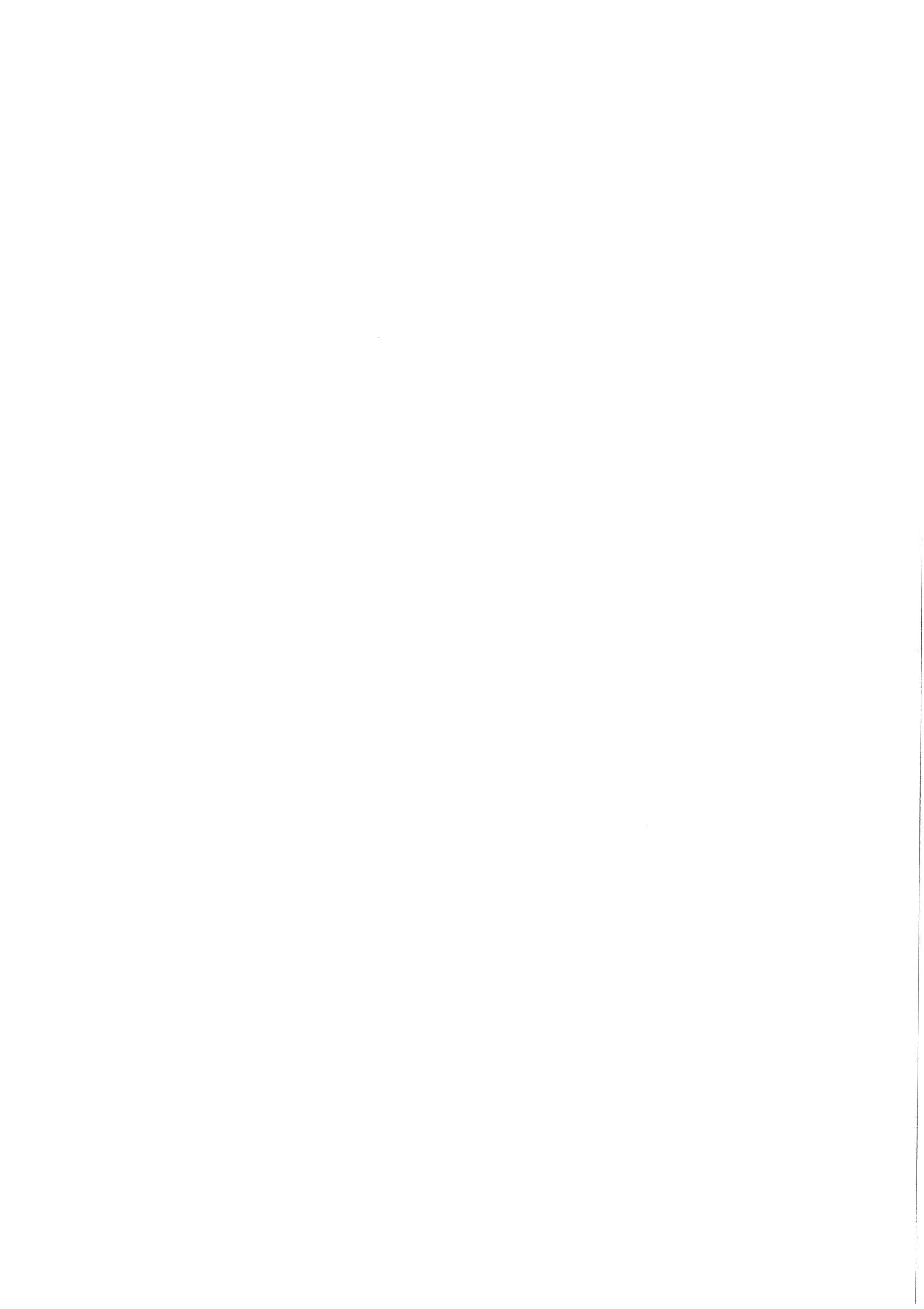


Australian UNIX systems User Group Newsletter

AUUGN

Volume 10, Number 3

June 1989



The Australian UNIX* systems User Group Newsletter

Volume 10 Number 3

June 1989

CONTENTS

AUUG General Information	3
Editorial	4
President's Letter	6
Secretary's Letter	7
Notice of Annual General Meeting	8
AUUG 1989 Election Results	9
Adelaide UNIX Users Group Information	10
Western Australian UNIX systems Group Information	11
AUUG Institutional Members	12
USENIX San Diego Proceedings Offer	14
USENIX 1990 Winter Conference Call for Papers	16
A Simple Data Entry Program	17
Off the Net	19
From the ;login: Newsletter - Volume 14 Number 2	52
Call for Papers - Workshop on Large System Administration III	53
Call for Participation - Distributed and Multiprocessor Systems Workshop	54
Call for Participation - Computer Graphics Workshop	55
Call for Papers - Current Developments and Future Trends of UNIX Systems	56
Call for Participation - Workstation Operating Systems Workshop	57
Book Review - C Traps and Pitfalls	58
C++ Software Tape	59
USENIX Online Index and Library	60
FaceSaver does UUNET	61
Future Events	62
Publications Available	62
Long-Term Calandar of UNIX Events	63
New Release of 2.10 BSD Available	64
4.3BSD Manuals	65

