing and Scheduling
- Data Entry Systems
- Transaction Benchmarking
- Transaction System Performance Modeling
- Operating System Support for Transaction Systems

For further information on attending this conference, contact the USENIX Conference Office.

Distributed Processing Workshop and Graphics Workshop

The USENIX Association will be holding a Distributed Processing Workshop in Fort Lauderdale, Florida October 5 – 6, 1989.

The fifth Graphics Workshop will be held in November or December of 1989. For information on the call for papers for these workshops, contact the USENIX Association Office at P.O. Box 2299, Berkeley, CA 94710, at +1 415 528 8649, or by email at *{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, as well as the new workshops on Software Management and Transaction Processing. The office can also provide information on the annual C conference and the semi-annual technical conferences.

UNIX Clinic

*Colston Sanger
colston@olibc1.oliv.co.uk*

Olivetti International Education Centre



Colston Sanger is a lecturer at the Olivetti International Education Centre, Haslemere, UK and a visiting lecturer in the Faculty of Engineering, Science and Mathematics at Middlesex Polytechnic. In his spare time he is an art historian and is the organiser of the J.L.Agassee exhibition which has just opened at the Tate Gallery, London.

No space on integral hard disk drive 0, partition 0

You're out of space. What's more, the message on the console says you're out of space on the root partition. What now?

First of all, don't panic: nothing is going to crash. But you are going to have to find some space before you'll be able to do anything sensible with your system. If you can, check for and delete any junk files (*core*, for example) in the root directory. Otherwise, try zeroing out */etc/wtmp* and */etc/utmp*. If all else fails, on an Olivetti-AT&T 3B2, you could try removing */unix* (it will be rebuilt automatically when you reboot the system).

Once you've got yourself some breathing space, it's time to analyse what happened and to ensure that it doesn't happen again. The best way, of course, is to prevent it altogether. As a well-seasoned system manager, you are supposed to know pretty much from hour to hour how much free space you have (the *df* command, remember?) and, really, the root partition shouldn't change very much at all. Apart from */etc/utmp* and */etc/wtmp* (which log user and accounting information for commands such as *login* and *who*) nothing much else changes – assuming that you have a separate */tmp* partition. (*vi*, *cc* and a few other commands write

their temporary files in */tmp*,† so it's certainly a good idea to have a separate partition mounted as */tmp*: 10–20 MB should be ample.)

By the same token, it's probably best to keep your user directories out of */usr* – otherwise you'll be running out of space there as well. Instead, have separate partitions for each group. On this system, for example, users' login directories are in */staff* for lecturing staff, */centre* for the office ladies and */course* for course delegates (spread over three machines connected with *RFS*, but the principle holds).

The complete picture is then:

```
/  
/tmp  
/usr  
/staff  
/centre  
/course
```

† Properly speaking, they reference the environmental variable *TMPDIR* – by default */tmp* in System V Release 2, */usr/tmp* in Release 3.

It's also tidier, in that there is a clear separation between the standard UNIX System distribution and user files. Backup is easier too, because we can backup a whole partition and know that we have captured everything – without the sort of 'pick and mix' that would be involved if user

```
# root crontab
#
# cleanups
# clear /etc/wtmp every morning at 7 am
0 7 * * * cp /dev/null /etc/wtmp ; \
chown adm /etc/wtmp ; chgrp adm /etc/wtmp
#
# cut down /usr/lib/cron/log at 4 am, Mon, Wed, Sat
0 4 * * 1,3,6 tail -50 /usr/lib/cron/log > /tmp/cronlog ; \
cp /tmp/cronlog /usr/lib/cron/log; rm -f /tmp/cronlog
#
# cut down /usr/adm/sulog at 4 am, Wed, Sat
0 4 * * 3,6 tail -50 /usr/adm/sulog > /tmp/sulog ; \
cp /tmp/sulog /usr/adm/sulog ; rm -f /tmp/sulog
#
# get rid of any core-dumps, dead.letters at 6 am, Mon-Fri
0 6 * * 1-5 /bin/find / -name core -o -name dead.letter ) \
-mtime +3 -print 2> /dev/null | xargs /bin/rm -f
#
# remove any leftovers in /tmp at 7 am, Mon-Fri
0 7 * * 1-5 rm -f /tmp/[Qq]* /tmp/*.dlf \
/tmp/calendar.* 1>&2 > /dev/null
#
# clear out everything once a week, at 6 am, Sat
0 6 * * 6 rm -f /tmp/* /usr/tmp/* 1>&2 > /dev/null
# get rid of preserved (but unused and forgotten)
# vi edit buffers
0 6 * * 6 rm -rf /usr/preserve/* 1>&2 > /dev/null
#
# Look through for old stuff and mail polite messages
# to users at 7 am, Sat
# 0 7 * * 6 /staff/colston/admin/old_files 1>&2 > /dev/null
```

It's probably worth going through this in some detail. The first thing I do is make sure that various log files are cut back periodically, otherwise they just grow and grow. */etc/wtmp* is a data file, so it is zeroed out after the system accounting has finished processing. On the other hand, */usr/lib/cron/log* (the log of all commands run by *cron*) and */usr/adm/sulog* (the log of all successful or unsuccessful attempts to run the *su* command) are text files: they are simply *tail-ed* – but not until some other reporting scripts have checked them for anything suspicious.

Everyday, I search the whole filesystem for *core*

directories were scattered all over */usr*.

What else can you do in the way of preventive system management to keep track of free space? Well, on this system, I do a lot of silent tidying-up with *cron*. Here are the relevant entries from the root *crontab*:

and *dead.letter* files older than three days and quietly remove them. The rationale here is that any coredumps or undeliverable mail that are useful are likely to be current. Programmers, for example, may be debugging with *sdb* and a current coredump, but anything else is junk and has been forgotten. I also remove any leftovers in */tmp*. (We run some third-party software and locally-developed programs that are not quite as punctilious about cleaning up as perhaps they should be.)

Then, once a week, I clear out everything in */tmp* and */usr/tmp*, and also get rid of unused and forgotten *vi* edit buffers saved under

/usr/preserve.

One other thing: when space is very short (or when I want to be especially objectionable) I run the script `old_files`. As you can see, it's commented out at the moment, but what it does is look through the user directories for old files with names like `temp` or `mailmerge.out`, then mails polite reminder messages to their owners asking that the files be removed. Anyway, here's the script:

```
# old_files
#
# Script to find old core, etc, files and send
# mail to their owners. The file old_ls will be
# the full list of old and bogus files.
#
# Adapted from a script posted to USENET,
# but I forget who originally wrote it.

# setup
PATH=/bin:/usr/bin
HOST='uname'
FILESYSTEMS='/centre /staff'
ADMIN=colston
ADMIN_NAME='Colston Sanger'
trap "rm -f ${TMPDIR}/old_ls ${TMPDIR}/old_unknown ; \
exit" 0 2 3 15

cd ${TMPDIR}
rm -f ${TMPDIR}/old_ls ${TMPDIR}/old_unknown

# How old files should be before we complain.
AGE=10

# You might want to tweak the predicates in here.
# Typically, whatever names people use for junk files
# on your system.
find ${FILESYSTEMS} -type f \
  \(-name a.out \
  -o -name '*', -o -name '.', '*' \
  -o -name '*~*' -o -name '.~*' \
  -o -name '*.out*' \
  -o -name '*.old*' -o -name '*.bak*' \
  -o -name 'temp*' -o -name 'junk*' \
  -o -name fred -o -name jim \) \
  -mtime +${AGE} -print | \
  xargs /bin/ls -l >old_ls

# Assuming a standard 'ls', third field is the users.
# Loop over list of all users. The grep commands ignore
# filenames; be careful if your 'ls -l' uses tabs.
#
for USER in `awk '{print $3}' < old_ls | sort -u`
do
```

```

case ${USER} in
root|bin)      continue ;; # system logins - do nothing
helmut|recept) continue ;; # Don't bother these people
[0-9]*)       grep " ${USER} " old_ls >> old_unknown ;;
*)           (
    echo "You have the following mouldy \
old files on ${HOST}:\n"
    grep " ${USER} " old_ls
    echo "\nIf you have finished with them, \
it would be nice\nto remove them, freeing up the disk space.\n"
Many thanks,\n
${ADMIN_NAME}
Your friendly system manager\n"
) | \
mailx -s "Old files on ${HOST}" ${USER} ;;
esac
done

if [ -s old_unknown ]
then
(
echo "Found unknown old files on ${HOST}:"
cat old_unknown
) | \
mailx -s "${HOST}: No owner for these ancient files." ${ADMIN}
fi

```

I also have something here called *diskhog*, written by Dave Settle and, I believe, posted to the net sometime last year. I haven't actually installed it – it seems a little drastic. It's based (rather loosely) on the disk quota mechanism of Berkeley UNIX, but uses *du* to count the number of blocks used by each person. Here is an extract from the README:

I've found this very useful on my system, where it encourages people to keep their disk usage down, without all the hassle of having to complain to them personally. The package runs a check each night on each user, and sends mail to those anti-social people exceeding their allowance.

If, after a suitable number of warnings, these people do not remove their extra files, their next login receives a restricted shell, where the only available commands are those for removing and backing up files.

You can decide on the range of commands available, and place these in the directories */diskhog* and */usr/diskhog*, so the restricted shell can be as nasty as you like. (You could be really vicious and just have *rm*, but it's probably advisable to have some commands to save to tape and/or floppies.)

Anybody who wants it is welcome to contact me.

A system manager's toolkit

Once you have disk usage firmly under control, you're ready for the next step: keeping track of what your users are up to, and how well your system is coping with the load being put upon it. After all, your job is to provide your users with a computing service. If that service isn't available when 'they' need it – because one of them is making unreasonable demands on it – they will complain.

There are, to my mind, five commands that form a system manager's basic toolkit on a stand-alone UNIX system: *who*, *who -u*, */etc/whodo*, *ps -ef* and *sar*. If you get into the habit of running them periodically, throughout the day, you will be able to anticipate and, maybe, head off many problems before they actually occur. What sort of problems? The sort that arise as questions:

Question: *Why is fred logged in when he is supposed to be on holiday?*

Answer: Possible security breach – someone else knows fred's password?

Question: *Why is alice logged in at six terminals?*

- Answer:** Good question – why indeed?
- Question:** Why is mutt still logged in – he left early today?
- Answer:** The silly person probably forgot to log off.
- Question:** Why has jeff got six vi sessions running?
- Answer:** Because he's been doing shell escapes from within vi and has probably forgotten where he is – may even be editing the same file over again!
- Question:** Why is the system so slow today?
- Answer:** Because carol is doing a 'super-make' of some new statistical software, richard is compiling a Fortran-77 image-processing program, william is troff-ing, sam is running a huge lisp object, henry is playing rogue, there is a C programming course going on downstairs, the office ladies are running the Informix database – and, frankly, how much more do you think you can squeeze out of a 4MB system? (Definitely losing cool here!)

Ahem... sorry about that.

Finally

Finally, and as an addendum to last issue's column on common (but tricky to solve) problems, one other: the case of the file with the impossible name.

```
/*
 *      fix.c - fix filenames containing funny characters
 *      From: UNIX/WORLD, February 1987
 *
 *      Usage: Do an ls -li to find out the inode,
 *      then fix [ inode ]
 */

#include <stdio.h>
#include <sys/types.h>
#include <sys/dir.h>
#include <ctype.h>

#define FIXOUT "fix.out"      /* output filename */

main(argc, argv)
int argc;
char *argv[];
{
    struct directdbuf;
```

It happens occasionally – particularly with software that puts the terminal into 'raw' mode, such as the infamous Whizzo-Office+ – that users create files with 'impossible' names, i.e., containing backspaces or other control characters. What can be done about it?

It depends. If all you want to do is remove the file (because it's junk anyway), then the easiest way is:

```
# cd to the directory containing the file
cd /staff/mutt
# If you have the -b or -q options
# to ls that display non-graphic characters
ls -b
# Be careful!
rm -i *
```

Alternatively, you could try:

```
cd /staff/mutt
# Find out the i-number of the file
ls -li
find . -inum NNN -exec rm {} \;
```

But what if the file contains valuable information that the user wants to keep? Nothing special: just a variation on the *find* example above:

```
cd /staff/mutt
# Find out the i-number of the file
ls -li
find . -inum NNN -exec mv {} new_filename \;
```

Or, if you really want, here's a C program to do the same thing:

```
int fd, fix_ino;

if (argc != 2 || !isdigit(argv[1][0])) {
    fprintf(stderr, "Usage: %s inode-number\n", argv[0]);
    exit(1);
}

fix_ino = atoi(argv[1]);
if ((fd = open(".", 0)) < 0) {
    fprintf(stderr, "Can't read current directory.\n");
    exit(2);
}
while (read(fd, (char *)&dbuf, sizeof(dbuf)) > 0) {
    if (dbuf.d_ino != fix_ino)
        continue;
    unlink(FIXOUT);           /* delete former fix */
    link(dbuf.d_name, FIXOUT); /* new link */
    unlink(dbuf.d_name);      /* delete old link */
    printf("Fixed it: inode-number = %d\n", fix_ino);
    printf("old name was: %s\n", dbuf.d_name);
    printf("new name is: %s\n", FIXOUT);
    exit(0);
}
close(fd);
fprintf(stderr, "Can't find it.\n");
exit(3);
}
```

Next issue

That's it. In the next issue I'm planning a complete change of subject: *regular expressions*, hopefully with lots of practical examples.

If anybody has any *awk*, *grep* or *sed* scripts they're particularly proud of, I'd be interested to see them. You know the address.

EUnet

Peter Houlder
uknet@ukc.ac.uk

Computing Laboratory, University of Kent



Peter Houlder has been in the Computing Laboratory at the University of Kent for the last 4 years and looked after day to day UKnet admin work in the last 3 years.

He graduated in Geography from Kings College, London in 1970 and then spent 9 years in business – dropping out in 1979. He then spent a year touring North, Central, South, and Caribbean America, became interested in archaeology and spent three years excavating in Britain and Europe.

Two Masters degrees, the first in Archaeological Sciences and the second in Computer Science, followed in successive years. Maggie in the meantime reduced archaeological funding, so he arrived in 1984 kicking and screaming into the world of Computing. He has since got to quite enjoy it.

He is married with two labradors.

Introduction

As I write it is three days since I received the dreaded "... Your contribution for the EUnet column is required by ..." The problem is nobody

had sent me anything to include. Since I started this column (7 issues ago) the EUUG newsletter has had articles on the network from the following countries:

aut		fra	Vol 7, No 4 1987	nor	
bel	Vol8, No 4 1988	gbr	Vol 7, No 2 1987	prt	
che		grc		swe	Vol 7, No 3 1987
deu	Vol 7, No 3 1987 Vol 8, No 4 1988	irl		yug	
dmk		isl		eunet	Vol 8, No 1 1988 Vol 8, No 3 1988
esp		ita		nld	Vol 7, No 3 1987
fin	Vol 8, No 2 1988	lux			

I realise that other articles have been placed prior to this column, but some up to date information, as a paragraph or whole article on any networking issue would be nice. Austria, Denmark, Ireland, Italy and Norway all have established networks with many sites, so information would be appreciated (to email address as above). It would also be nice to hear from the other countries about problems of setting up networks, future plans requests for help or any other topics. (End of Advert.)