From the EUUGN Newsletter - Volume 9 Number 2	66
An extension to the troff character set for Europe	67
Mk'ing Hardware: A Tutorial	75
A European Representation for ISO C	80
Alternatives to OSI	84
Brussels Conference Report	87
News from the Netherlands	92
EUUG Executive Report	94
UKUUG News	95
UKUUG UNIX Security Workshop	97
Networking Column	101
Puzzle Corner	104
UNIX Clinic	105
Standards Report on C and POSIX	113
UNIX Standards: Repertory, Reps and Representation	115
Software Review	117
USENIX Association News for EUUG Members	122
UNIX System V Data Networking Architecture	124
UNIX Security Workshop	134
Window Systems	135
Book Reviews	144
Calendar of UNIX Events	146
UKUUG Summer Abstracts	149
Management Committee Minutes - February 1989	151
AUUG Incorporated Constitution	156
AUUG Newsletter Back Issues	167
AUUG Membership Categories	169
AUUG Forms	171

Copyright © 1989. AUUGN is the journal of the Australian UNIX* systems User Group. Copying without fee is permitted provided that copies, are made without modification, and are not made or distributed for commercial advantage. Credit to the AUUGN and author 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	Secretary	Tim Roper
	<i>greg@softway.sw.oz</i> Softway Pty. Ltd., New South Wales	<i>timr@labtam.oz</i> Labtam Information Systems Pty. Ltd. Victoria	
Treasurer	Michael Tuke		
	<i>mjt@conjure@labtam.oz</i> Vision Control Australia Victoria		
Committee Members	Frank Crawford	Richard Burridge	
	<i>frank@teti.qhtours.oz</i> Q.H. Tours New South Wales	<i>richb@sunaus.sun.oz</i> Sun Microsystems Australia New South Wales	
	Chris Maltby	Tim Segall	
	<i>chris@softway.sw.oz</i> Softway Pty. Ltd. New South Wales	<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.

Full details will be sent to members in the mail in early July.

AUUG Newsletter

Editorial

Welcome to the Newsletter.

In this issue we have an article from John Lions and also David Horsefall's Off the Net.

We also have our regular reprints from the USENIX and EUUG Newsletters. Please note that we also have a USENIX proceedings offer.

My next task for AUUG will be assemble the proceedings for AUUG89. I am told that Programme Committee headed by Peter Barnes has assembled a good range of speakers and papers for the Conference. As AUUG's major event of the year it is well worth attending. Watch out for our mailer which should get to you in early July.

For the bush lawyer's among you we have a special treat. The AUUG Incorporated Constitution is printed in this issue. Those attending the Annual General Meeting held during AUUG89 perhaps should become acquainted with it.

REMEMBER, if the mailing label that comes with this issues is highlighted, it is time to renew your AUUG Membership.

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 14th of July 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 Labtam Information Systems Pty Ltd.

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

Just a short note this time, in the lead up to the mad rush of AUUG'89. Early indications are that the conference will be a big success, with all the display stands and sponsorships filled, and some extremely interesting speakers, local and overseas.

Peter Barnes and his program committee are doing a great job, as are the organising committee including Frank Crawford and Pat Duffy.

Hopefully, this newsletter will also show the results of the election of officers. I would like to express, on behalf of myself and the treasurer and secretary, Michael Tuke and Tim Roper respectively, gratitude for the vote of thanks implicit in being re-elected unopposed. As Tim said recently, it takes a whole year just to learn how to do the job, it would be a shame to lose that experience. It is time for us to start training replacements, though...

This will be the last one of these you get to read before AUUG'89, so I can say "see you there".

Best regards,

Greg Rose.

Secretary's Letter

The collective mind of AUUG is currently occupied by AUUG89, the 1989 Conference and Exhibition. All the indicators point to this being the most successful AUUG event ever. Acceptance of papers for consideration has closed and rumour has it that Peter Barnes has been sighted somewhere in his office underneath a pile of A4 envelopes and faxes. ACMS have exhibition space booked at 100% capacity with about 30 vendors occupying over 50 stands (and there is a waiting list). We are delighted to have Pat Duffy managing a professional publicity campaign for the conference. Those details again: Tuesday 8th – Friday 11th August, 1989 at the Hilton Hotel in Sydney. All members and previous attendees will receive a brochure containing registration, accommodation and general information in early July. These matters are being managed by ACMS whose phone number is (02)3324622.

Look out for AUUG's own stand at AUUG89. There you will find the FaceSaver camera, AUUGN back issues, USENIX conference proceedings, T-shirts and of course membership information. Why not take the opportunity to renew your membership? Cheques, Bankcard, Visa, Mastercard and even cash accepted.

Elsewhere in this issue you will find the results of the election of office bearers for 1989/90. I would like to record my thanks to the retiring committee members and extend my congratulations to the new ones.

A current project, expected to be off the ground before AUUG89, is, at long last, membership cards. These will serve as receipts, proof of membership (for discounts etc.), and perhaps a useful renewal reminder. (Reminders will continue to be posted.)