The information below is strictly applicable to UK only users as our information servers are not authorised to send international mail, but I thought if I explained our on-line services in the UK, that it will hopefully inspire other countries to write about similar services in their countries. (Second End of Advert.)

On-Line Information Within UKnet

We now have three methods by which you can get automatic systems to send you information in the mail from UKC's machine.

Mailing information@ukc

The first method of using the information system is really designed to send simple files It is mostly used to store fundamental information about the network. To obtain an index of all the available information in the system UK users type:

```
mail information@ukc
Subject: index
```

The index is simply a file and all the other entries are obtained in a similar fashion by supplying a particular Subject line. We also put certain special information in these files like the ARPA INTERNET worm reports and on-line forms to join both the UKUUG and the UKnet network.

Mailing info-server@ukc

The second method is more complicated but is more flexible. This is the information server. The info-server provides a way for users to obtain public domain software and other information from the UKC gateway. To access it:

```
mail info-server@ukc
```

This information server will mail the files in response to a request contained in the mail message that you have sent it. Requests are of the form:

```
request: subject
topic: topic within that subject
request: end
```

As an example suppose you want to be mailed information about 'mailing-lists' in the subject 'uknet'. You would send a message of the form:

```
request: uknet
topic: mailing-lists
request: end
```

and the mailing-lists information would be mailed back to you. The 'uknet' request points at the same information used to run the information system. A list of the 'top-level' requests can be obtained by sending the following request to the info-server:

```
request: index
topic: index
request: end
```

Within a request subject, an index and some help information are both available. These would be (using catalogue as the subject example):

```
request: catalogue
topic: index      (or help)
request: end
```

All blank lines are ignored, and the 'request end' is optional, however if it is omitted and there are other lines in the message an automatic error message will be sent to you.

The UKC info-server contains all the files which can be returned by the information system, several public domain systems including the news code and *comp.sources.unix*, the news-group used to transmit useful UNIX sources.

Please note if the 'Subject:' line begins with 'Request', it is expected to hold all the information, using ';' to separate logical lines. Thus the following will have the same effect as the previous example:

```
mail info-server@ukc
Subject: request: catalogue;
topic: index; request: end
```

The current list of 'Requests' that can be requested from the UKC info-server are:

index	This index
catalogue	Information on who has what and where in fixed format.
sources	Public domain shared software system (the actual files)
uknet	UKnet information files
comp.sources.unix	Copies of the UNIX source archive (from volume 8)
siteinfo	Access the UUCP maps on a site basis.
uucpmap	Access to the World UUCP map files

Finally it should be emphasised that UKC is only one information service. It is linked to the information servers at The Universities of Cambridge, Nottingham, Glasgow and Imperial College London. If a user requests the catalogue index (as in one of the examples above) the user can see what information is held by other sites. The user can then mail that information service directly or remail UKC, who will automatically forward the request. All the info-servers can

perform this automatic rerouting.

Mailing netdir@ukc

The final information service is a simplistic method of providing a USENET directory. It is used to obtain information about a particular USENET site. To obtain a map of a particular site:

mail netdir@ukc
Subject: ariadne

Back will come:

UUCP map data for site ariadne:

System name	:	ariadne
System type	:	
Organization	:	Cretan Research Center
Contact person	:	Kostas Vassilakis
Electronic Address	:	ariadne!kostas
Telephone	:	+30 81 221171
Postal Address	:	P.O. Box 527, Iraklion, Crete, Greece
Longitude/Latitude	:	25 09 E / 35 20 N
Remarks	:	
Usenet (news) links	:	
Last editor & date	:	mcvax!piet 841215, 870609

Connect list etc. :

ariadne ermhs (HOURLY*4), theseas (DAILY)

UNIX is Chauvinistic

*Dominic Dunlop
domo@sphinx.co.uk*



Dominic Dunlop, surprisingly described by the London Times as a "software wizard", was co-founder of Sphinx Ltd., a company which distributes application software for the UNIX operating system throughout Europe.

Have you got your CD-ROM copy of the *Oxford English Dictionary* yet? Well, neither have I, so I still have to reach for the shelf, biceps tense, to pull down the first volume of the *Shorter Oxford*. What does it tell me?

¶ **Chauvin** [Fr.; from Nicholas *Chauvin* of Rochefort, a veteran soldier of the first Republic and Empire, whose demonstrative patriotism was ultimately ridiculed by his comrades.] Popularized¹ as name of a character in Cognard's vaudeville, *La Cocarde Tricolore*. 1831.

Chauvinism 1870. [-Fr. *chauvinisme* (1843), f. prec.; see -ISM.] Exaggerated and bellicose patriotism. So **Chauvinist**. **Chauvinistic** *a.*

What does this tell us (apart from the fact that I'm too lazy to work out how to persuade *troff* to print the symbols which indicate the phonetic

1. This† must be one of the words we English really are supposed to spell with a *z* – even if *Spell-b* ("OED notwithstanding") thinks otherwise.

† Although even the most unprogressive of dictionaries would seem to allow both 'popularize' and 'popularise' as legitimate British spellings, it is EUUGN's policy to use the 's' alternative wherever this is the case. Dominic has been allowed the liberty of a 'z' just to make his point.

pronunciation)? That, back in the last century, we English appropriated the name of a Frenchman whom his countrymen regarded as a joke, using it to name a concept which was just too good for the French to be allowed to keep it to themselves.

What does this have to do with the UNIX operating system? Well, it turns out that most application software is riddled with cultural chauvinism of one sort or another. For example, a program which can communicate only in English ignores the needs of those who speak other languages; a program which prefixes currency amounts with a dollar sign, and which insists that they have two decimal places, is of limited use outside the USA. If that same program is concerned with taxation, and understands sales tax, but not value added tax, the whole thing becomes about as useful to a European customer as a chocolate teapot.

How has the software industry been able to get away with such chauvinism? Firstly because many individual markets in European countries are large enough – and rich enough – to support local developers willing to produce software tailored to local needs. For example, if you are a Dane wanting an accounting package which speaks Danish, you can certainly find one. It is almost bound to cost more than a package targeted at the far larger British market, but it is available.

Similarly, you can get Danish databases, Danish word-processors, and more. You just have to put your hand rather more deeply into your pocket than you do if you are prepared put up with software which speaks English. These packages tend not to be locally-developed, but instead have been developed for another market and subsequently "localized". The localization costs money -- money which must be recovered from sales in the Danish-speaking market. You may be prepared to save money by putting up with software which is not localized.

Localization also takes time, resulting in delays in the availability of improved versions of the software. Unlocalized software is often preferred because it offers facilities that no local product can match. This is particularly true in the markets for MS-DOS and UNIX software, which are dominated by US-based software authors -- authors who were able to build sizable markets in Europe by selling products originally written for their own home market. Only when winning new customers begins to require speaking their language, does work on localization begin -- work funded out of the revenue from continuing sales of the English language version.

Of course, UNIX itself is a case in point. It is only in the last couple of years that AT&T has come up with an "internationalised" UNIX -- a version which allows localizations for particular markets to be added.

It is also only in the last couple of years that widely-accepted tools for the production of internationalised applications have appeared. Most notable among these is X3.159-198, the ANSI standard for the C language, although the IEEE's 1003.1 operating interface standard also has a short chapter devoted to the subject. Before these developments, any application which supported localization had to be written without the benefit of standards, and consequently each developer solved the problem in a different way.

Even now, while the standards cover the basic issues, they do not address important areas such as the means of access to sets of error messages in different languages. Thus, if you want to run a localized database and a localized word-processor on top of your preferred localization of UNIX today, you may have to go through three set-up procedures. Only when there is widespread agreement on -- and use of -- mechanisms for internationalisation will the need for this unnecessary duplication be a thing of the past.

The coming of 1992, with its removal of internal barriers to trade in the European Community, is exciting the interest of politicians and pundits. But if there is to be a truly open market in applications software, it has to involve programmers as well. It is programmers who will have to work out how to write software which does not suffer from chauvinism. I see late nights ahead for all of us!

Overview of UNIX System V Release 4.0

*Janet Davis
janet@uel.uucp*

*UNIX Europe Limited (UEL)
International House
Ealing Broadway
London, UK*



UNIX System Evolution

The UNIX System became commercially available for the first time in 1978, with the licensing of Version 7 of the operating system.

Since then standards efforts and technology advances have resulted in many versions of the UNIX System, and brought maturity to the UNIX System industry.

Three major variants have evolved:

- Berkeley System Distribution (BSD)
- The XENIX System
- UNIX System V

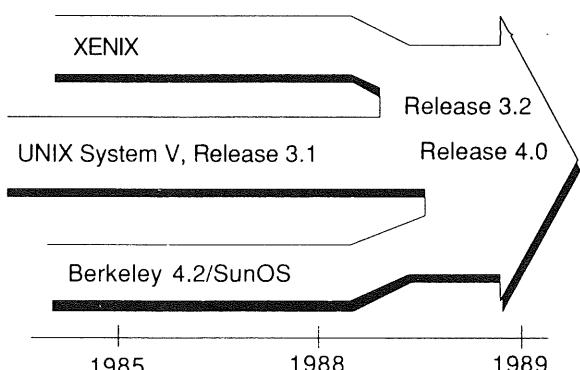


Figure 1: UNIX System V Unification

UNIX System V Release 4.0 will provide the first consolidation and implementation of these variants.

This evolution has seen a significant growth in units, revenue and software availability. For example:

- There are over 180 hardware companies and 700 software companies that sell UNIX System-based products.
- the 1987 /usr/group UNIX System products directory lists over 3164 products from 826 vendors, 170 of which are outside the United States, making the UNIX System marketplace international in scope.

Each of the major variants of the UNIX operating system has captured a significant segment of the marketplace.

BSD remained most popular among scientific and engineering users, capturing the high-end technical workstation segment of the market. XENIX systems dominate the so-called 'low-end' segment of the marketplace and desk-top systems. UNIX System V represents the business and commercial market that consists of multi-user medium-sized computers and large mainframes. These market segments are converging, forcing the UNIX System variants to converge as well.

Technical workstations have found their way into business and financial departments; the power of microprocessors has increased to such a degree that personal computers now rival the technical workstations; and multi-user systems have become servers in networks for personal computers and workstations.

Technology has forged a single market out of traditionally distinct market segments, and the market demands a single, unified UNIX System to serve all its needs.

SVR4.0 General Operating System Services

UNIX System V Release 4.0 developments for General Operating System Services (see figure 2) concentrate on the following:

- file-system enhancements to support SunOS, BSD and new types of file-system:
 - memory-mapped files
 - fast file-system
 - symbolic links
 - virtual file-system
- signal-handling mechanisms that POSIX adopted from BSD
- remote procedure call capabilities required to support NFS
- XENIX system-call compatibility required for application portability
- STREAMS I/O service from UNIX System V Release 3.0 to facilitate the implementation of protocol handling and networking services
- additional shell command interfaces in the form of the Korn Shell and the C Shell.
- continued work to further internationalisation:
 - collating sequences
 - conventions for date and time
 - numeric representations
 - clean-up of remaining libraries and utilities.

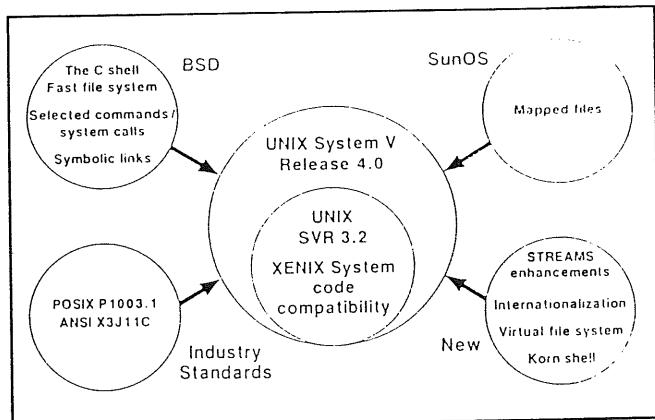


Figure 2: SVR4.0 Basic Operating System Services

SVR4.0 Application Development Environment

The UNIX System is a system created by programmers for programming, with the expectation that by creating the best possible environment in which to develop software, the best possible software will be developed.

The UNIX System is the premiere software development environment, providing a wealth of tools and utilities to make writing and testing programs easier.

UNIX System V Release 4.0 continues the UNIX System tradition of providing the foremost computing environment for the creation and use of software tools.

UNIX System V Release 4.0 expands upon the already superb UNIX System programming environment, principally in the areas of standards conformance but also in file-system enhancements (see figure 3).

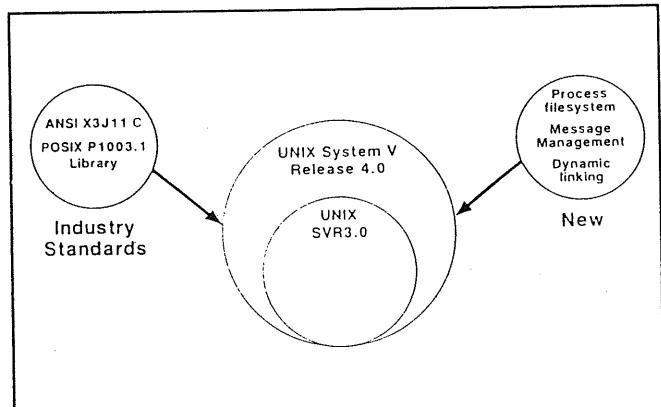


Figure 3: SVR4.0 Application Development Environment

SVR4.0 Open Systems Networking

With the introduction of STREAMS I/O in Release 3.0, UNIX System V defined the fundamental capabilities needed to support Open Systems Networking. Release 4.0 builds on the foundation laid by Release 3.0 extending the networking capabilities of the UNIX System V to include the data communication services defined by the Defence Advanced Research Projects Agency (DARPA) and supported by the 'sockets' interface of BSD.

For the first time, Release 4.0 brings together Remote File Sharing (RFS) from System V and the Network File System (NFS) from SunOS, including the capabilities for Remote Procedure Call (RPC) and eXternal Data Representation (XDR) used by NFS (see figure 4).