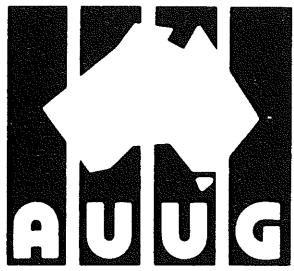
At the last Management Committee Meeting the schedule of membership fees for the 1989/90 financial year was set as follows.

Ordinary members	\$ 78
Institutional members	\$325
Student members	\$ 45

Membership still includes a subscription to AUUGN. You may be interested to know that the Ordinary membership fees is very close to the actual cost of publishing AUUGN and of mailing reminder notices and ballot papers annually to members. Fees for overseas mailing have been brought into line with current costs and there are now two overseas airmail fees according to country zone. New membership application forms are being prepared; in the mean time the old forms and fees will be accepted.

Finally I would like to extend apologies to Hewlett-Packard Australian Software Operation who, due to a clerical error on AUUG's part, have been omitted from the listings of Institutional members published in the previous three issues of AUUGN.

Tim Roper



AUUG Incorporated

P.O. Box 366
Kensington N.S.W.
Australia 2033
auug@munnari.oz.au

NOTICE OF ANNUAL GENERAL MEETING

The 1989 Annual General Meeting of AUUG Incorporated will be held at the Hilton Hotel in Sydney on Thursday 10th August, 1989 at five PM. This is in conjunction with the 1989 Conference and Exhibition.

The agenda is attached. The minutes of the last general meeting of the association, being the 1988 Annual General Meeting, were published in **AUUGN** Volume 9 Number 5 dated October 1988.

Timothy Roper
Secretary

AGENDA

1. Apologies
2. Minutes of last meeting (15th September, 1988)
3. Business arising from Minutes
4. Returning Officer's Report
5. President's Report
6. Secretary's Report
7. Treasurer's Report
8. Meetings in 1990
9. Next Annual General Meeting
10. Other Business

AUUG 1989 Election Results

Returning Officer's Report - Results of ballot for office bearers 1989-1990

President:

Greg Rose

Secretary:

Tim Roper

Treasurer:

Michael Tuke

Committee

Peter Barnes

John Carey

Pat Duffy

Chris Maltby

Returning Officer

John O'Brien

Assistant Returning Officer:

David Purdue

87 votes were received out of 357 ballots sent out

John O'Brien
Returning Officer

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
Adept Business Systems Pty Ltd
Aldetec Pty Ltd
Apscore International Pty Ltd
Australian Electoral Commision
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
Computerscene International
Cybergraphic Systems Pty Ltd
DBA Limited
Data General
Davey Products Pty Ltd
Dept of Industry, Technology and Resources, Victoria
Dept of Lands - Central Mapping Authority
Digital Equipment Corporation (Australia) Pty. Limited
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
Hewlett-Packard Australian Software Operation
IPS Radio and Space Services
Ipec Transport Group
Labtam Information Systems Pty Ltd
Lands Department, Qld
Lionel Singer Corporation
Macquarie Bank Limited
Macquarie University
Main Roads Department, Queensland
Monash University Computer Science
NEC Information Systems Australia Pty Ltd
NSW Parliament
National Engineering Information Services P/L
Nixdorf Computer Pty Limited

AUUG Institutional Members

Olivetti Australia Pty Ltd
Olympic Amusements P/L
Overseas Telecommunications Corporation
Prentice Computer Centre
Prime Computer of Australia Ltd
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 Bank Victoria
State Library of Tasmania
Sun Microsystems Australia
Swinburne Institute of Technology
Tattersall Sweep Consultation
The Australian National University
The Department of Industry Technology and Commerce
The University of Adelaide
The University of Melbourne
The University of New South Wales
The University of Wollongong
University of New England
University of Technology Sydney Computing Services Division
Vicomp
Wang Australia Pty Ltd
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):

USENIX 1990 Winter Conference Call for Papers

Papers are sought in all areas of Unix-related research and development, for the technical program of the 1990 Winter USENIX Conference. Papers which are accepted for the conference will be published in the conference proceedings and shall be presented during the three days of technical sessions at the conference.

Appropriate topics for presentation include, but are not limited to:

- New Tools and "Little Languages"
- Unix and AI:
 - Intelligent Systems
 - Neural Nets
 - Is there any I in AI, or is it all just A?
- Ada and Unix – Real Experience, Future Expectations
- Software Release Systems and Servers
 - Documentation Issues
 - Installation Guides and On-line Help Systems
- Architectures and Compilers
- File Systems and Servers
 - Failsafe/Failsoft File Managers
 - Hierarchical File Migration
 - Version Control
- Distributed Systems and Services
 - Networking and Security
- User Interfaces and UIMS Systems
- Experiences and Novel Applications

All submissions will be considered – however, papers detailing new and interesting work will be regarded much more favorably than thinly disguised product announcements or re-runs of previous reports. The Winter 1990 conference is requiring that **extended abstracts** (and *not* full papers, as in previous conferences) be submitted. An extended abstract should describe the nature of the work, summary of results and conclusions, and should be between 1000-2000 words long (two to three typeset pages). Time is scheduled for authors of accepted papers to complete their submissions; therefore, extended abstracts will only be accepted when it is felt that a complete and worthy paper can be produced by the final due date.