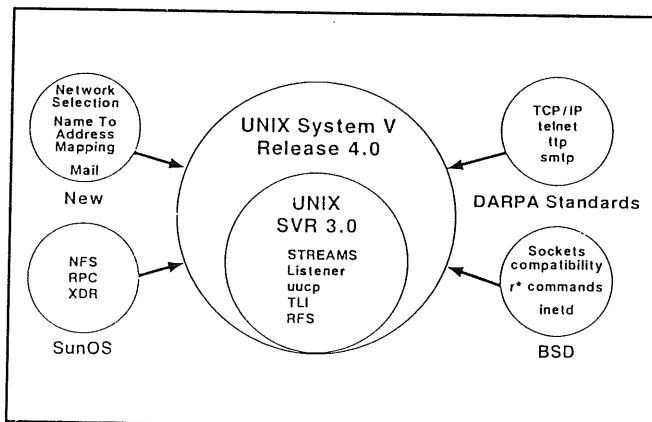


Figure 4: SVR4.0 Networking

SVR4.0 User Interface

User Interface Extensions to the UNIX System V Base provide services that simplify the development of applications that interact with users. Human-computer interface components support device-independent display management in both character-oriented and bit-mapped graphic display devices. Style guidelines and development toolkits help programmers to maintain consistency in the 'look' and 'feel' of the user-interface to promote 'user portability' between different applications that follow the same guidelines. This further reduces software costs by insulating applications from display device dependencies and reduces user training costs and loss of productivity as users move between different applications.

Software developers benefit from a rich and robust set of tools for developing applications that share the same 'look' and 'feel'. End-users

benefit because they can transfer existing skills and knowledge of how to use one application to the use of another and thereby avoid the need to learn a new and different user-interface.

UNIX System V Release 4.0 includes extensive feature/functionality in the area of user-interface (see figure 5):

- Character-oriented user interface
 - curses/terminfo
 - Extended Terminal Interface (ETI)
 - Form and Menu Language Interface (FMLI)
 - Framed Access Command Environment (FACE)
- Graphic-oriented user interface (not bundled in SVR4.0)
 - X Window System
 - OPEN LOOK

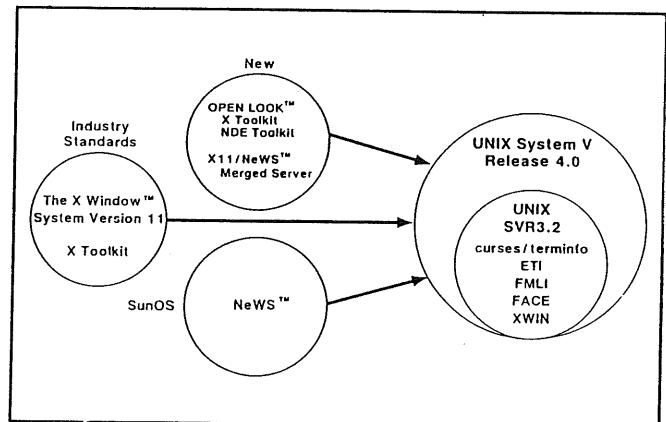


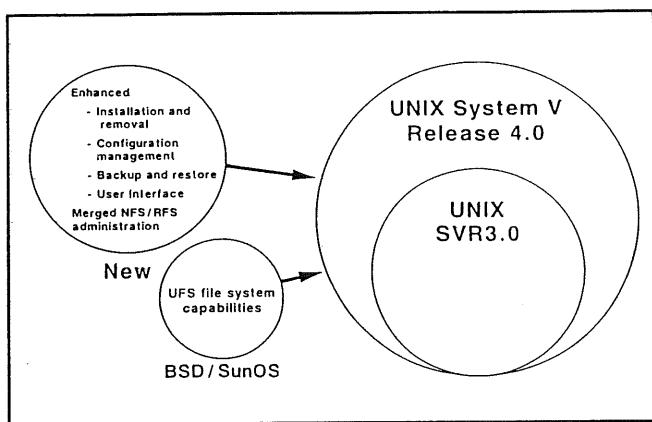
Figure 5: SVR4.0 User Interface

SVR4.0 Enhanced Administration

UNIX System V Release 4.0 further enhances UNIX System administration in the areas of:

- software distribution and installation
- configuration management
- backup and restore facilities
- message management and monitoring
- administrative user-interface based on FACE

Special attention has been given to merging the administrative capabilities of the NFS and RFS (see figure 6).

**Figure 6: SVR4.0 Enhanced Administration****SVR4.0 Application Binary Interfaces**

UNIX System V Release 4.0 takes the next step to establish a single unified UNIX software market by defining Applications Binary Interface (ABI) for computers running UNIX System V.

ABI brings to UNIX software developers the advantages enjoyed by developers of PC software, allowing applications to run interchangeably across the platforms that support the ABI as easily as software can run in the PC world today. An ABI for the UNIX System opens up the opportunity for 'shrink wrap' software with all the benefits that go with it.

Software dealers and computer stores do not stock source-code, they stock packaged software. The mass market demands programs capable of running 'out of the box', there must be a 'binary standard' for UNIX software.

The System V Application Binary Interface, or ABI, defines a system interface for compiled application programs on computer systems based on a specific microprocessor architecture.

At this time, there are plans to have an ABI specification for the following architectures:

- Intel 80X86
- Motorola 680XX and 88000
- SPARC

Other architectures will be added as we move forward.

The remainder of this column is devoted to a more technical look at ABI.

Foundations and Structure of the ABI

The System V ABI specifies two types of interfaces:

- Architecture Specific (Low-level)
- Generic (High-level)

Low-level interfaces were drawn from the specific processor reference materials and include definitions such as the processor's instruction set and register usage conventions. It also includes other definitions that are strongly affected by the characteristics of the processor's architecture. This includes information on the formats of executable and linkable files, and other implementation specific data important for application programs.

The higher-level interfaces were drawn from existing standards for operating systems, user interfaces, programming languages, and networking, including:

- The System V Interface Definition, or SVID.
- The IEEE POSIX P1003 standard operating system specification.
- The MIT X11 X Window System graphical user interface specification.
- The ANSI X3J11 C Language specification.

How to use the ABI Interface

The ABI Interface Definition is a reference document that should be used in conjunction with the publicly-available standards documents it references. The ABI enumerates the system components it includes, but descriptions of those components may be included entirely in other reference documents.

Application programmers who wish to produce binary packages that will install and run on any ABI conforming computer (for a specific architecture) should follow this procedure:

1. Write programs that use only the system commands, library functions, system calls, and other facilities explicitly included in this specification.
2. Compile the programs so that the resulting executable programs use the specified interface to all library functions and system calls, and so that the format of the executable programs conforms to the details in the specification.

3. Package the application in the format and on the media described in the specification, and install or create files only in the specified locations provided for this purpose.

The manufacturers of ABI-based computer systems who wish to provide the system interface described in the specification must satisfy a complementary set of requirements:

1. Their system must implement the specific processor architecture.
2. The system must be capable of executing compiled programs having the format described in the specification.
3. The system must provide libraries that are linked to application programs using the methods described in the specification, and contain all the library functions included in the ABI.
4. The system's map of virtual memory must conform to the requirement of the specification.
5. The system's low-level behaviour with respect to system call linkage, system traps, signals, and other such activities must conform to the formats documented in the specification.
6. The system must provide all commands, files, and utilities specified as part of the ABI, in the format defined in the ABI specification and other referenced documents. It must also provide all other components of an application's run-time environment that are included in the ABI specification.
7. The system must install packages using the formats and procedures described in the

ABI, and must be capable of accepting installable software packages, either through physical media or through a network interface.

Base and Optional Components of the ABI

The ABI provides two levels of interface specification: Base and Optional. Base components of the ABI are required to be present in all ABI-compliant systems. Optional components may not be present on an ABI-based system, but, when they are present, they must conform to the specification given in the ABI.

This distinction is necessary because some ABI capabilities depend on the presence of hardware or other facilities that may not be present, such as integral graphics displays or network connections and hardware. The absence of such facilities does not prevent an ABI-based system from complying with the ABI specification, through it may prevent applications that need these facilities from running on those systems.

Evolution of the ABI Specification

The ABI will evolve over time, and will be reissued at intervals of approximately three years. Each new edition of the specification is likely to contain extensions and additions that will increase the potential capabilities of applications that are written to conform to the ABI.

As with the System V Interface Definition, the ABI will also include the provision for Level-1 and Level-2 support for its constituent parts. Level-1 support implies that a portion of the specification will continue to be supported indefinitely, while Level-2 support means that a portion of the specification may be withdrawn or altered after the next edition of the ABI is made available.

Book Review

Lindsay F. Marshall
Lindsay.Marshall@newcastle.ac.uk

The Computing Laboratory
University of Newcastle-Upon-Tyne

Numerical Recipes in C William Press, Brian P. Flannery, Saul Teuklovsky and William T. Vetterling Cambridge University Press 1988 ISBN 0-521-35465-X (UK) Price \$27.50, Hard Back, 818 pp,

Numerical Recipes in C – diskette Cambridge University Press 1988 ISBN 0-521-35466-8 (UK) Price \$15.00 IBM PC format

Numerical Recipes Example Book (C) William T. Vetterling, Saul Teuklovsky, William Press and Brian P. Flannery Cambridge University Press 1988 ISBN 0-521-35746-2 (UK) Price \$20.00, Paper Back, 239 pp,

Numerical Recipes Examples (C) – diskette Cambridge University Press 1988 ISBN 0-521-35467-2 (UK) Price \$15.00 IBM PC format

This package from Cambridge University Press comes fairly close to meeting the requirements for the perfect recipe book – it is well written, entertaining to read and the dishes it describes are tasty. The main text covers a large amount of ground in an accessible style allowing readers to derive as much or as little information about particular topics as they require. Every section provides a list of references and further reading that can be followed by those who wish to learn more about a particular method or algorithm. Topics covered include simple Julian date manipulation, random number generation (including information on DES), sorting, and a vast range of numerical and statistical techniques. This reviewer is, to say the least, not one of nature's applied mathematicians but found the text easy to follow and has been assured by his numerical analyst colleagues that the theoretical content is of a high standard.

However, there are some drawbacks to the book. The most important of these is that it shows its origins all too clearly – this volume is a reworking of the original text which was written for Fortran users. The majority of the functions have been translated literally from one language to the other and this means lots of variables with meaningless

names like *izzy*. Only the DES functions really make use of C's more sophisticated data handling facilities though a reasonable package for complex number handling is provided. The authors also have some rather odd ideas about program structures – they dislike *continue* for some reason and suggest that one ought to use *if...else* instead of *switch* statements! They justify this later idea by invoking some supposed uncertainty about the types allowed in the control expression and by the even more dubious assertion that *if...else* is more 'recognisable' – whatever that may mean. Nevertheless these are very minor quibbles and the fact that the authors do not use the ghastly 'kernel' style for formatting their code means that they cannot be too far from the true path (at least in this reviewer's eyes). Ideally one should purchase the floppy disk (available for the IBM PC or the Mac) and so could avoid having to look at the code at all! Having the disk also eliminates the unreliable process of copying functions from the text, a task not eased by the FORTRAN-style names. The disk also includes header files with function prototypes that can be tailored for both K&R and ANSI versions of C which is very useful in this period of transition between standards. The only slightly odd thing about the IBM PC version of the disk is that the sources are held in a hidden directory which is most confusing as the disk appears at first glance to be missing all the software that one has paid for!

The companion volume contains listings and brief descriptions of some simple programs that make use of the functions provided as well as the data needed to drive them. Particularly nice to have is the listing of the NPS DES validation data which can be used to check out the encryption procedure given in the main text. However, neither this book nor the floppy that goes with it are really necessary for the programmer who just wants to get hold of a particular numerical function. The examples would be of much more use to someone teaching a course on numerical work in C – if

such a person exists - or, more likely, for someone teaching C to numerical analysts. Here again the floppy is pretty well essential as some of the datasets need to drive the sample programs are quite large (particularly the NPS data).

All in all this is an excellent package for both the programmer who only occasionally needs some numerical functions and the specialist who needs to work in C. At \$27.50 the main text is quite expensive but much better value than many other Computer Science texts in this price range.

C Programming in a UNIX Environment

Judy Kay and Bob Kummerfield
Addison-Wesley International Computer Science
Series ISBN 0-201-12912-4
(UK) Price £15.95, Paper Back, 340 pp,

The Australian UNIX tradition goes back a long way so it is not surprising that several publishers have recently set off down the Yellow Brick Road and signed up some of the Wizards of Oz. The authors of this book hail from the University of Sydney and if it is a representative sample of what we can expect from down-under then we have some treats in store. The text is easy to read and well paced for experienced programmers wishing to learn or improve their C. The examples are sensibly chosen, clearly presented using the format favoured by this reviewer and lead up to the implementation of a small but real application. This allows most aspects of developing a C program under the UNIX operating system to be covered. The authors place particular emphasis on the development of good programming style as a basis for creating reliable programs and encourage the use of C idioms to overcome the 'Pascal in C' syndrome that is often seen amongst newcomers to the language. They also stress the need to check return codes and to make use of lint to help find problems.

One of the weakest points in most books about C and UNIX is their treatment of system provided functions and libraries. The temptation to parrot the manual page is hard to avoid and most authors simply make no attempt to provide more readable information about this important area. This book is an exception, the description of the standard functions fits naturally into the style of the rest of the book and various common pitfalls are pointed out (e.g., calling *time* with a value rather than a pointer!!). The chapter on libraries takes up nearly

However when you add the cost of the floppy disk into this it does begin to look rather too dear, particularly for students. This is probably unavoidable but will tend to put off quite a few potential purchasers. If the floppies were sold at media price to those who had bought the text, or were simply included with the books themselves then sales would be much higher.

a quarter of the book and serves also to reinforce the lessons of the earlier chapters about programming style and technique.

What of the shortcomings of this book? There are a few. Firstly the book describes K&R C and only devotes four pages in an appendix to the ANSI Standardisation effort. This is written very tentatively and would not help anyone who had to get to grips with what is actually happening in the practice. The second major problem is that the book is written for someone using V7 UNIX. Whilst we all know that V7 was a much better system than most of its successors, the world has moved on and there have been a few developments that would be worth mentioning. Perhaps the authors should cash in on this and, like so many others, have a BSD version and a System V version? This would be much more useful for someone trying to take full advantage of their environment. Even if they do not do this it would certainly be worth their while to produce a version of the book for ANSI C (when it eventually settles down). It would also be worth tidying up the chapter on arrays and pointers as there are some areas of difficult that they do not touch upon. However, when all is said and done this is a good book and is reasonably priced - only three times what one would expect to pay for an equivalent sized, non-technical paperback.....

UKUUG Winter Abstracts

Here are the abstracts of the papers delivered at the UKUUG Winter meeting held at the University of Kent, England on the 19th-20th December 1988.

Copies of the proceedings are available from Owles Hall at £10 each including post and packing.

Thanks are due to Peter Houlder <uknet@ukc.ac.uk> who organised the conference and the typesetting.