The final paper should include a 100-300 word abstract, a discussion of how the paper relates to other work, illustrative figures (where appropriate), and citations to relevant literature. Only previously unpublished submissions will be considered. Final papers should contain on the order of 8-12 pages of single spaced typeset materials. All final papers must be submitted in a camera-ready format or electronic format (*troff -ms* if possible) – typewritten or dot-matrix output is **not** acceptable as final output. For authors without access to a laser printer or typesetter, appropriate facilities will be provided by the program chair.

Please submit abstracts as soon as possible, and mail one hard-copy and one electronic-copy to the addresses below. The final deadline for receipt of submissions is **August 14, 1989**; abstracts received after this deadline will not be considered. Notification of acceptance or rejection will be made by **September 25, 1989**. Final camera-ready papers are due by **November 17, 1989**.

To submit an extended abstract or request additional information, please contact:

Daniel V. Klein
Washington USENIX Technical Program
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Internet: wash-usenix@sei.cmu.edu
UUCP: uunet!sei.cmu.edu!wash-usenix

Telephone: +1 (412) 268-7791 (W)
+1 (412) 422-0285 (H)
+1 (412) 268-5758 (FAX)

Please include your physical and electronic mail address in all correspondence.

Technical Program Committee

Daniel V. Klein, Chair – Software Engineering Institute, CMU

Eric P. Allman – International Computer Science Institute
Pat Caruthers – Aratar
John Devitofrancesi – University of Illinois, Urbana
Michelle Dominijanni – Concurrent Computer Corporation
Daniel Geer – Massachusetts Institute of Technology
Chet Juszczak – Digital Equipment Corporation

John R. Mashey – MIPS Computer Systems
Michael D. O'Dell – Prisma, Incorporated
Charlie Perkins – IBM, T.J. Watson Research
Dennis M. Ritchie – AT&T Bell Laboratories
Susanne Smith – Windsound Consulting
Alix Vasilatos – Open Software Foundation

A Simple Data Entry Program

J. Lions

University of New South Wales, Kensington 2033

1. Introduction

Some time ago I implemented a simple, pattern directed data entry program for creating sets of similar data files. The original design started as a student exercise in C programming and the use of system calls, and other features were gradually added later. The program is conceptually simple (only 187 lines of C including comments), and satisfies about 90% of my requirements. Unfortunately I do not see any easy way to satisfy the remaining ones without a drastic overhaul and expansion. The purpose of this note is to advertise my program (send mail to `johnl@cheops.eecs.unsw.oz` if you are interested) and also to enquire if anyone knows of another program that is better for the same task but is not significantly more complex.

2. The Present Program

Enter requires a **pattern** file that it uses as a template. In general, there is a one-to-one correspondence between the lines of the **pattern** file and those of a newly created data file. **Enter** reads the **pattern** file line by line. A line that does not contain '?' is simply copied to the output. Otherwise the line is divided: the characters before '?' form the *prefix*, the character after the '?' forms the *operator*, and the characters after the operator form the *suffix*. Depending on the operator **enter** may query the user to obtain needed information to determine whether an output line should be generated, whether the prefix or the suffix or both should be replaced, and whether the line should be repeated.

Any output line generated consists of the prefix followed by a revised version of the suffix without the '?' or the operator. The following rules apply for the currently valid operators shown in brackets:

(\$) the suffix names a string in the command environment that replaces the suffix.

(!) the suffix should be a shell command. This is executed and the first line of its standard output file replaces the suffix.

(?) the suffix is copied to the user's terminal as a prompting message. The user's response replaces the suffix.

(*) this is similar to the previous case (?) except that if the user's reply is an empty line (null string) no output line is generated.

(+) this is similar to the previous case (*) except that the question will be repeated (adding a new line to the output file each time) until the user's reply is an empty line (null string).

(y) the suffix is copied to the user's terminal as a prompting message. If the user's response begins with 'y', the suffix is replaced with a null string otherwise no output line is generated.

The user may respond to a question posed by **enter** with a line beginning with '!' or '!!', for example, '!grep Smith names'. If the case of a single '!', the command will be executed directly with output to the user's terminal and the question will be posed again. In the other case, e.g. '!!grep Smith names', the standard file output generated by the command will be used for the suffix for the new data line.

If in the second case the command output consists of several lines then each, up until the tenth, will give rise to an output line. (Any lines after the tenth will be discarded.) Usually in such cases only one line is correct, and the others will have to be removed using an editor either immediately or after all data have been entered. At each stage the current output file may be viewed using e.g. '!cat %' or edited by responding e.g. '!ed %' to the next question. In this context the usual convention is followed that '%' surrounded by white space is a metacharacter that is replaced by the current output file name.