A Low-Cost Bitmapped Terminal on the Atari ST

Peter Collinson

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

The development of workstations with large bitmapped screens and a pointing device has meant a revolution in computing practice. This paper describes a program written for the Atari ST micro which is an attempt to bring some of this revolution to the a wider user base by the use of low cost technology. The program, called *Term*, allows the user to control and maintain a windowing terminal. The environment may be used on BSD UNIX systems to run several shells supported by a host multiplexer. The terminal uses many features of the Atari ST, including the ability to display differently sized fonts. The paper discusses the various design issues and comments on how well the various system components perform on the low cost hardware.

Mail Interface Enhancements to MH

Donal Daly

Computing Science Dept
Trinity College
Dublin 2
Eire
daly@cs.tcd.ie

An introduction and some history and comments on the MH system, followed by a brief breakdown of its various features and some of the author's modifications and enhancements.

Developing and Adapting UNIX Tools for Workstations

David Barnes

Mark Russell

Mark Wheadon

The Computing Laboratory
The University
Canterbury, Kent CT2 7NF
England
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.

Direct Manipulation Tools for UNIX Workstations

J D Bovey, M T Russell and O Folkestad†

The University of Kent at Canterbury
jdb@ukc.ac.uk

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.

Installing and Operating NFS on a 4.2 BSD VAX

Jim Reid

Dept of Computer Science
University of Strathclyde
Richmond Street
Glasgow
G1 1XH
Scotland
jim@cs.strath.ac.uk

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

The author's experience in the installation and operating of Sun's Network File System (NFS) on a VAX 11/750 running 4.2BSD UNIX. Problems encountered during installation and operational difficulties while running the system are discussed, along with the adjustments needed to solve these problems.

A MINIX Port to the Atari ST

Aaron Gull and Sunil Das

City University
Dept of Computer Science
City University
Northampton Square
London EC1 OHB
England
aaron@cs.city.ac.uk

The presentation covers the port of the MINIX operating system to the Atari ST. The discussion will center on the novel process and memory management techniques which were employed. The impact of these upon the STIX system call interface, in particular *fork()*, *exec()* and *exit()*, concludes the discussion.

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.

UKnet Overview

Peter Houlder

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

This paper is in two parts. The first part deals with the internal aspects of UKnet as a network; what it does, how it is administered, services offered costs, etc. The second part tries to fit UKnet into the scheme of international Electronic Mail networks. The aim is to give an overview of the main world networks along

with the mail systems and transport protocols used. The relationship of these networks to the OSI 7-layer model is also discussed.

UK Coloured Books – Current Facilities and Transition Plans

Prof P.F.Linington

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

Most EUnet users think of communication with the UK academic community in terms of mail and news. These, however, form only a part of the facilities in regular use. Other protocols support file transfer, job transfer and manipulation and screen oriented interactive terminal access, and these facilities will be described. The intention is to move towards the use of international standards, and a strategy for doing so has been drawn up. This involves both technical and organisational choices which are aimed at providing continuity of service to a steadily expanding community. The future directions for this transition will be outlined.

EARN and BITNET

Paul Bryant

Rutherford Appleton Laboratory
Didcot
Oxfordshire, OX11 0QX
England
peb@ib.rl.ac.uk

The transition of EARN to use ISO protocols. EARN is currently putting in an X.25 international network as the first part of its transition to use ISO protocols. This paper describes the initial network and shows how the current services will be preserved. The outline plans for the further development of the network are described.

News – Present and Future Developments

Chris Downey

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

This paper covers the history of the USENET news system, and how it works; followed by a description of the current state of the news software, and some observations about future developments, both in the medium and the long term.

EUnet and OSI Transition Plans

Daniel Karrenberg

Centrum voor Wiskunde en Informatica
P.O. Box 4079
NL-1009 AP Amsterdam
Netherlands
djk@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.

AUUGN Back Issues

Here are the details of back issues of which we still hold copies. All prices are in Australian dollars and include surface mail within Australia. For overseas surface mail add \$2 per copy and for overseas airmail add \$10 per copy.

pre 1984	Vol 1-4	various	\$10 per copy
1984	Vol 5	Nos. 2,3,5,6 Nos. 1,4	\$10 per copy unavailable
1985	Vol 6	Nos. 2,3,4,6 No. 1	\$10 per copy unavailable
1986	Vol 7	Nos. 1,4-5,6 Nos. 2-3	\$10 per copy unavailable (Note 2-3 and 4-5 are combined issues)
1987	Vol 8	Nos. 1-2,3-4 Nos. 5,6	unavailable \$10 per copy
1988	Vol 9	Nos. 1,2,3 Nos. 4,5	\$10 per copy \$15 per copy

Please note that we do not accept purchase orders for back issues except from Institutional members. Orders enclosing payment in Australian dollars should be sent to:

AUUG Inc.
Back Issues Department
PO Box 366
Kensington NSW
Australia 2033

AUUG

Membership Categories

Once again a reminder for all “members” of AUUG to check that you are, in fact, a member, and that you still will be for the next two months.

There are 4 membership types, plus a newsletter subscription, any of which might be just right for you.

The membership categories are:

- Institutional Member
- Ordinary Member
- Student Member
- Honorary Life Member

Institutional memberships are primarily intended for university departments, companies, etc. This is a voting membership (one vote), which receives two copies of the newsletter. Institutional members can also delegate 2 representatives to attend AUUG meetings at members rates. AUUG is also keeping track of the licence status of institutional members. If, at some future date, we are able to offer a software tape distribution service, this would be available only to institutional members, whose relevant licences can be verified.

If your institution is not an institutional member, isn't it about time it became one?

Ordinary memberships are for individuals. This is also a voting membership (one vote), which receives a single copy of the newsletter. A primary difference from Institutional Membership is that the benefits of Ordinary Membership apply to the named member only. That is, only the member can obtain discounts on attendance at AUUG meetings, etc, sending a representative isn't permitted.

Are you an AUUG member?

Student Memberships are for full time students at recognised academic institutions. This is a non voting membership which receives a single copy of the newsletter. Otherwise the benefits are as for Ordinary Members.

Honorary Life Memberships are a category that isn't relevant yet. This membership you can't apply for, you must be elected to it. What's more, you must have been a member for at least 5 years before being elected. Since AUUG is only just approaching 3 years old, there is no-one eligible for this membership category yet.

Its also possible to subscribe to the newsletter without being an AUUG member. This saves you nothing financially, that is, the subscription price is the same as the membership dues. However, it might be appropriate for libraries, etc, which simply want copies of AUUGN to help fill their shelves, and have no actual interest in the

contents, or the association.

Subscriptions are also available to members who have a need for more copies of AUUGN than their membership provides.

To find out if you are currently really an AUUG member, examine the mailing label of this AUUGN. In the lower right corner you will find information about your current membership status. The first letter is your membership type code, N for regular members, S for students, and I for institutions. Then follows your membership expiration date, in the format exp=MM/YY. The remaining information is for internal use.

Check that your membership isn't about to expire (or worse, hasn't expired already). Ask your colleagues if they received this issue of AUUGN, tell them that if not, it probably means that their membership has lapsed, or perhaps, they were never a member at all! Feel free to copy the membership forms, give one to everyone that you know.

If you want to join AUUG, or renew your membership, you will find forms in this issue of AUUGN. Send the appropriate form (with remittance) to the address indicated on it, and your membership will (re-)commence.

As a service to members, AUUG has arranged to accept payments via credit card. You can use your Bankcard (within Australia only), or your Mastercard by simply completing the authorisation on the application form.

AUUG

Application for Ordinary, or Student, Membership Australian UNIX* systems Users' Group.

*UNIX is a registered trademark of AT&T in the USA and other countries

To apply for membership of the AUUG, complete this form, and return it with payment in Australian Dollars, or credit card authorisation, to:

AUUG Membership Secretary
PO Box 366
Kensington NSW 2033
Australia

- Please don't send purchase orders — perhaps your purchasing department will consider this form to be an invoice.
- Foreign applicants please send a bank draft drawn on an Australian bank, or credit card authorisation, and remember to select either surface or air mail.

I, do hereby apply for

<input type="checkbox"/> Renewal/New *	Membership of the AUUG	\$65.00
<input type="checkbox"/> Renewal/New *	Student Membership	\$40.00 (note certification on other side)
<input type="checkbox"/> International Surface Mail		\$10.00
<input type="checkbox"/> International Air Mail		\$50.00
Total remitted		AUD\$_____

(cheque, money order, credit card)

* Delete one.

I agree that this membership will be subject to the rules and by-laws of the AUUG as in force from time to time, and that this membership will run for 12 consecutive months commencing on the first day of the month following that during which this application is processed.

Date: ____ / ____ Signed: _____

Tick this box if you wish your name & address withheld from mailing lists made available to vendors.

For our mailing database - please type or print clearly:

Name: Phone: (bh)

Address: (ah)

Net Address:

.....
.....
..... Write "Unchanged" if details have not
..... altered and this is a renewal.

Please charge \$____ to my Bankcard Visa Mastercard.

Account number: _____ Expiry date: ____ / ____

Name on card: _____ Signed: _____

Office use only:

Chq: bank _____ bsb _____ - a/c _____ # _____

Date: ____ / ____ \$ CC type _____ V# _____

Who: _____ Member# _____

Student Member Certification (*to be completed by a member of the academic staff*)

I, certify that
..... (name)
is a full time student at (institution)
and is expected to graduate approximately ____ / ____ / ____.

Title: _____

Signature: _____

AUUG

Application for Institutional Membership Australian UNIX* systems Users' Group.

*UNIX is a registered trademark of AT&T in the USA and other countries.

To apply for institutional membership of the AUUG, complete this form, and return it with payment in Australian Dollars, or credit card authorisation, to:

AUUG Membership Secretary
P O Box 366
Kensington NSW 2033
Australia

- Foreign applicants please send a bank draft drawn on an Australian bank, or credit card authorisation, and remember to select either surface or air mail.

..... does hereby apply for
 New/Renewal* Institutional Membership of AUUG \$300.00
 International Surface Mail \$ 20.00
 International Air Mail \$100.00

Total remitted

AUD\$_____

(cheque, money order, credit card)

* Delete one.

I/We agree that this membership will be subject to the rules and by-laws of the AUUG as in force from time to time, and that this membership will run for 12 consecutive months commencing on the first day of the month following that during which this application is processed.

I/We understand that I/we will receive two copies of the AUUG newsletter, and may send two representatives to AUUG sponsored events at member rates, though I/we will have only one vote in AUUG elections, and other ballots as required.

Date: ____ / ____ / ____

Signed: _____

Title: _____

Tick this box if you wish your name & address withheld from mailing lists made available to vendors.

For our mailing database - please type or print clearly:

Administrative contact, and formal representative:

Name:

Phone: (bh)

Address:

..... (ah)

..... Net Address:

.....

..... Write "Unchanged" if details have not
altered and this is a renewal.

Please charge \$____ to my/our Bankcard Visa Mastercard.

Account number: _____ Expiry date: ____ / ____ .

Name on card: _____ Signed: _____

Office use only: _____ Please complete the other side.

Chq: bank _____ bsb _____ - _____ a/c _____ # _____

Date: ____ / ____ \$ CC type _____ V# _____

Who: _____ Member# _____

Please send newsletters to the following addresses:

Name: Phone: (bh)
Address: (ah)

..... Net Address:

.....

.....

.....

Name: Phone: (bh)
Address: (ah)

..... Net Address:

.....

.....

.....

Write "unchanged" if this is a renewal, and details are not to be altered.

Please indicate which Unix licences you hold, and include copies of the title and signature pages of each, if these have not been sent previously.

Note: Recent licences usually revoke earlier ones, please indicate only licences which are current, and indicate any which have been revoked since your last membership form was submitted.

Note: Most binary licensees will have a System III or System V (of one variant or another) binary licence, even if the system supplied by your vendor is based upon V7 or 4BSD. There is no such thing as a BSD binary licence, and V7 binary licences were very rare, and expensive.

- | | |
|--|--|
| <input type="checkbox"/> System V.3 source | <input type="checkbox"/> System V.3 binary |
| <input type="checkbox"/> System V.2 source | <input type="checkbox"/> System V.2 binary |
| <input type="checkbox"/> System V source | <input type="checkbox"/> System V binary |
| <input type="checkbox"/> System III source | <input type="checkbox"/> System III binary |
| <input type="checkbox"/> 4.2 or 4.3 BSD source | |
| <input type="checkbox"/> 4.1 BSD source | |
| <input type="checkbox"/> V7 source | |
| <input type="checkbox"/> Other (<i>Indicate which</i>) | |

AUUG

Application for Newsletter Subscription Australian UNIX* systems Users' Group.

*UNIX is a registered trademark of AT&T in the USA and other countries

Non members who wish to apply for a subscription to the Australian UNIX systems User Group Newsletter, or members who desire additional subscriptions, should complete this form and return it to:

AUUG Membership Secretary
PO Box 366
Kensington NSW 2033
Australia

- Please don't send purchase orders — perhaps your purchasing department will consider this form to be an invoice.
- Foreign applicants please send a bank draft drawn on an Australian bank, or credit card authorisation, and remember to select either surface or air mail.
- Use multiple copies of this form if copies of AUUGN are to be dispatched to differing addresses.

Please *enter / renew* my subscription for the Australian UNIX systems User Group Newsletter, as follows:

Name: Phone: (bh)
Address: (ah)

..... Net Address:
.....
.....
.....

*Write "Unchanged" if address has
not altered and this is a renewal.*

For each copy requested, I enclose:

<input type="checkbox"/> Subscription to AUUGN	\$ 65.00
<input type="checkbox"/> International Surface Mail	\$ 10.00
<input type="checkbox"/> International Air Mail	\$ 50.00

Copies requested (to above address) _____

Total remitted

AUD\$ _____

(cheque, money order, credit card)

Tick this box if you wish your name & address withheld from mailing lists made available to vendors.

Please charge \$_____ to my Bankcard Visa Mastercard.

Account number: _____ Expiry date: _____. _____

Name on card: _____ Signed: _____

Office use only:

Chq: bank _____ bsb _____ a/c _____ # _____

Date: ____ / ____ \$ CC type ____ V# _____

Who: _____ Subscr# _____

AUUG

Notification of Change of Address

Australian UNIX* systems Users' Group.

*UNIX is a registered trademark of AT&T in the USA and other countries.

If you have changed your mailing address, please complete this form, and return it to:

AUUG Membership Secretary
PO Box 366
Kensington NSW 2033
Australia

Please allow at least 4 weeks for the change of address to take effect.

Old address (or attach a mailing label)

Name: Phone: (bh)

Address: (ah)

.....
.....
.....
.....
.....

Net Address:

New address (leave unaltered details blank)

Name: Phone: (bh)

Address: (ah)

.....
.....
.....
.....

Net Address:

Office use only:

Date: ____/____/____

Who: _____

Memb# _____