3. Samples

A sample pattern file is as follows:

```
.ds Pn ?$NUM
.ds Dr "?#!date
.ds Nc "?Formal name:
.de Ac
??Address:
?+ (more):
..
.ds Tc "?Greeting:
.ds Ec "?*ACSNet address:
.DO
.b4?ySend letter now?
.b5?ySend acknowledgement?
```

If **Enter** is called with this pattern file, the ensuing dialogue might appear on a CRT screen as follows (with the user's replies in **bold**):

```
Formal name: Mr F. Flintstone
Address: 123 Pebble Lane,
(more): Boulder, NSW,
(more): Australia
(more):
Greeting: Fred
ACSNet address: fred@flints.oz
Send letter now? y
Send acknowledgement? n
```

As a result of this dialogue, and supposing that the environment variable NUM happens to be 735, the created data file would look like:

```
.ds Pn 735
.ds Dr "Tue May 2 13:56:02 EST 1989
.ds Nc "Mr F. Flintstone
.de Ac
123 Pebble Lane,
Boulder, NSW,
Australia
..
.ds Tc "Fred
.ds Ec "fred@flints.oz
.DO
.b4
```

4. Residual Problems

The present program makes a single pass through the pattern file. It doesn't remember answers from one pattern line to the next, and in particular, it doesn't modify its behaviour based on previous answers. For example, the last question in the above dialogue should not have been asked given that the previous question had been answered affirmatively. The main remaining difficulty that I have is not with saving all the answers, but in devising an improved syntax for the pattern file so that the appropriate concepts can be expressed neatly and succinctly.

OFF THE NET

Hello once again to the next {exciting,boring} issue of "Off the Net". As I said in my last issue, it's a bit difficult writing this stuff under the influence of LDF (Long Delayed Feedback). To put it another way, I received little comment on my first attempt (and what comment there was, was along the lines of "Yes, well...") so I can only hope that everyone likes it. If you don't, write your own articles. At the time of typing this, the second submission has yet to manifest itself, and so once again I must submit gracefully to the printing pipeline.

Young David Purdue sort of flamed me for not getting into the first issue, so I challenged him to post something Worthwhile. This is what eventuated... I was almost tempted to repost it as a forged checkgroups message, but thought better of it:

From: davidp@labtam.oz (David Purdue)
Newsgroups: aus.flame
Subject: Re: Please keep trash out of the wrong newsgroups

In article <1239@murtoa.cs.mu.oz.au>, kreb@cs.mu.oz.au (Robert Elz) appended:
> I append the current descriptions of the aus.* groups for those whose
> newsreaders don't display this information, and as a reminder for those
> who may have unsubscribed and forgotten some of these groups exist.

Close, Robert, but no banana. Try these descriptions:

aus.general	High ranking military officers
aus.ai	Short, high pitched screams (ai!!)
aus.aviation	Planes, Lines, Spheres, etc.
aus.ex-comms	Convicted criminals returned to society
aus.conserve	Jams and Marmelades.
aus.flame	Discussion on the latest fashions in asbestos suits
aus.followdown	Followups to aus.flame articles
aus.fps	Flying Picket Songs
aus.genstat	Statistics on high ranking military officers
aus.jobs	You really don't want to read this, signed, your Boss
aus.kermit	Discussion of the Muppet Show.
aus.lp	New features for the System V print spooler command
aus.mac	Raincoats.
aus.mail	Post invoices, accounts, bills, etc. to aus.fee-mail
aus.netstatus	Can we catch enough fish for dinner?
aus.parallel	Lines, Planes, Spheres, etc.
aus.politics	Hot air ballooning.
aus.pyramid	The wonders of ancient Egypt.
aus.radio	Discussion on your favourite DJ.
aus.scheme	Plots, plans and hidden agendas.
aus.spearnet	Who put all the holes in my net? No fish for dinner.
aus.sun-works	Confirming that dawn will occur tomorrow.
aus.wanted	Criminals still on the run.

Like I said, I was tempted to repost it, but overcame it. Children, don't try this at home...

This next one more properly belongs in aus.bizarre, had it existed at the time. At the time of writing, its existence is still in some doubt. Bizarre, one might say...

From: daveh@cheops.eecs.unsw.oz (David Hume)
Newsgroups: aus.jokes
Subject: Dont run over a duck

relayed to me from my brother in USA:

PM-Duck Death,0353<
Town Pet's Death Brings Unusual Sentence<
Eds: Pekin in 3rd graf is cq.<

STRONGSVILLE, Ohio (AP) — A man who ran over a beloved neighborhood duck has been ordered to do time at an animal farm, buy a replacement duck for the owners and train it to cross the road.

Dozens of people sent cards, letters and flowers to Joseph and Shirley Spak after Danny the Duck's death Jan. 23.

Danny, a Pekin duck about 9 or 10 years old, was a familiar sight to area residents during his frequent jaunts across the road to swim in a creek next to the fire station, his owners said.

"When they'd walk their dogs, they'd stop and talk to him," Mrs. Spak said. "Everybody sure did love the duck."

Mayor Walter Ehrnfelt said he threw the book at Robert Yarsky, 20, of Strongsville, who pleaded no contest to a second-degree misdemeanor charge of killing an animal Feb. 9.

"He said he didn't intentionally run the duck over," Ehrnfelt said Tuesday. The mayor said Yarsky's lack of remorse during the court appearance made him think "it was a deliberate act."

Yarsky was caught because a witness copied down his license number and reported it to police, Ehrnfelt said.

Ehrnfelt fined Yarsky \$750 plus \$28 in court costs and 90 days in jail.

"I suspended the 90 days based on the fact that maybe we could help this individual learn a little bit about animals and the love of pets," the mayor said.

To avoid jail time, Yarsky has to read a book on ducks and spend two days a week for the next two months as a volunteer at the Burnette Animal Farm in Olmsted Township. Ehrnfelt also ordered Yarsky to buy the Spaks another duck and train it, including how to cross the road to the creek.

"I even said I would help him if necessary," Ehrnfelt said.

Yarsky was not at home when telephoned for comment this morning.

His mother, Margaret, reiterated his claim that the death was an accident.

AP-NR-02-16-89 0856EST<

This one will probably get me some time in purgatory...

From: fred@basseroz (Fred Curtis)
Newsgroups: aus.religion,aus.jokes
Subject: Deus ex Machina
Summary: Been there, done that.
Keywords: Very long.
Followup-To: aus.flame

In <642@agsm.unsw.oz> [aus.religion], crifprog@agsm.unsw.oz writes
> I am left with the impression that God follows a set of deterministic
> rules. I reckon I could simulate him with an expert system if
> I could find out what these hypothesized rules and absolutes are.
> ...
> So come on, folks, help me write my God-simulator
> ...
> j.c

What do they teach computer scientists these days? Research has been underway in this field for a very long time, and its fruit can be seen in many applications, with hardware ranging from the earliest efforts of Cabbage's "Ecumenical Engine" to the modern PRAY II supercomputer. The software/theory side is also well established, although I can't think of any commercial products apart from the popular StainedGlassWindows and several DeityBase packages running under the MS-GOD oracular system.

In relation to your particular enquiry, I believe that some Japanese researchers recently derived an equation describing God. The results have not been verified (several key people apparently achieved Enlightenment and promptly vanished), but even if the desired equation did not exist, it would be necessary to invent it.

History

~~~~~

Modern machine-based mysticism has its seeds in the Indeconstructural Revolution, when several CS departments in tertiary institutions around the world were bought out by large Evangelical conglomerates. Not much progress was made until a solitary PDP in Lower Bologna suddenly made the astounding claim that it had been God and God had been it. A great deal of time has been spent trying to reproduce this since the original PDP was deconstructed by zealous undergraduates concerned about the possibly heretical implications of the claim.

#### Computer Networks

~~~~~

This subject is not really my forte, but several researchers here in the Basidiogenetic Dept of Cybertheology are at the forefront

of this technology. There is great excitement at the moment over the possibility of building an omnipresent computer network. The prototype version achieves only partial or 'local' omnipresence, but even still there is a bottleneck which arises because all computers on the cogito-ergo-SUM-IV network receive all messages (a number of my colleagues believe that the network may be interacting with extraterrestrial computers).

The theorists already have plans for an omniscient network where each computer has a Peripheral Machine Deity (PMD) which knows what messages would be received and transmitted. This has the tremendous advantage that the computer may be taken down (or even permanently removed) without causing havoc. No physical connection of machines is needed, obviating the need for error detecting/correcting protocols. The network functions perfectly as long as at least one PMD remains active since it can, of course, simulate all the others. Actual IO between the computer and PMD is accomplished by a Demigod's Magical Interference (DMI) controller.

There is some concern among the uninformed over the possibility that an omniscient network may decide that humanity is superfluous and destroy it. This, of course, runs counter to the Church-Tubing thesis which states roughly that "Computers never go wrong".

Underlying theory

Cybertheologohistorians generally agree that the early formulations of Alfredo Church and Alan Tubing (in particular, the Polycyclic Meta-calculus of Transubstantiated Incarnations) were the crucial turning point in the subject's development. In layman's terms, Tubing proved that all intelligent activity can be shown to be a byproduct of simple chemical processes - photosynthesis for instance. (Experimental evidence was obtained by demonstrating that plants do not think in the dark). The real breakthrough did not come, however, until the development of fractal AI; self-similar intelligences could be compressed into a few-hundred kilobytes of memory. (This gave rise to a flood of make-your-computer-smarter-than-you-if-it-isn't-already boards for PCs). Refinement of these techniques led to more and more godlike pseudo-beings, culminating most recently in myself.

Other applications:

Omniscience is an extremely useful property in a computer program; a compiler can take an incorrect program and produce working object code on the first pass, since it knows what the user intended and how he/she would have corrected the errors. Recent efforts have allowed versions of the UNIQUES operating system to be compiled from the pseudo-random text found in aus.flame [UNIQUES is a trademark of Belle Lobotomies].

GOD@basser.cs.su.oz (Globally Omnipresent Device-driver)

/##\|
|##o#|\|
|#xxx|_|
|xxx/
IUCFL

And I did promise CB YEO a look-in, so what the heck....

From: u3169409@ucssvc.unimelb.edu.au

Newsgroups: aus.jokes

Subject: Kiwis the new long name champs

Kiwis the New Long Name Champs

A New Zealand mountain

Taumatawha katangihangako auauotamatea

10

14

12

(turipukakapikimaungahoronu ku) pokaiwhenuakitanatahu

21

[57 letters]

has replaced the Welsh

Llanfairpwllgwyngyllgogerychwyrndrobwllllantysiliogogogoch

4

25

29

[58 letters]

as the longest place name in the world following the latter's decision to rename itself

Llanfair Pwllgwyngyll.

The New Zealand name means

"The brow of the hill where Tamatea (the man with the big knees who slid, climbed and swallowed mountains) who travelled the land, played his flute to his loved one".

[Does anybody knows the meaning of the Welsh name?]

— — —

cb yeo

[To be followed up of course by various followups...]

From: djh@munnari.oz (David Hornsby)
Newsgroups: aus.jokes
Subject: Re: Kiwis the new long name champs

In article <850@ucsvc.unimelb.edu.au> u3169409@ucsvc.unimelb.edu.au writes:
> Llan fairpwllgwyngyllgogerychw ryndrobwllllantysiliogogogoch
> [Does anybody knows the meaning of the Welsh name?]

>cb yeo

>From an old postcard ...

llan	fair	pwall	gwyn	gyll	goger
church	mary	a hollow	white	hazel	near to
y	chwyrn	drobwll	llan	tysilio	gogo
by	rapid	whirlpool	church	saints	cave
					goch
					red

And another old postcard ...

"Mary's church by the white hazel pool, near the fierce whirlpool,
with the church of tysilio by the red cave."

Personally, I like ...

"Mary's church in a hollow, near to a white hazel, by a rapid
whirlpool, with the church of saints by the red cave."

Any Welsh speakers out there ?

- David.

From: jz@murtoa.cs.mu.oz (Justin Zobel)
Newsgroups: aus.jokes.d
Subject: Re: Kiwis the new long name champs

In article <850@ucsvc.unimelb.edu.au> u3169409@ucsvc.unimelb.edu.au writes:
> Llan fairpwllgwngyllgogerychw ryndrobwlllantysiliogogogoch
>[Does anybody knows the meaning of the Welsh name?]

My mother grew up near this place. According to her, the name is no more than the names of four or five local features, including a couple of nearby towns, run together; giving a kind of invented Welsh peculiarity, intended to attract the tourists, who I understand prowl the vicinity looking for the features the name is supposed to describe. I've got a couple of postcards featuring the sign on the railway platform, and each gives a different interpretation. They generally come down to 'Mary near the hazel tree by a whirlpool of the red cave where there was a church' or similar rubbish.

On the count of three, lets all say 'll' the Welsh way ...

jz

[I understand it is pronounced sort of like a slurred "SSHHRR...".
At least, that's what it sounded like when I asked my Welsh mate
to say it. You have to twist your mouth in a funny way...]

Last issue, I had a piece on signature quotations. Here are a few more. Of course, no-one can possibly out-do Henry Spencer...

--
Gregory Bond (gnb@melba.bby.oz), Burdett, Buckeridge & Young Ltd., Melbourne
"[It] is not a phrase which has been coined by God. It was written by kre
(known to mere mortals as Robert Elz), who is only a demi-god. Therefore,
do not treat his words as gospel, they are merely truth." - djk@munnari

From: anthony@flinders.cs.flinders.oz (Anthony Thyssen)

As I hold aloft my Binary Bible, translated from ancient COBOL, and read
from the first book SYSGEN 1:1

"In the beginning, the Giver of Data (GOD) generated silicon and carbon,
and the system without architecture, and un-initialized, and randomness was
upon the arrangement of the matrix....."

--
Modeless editors and strong typing: |Maarten Litmaath @ VU Amsterdam:
both for people with weak memories. |maart@cs.vu.nl, mcvax!botter!maart

Regular readers will know my opinions on comp.arch. Once in a while,
it comes up with some gems. Here is another:

From: eric@snark.uu.net (Eric S. Raymond)
Newsgroups: comp.arch
Subject: In Search of Breakthroughs -- A reply
Organization: Benevolent Space Xist Retirement Home

In article <745@tetons.uucp>, bb@tetons.UUCP (Bob Blau) writes:
> This is the continuing saga of a startup company and its quest for
> breakthroughs. [You are the president, etc.]

> The first marketeer suggests, "We could be the first to sell a Vax
> plug compatible system!"

"Huh?" I say, "The VAX was a great machine for its day, sentimental favorite
and all that, but its day is past -- that market's all going to supermicros
as fast as they can get comptrollers to sign checks. You're fired!"

> "No, no," another replies, "if you want to go after the fastest
> growing market segment you should build a Crayette on a chip."

"Hmm," say I, "Sexy idea but a loser in the long term. Cray's gonna get
their lunch eaten by the graphics-supercomputer crowd as soon as that crew
figures out how to plug cost-effectiveness, or maybe by the Japanese."

> "You people don't know what big is," retorts a third marketeer. "If
> you want a BIG installed base you should implement the 370/ESA
> architecture on a chip."

"Foo on that!", I reply. "Look what a beating the PCMs have been taking
lately. Let them and IBM fight over a shrinking pie; I want an expanding
one! You're fired, too!"

> "Well," a fourth says definitively, "you could gain access to the
> biggest market by far if you implement a 386, floating point unit,
> super VGA controller, floppy and hard disk controller, and bus logic
> on a chip."

"Congratulations," I say "You've obviously been paying attention to reality.
You're now the new VP for marketing."

> The first engineer suggests "The best use of the extra area would be
> to put in really large caches."

"I can dig it," I say. "We do that, we can get better bang-per-buck out
of our memory-procurement dollars."

> "No, no," another replies, "a better use would be to implement lots
> of parallel functional units to get the cycles per instruction (CPI)
> below 1."

"Who cares about CPI?" I snort. "The customers? Nah. Wake up and smell the
coffee -- most current systems, especially UNIX boxes, are bottlenecked in
I/O, not at the processor."

> "Au contraire," shouts a third engineer, "the most efficient usage
> would be to squeeze two RISC systems on the chip in a multiprocessing
> configuration."

"Forget it," I reply. "Packaging overhead isn't *that* high!"

> "If you could do that," sneers a fourth, "then you'd be better off
> halving the chip area and making cheaper RISC chips."

"Right you are!" I said. "You and that first guy are now running the project.
Go build me a dirt-cheap RISC uniprocessor with big caches."

> The first scientist suggests, "Hey, you could implement a 64 bit
> architecture, with 64 bit integers and 128 bit floating point
> hardware!"

"Right," I say, "and dominate the whole 2% of the computer market that
cares about blindingly fast floating point. Get real!"

> "No, no," the second replies, "a fast scalar RISC with an attached
> vector unit on chip would be perfect."

"Better, much better," I say. "Go talk to the engineers."

> "Hogwash," the third scientist scoffs, "a VLIW on a chip would be a
> screamin' demon."

"Maybe," I say. "But I don't see VLIW as enough of a proven win over RISC
to justify the development risk. Get back to me in three years."

> "If you really want performance," the fourth proclaims, "you'd go to
> massive parallelism, putting a 64 processor (with memory) connection
> machine on a chip."

"I don't care about parallelism," I say, "and neither does anyone else but petroleum geologists and weather forecasters and the rest of that floating-point crows, plus a couple academics. Get back to me in five years."

> What will give the most performance?

Most performance at what?

> What will make the most money?

A dirt-cheap, blazingly fast, UNIX-capable RISC micro.

> What will be the most fun?

Ditto.

--

Eric S. Raymond (the mad mastermind of TMN-Netnews)
Email: eric@snark.uu.net CompuServe: [72037,2306]
Post: 22 S. Warren Avenue, Malvern, PA 19355 Phone: (215)-296-5718

I found this one particularly appealing. I just wish it wasn't so close to reality...

From: crifprog@agsm.unsw.oz
Newsgroups: aus.flame,aus.jokes
Subject: Feminism and violence.

I recently asked a friend of mine (who happens to be a psychology undergraduate student and a self-proclaimed feminist) what were her general opinions of war. Her reply ran something like: "Oh, it's just silly men playing with their phallic-symbol weapons....".

I spent nearly two weeks pondering this profound socio-political insight: why DO most weapons of war resemble the penis? Then, just the other day, I perchanced upon an article in the little-known U.S. publication "International Mercenary's (hung, drawn &) Quarterly". It seems that two companies, Enfield (traditionally a manufacturer of small armaments) and Royal Doulton (who make toilets, porcelain, etc.) have both undergone radical management restructuring as a result of Affirmative Action legislation.

One of the outcomes of this was their joint development of a new military antipersonnel device. Although based on the common steel "bear trap" (highly-sprung serrated jaws which snap closed and bite into the ankle of whoever steps in it), Royal Doulton's ceramic expertise has made it possible to construct the device entirely out of glass! It has been dubbed "the Royal-Doulton - Enfield Vitreous Underground Leg Vice (Antipersonnel)".

The device is designed to be buried just below the surface of

the ground, much like a land mine. The principle advantage of the R.E.V.U.L.V.A, however, is that its glass fabrication renders it invisible to metal detectors. Once buried, it waits, a treacherous trap lurking below an innocent surface, harmless until the shaft of some unfortunate soldier's leg plunges into its opening, rendering him at its mercy.

Because it is designed to be buried, the R.E.V.U.L.V.A. is best suited to Jung-le use; accordingly, the North Vietnamese Army has shown interest, with the intention of supplementing their existing arsenal of booby-traps.

So, folks, here is a fine example of Affirmative Action at work: the production of military hardware resembling WOMENS' genitalia, rather than mens'. Strike another blow towards equality of representation.

SOLIDARITY, SIBLINGS!

j.c

Disclaimer: Don't believe a single word.

And I just had to include this one! Thanks Joe, and the unknown (but traceable) poster...

From: jvoros@monash.edu.au (Joe Voros)
Newsgroups: aus.flame
Subject: an aus.bizarre type occurrence

I spotted this posting, and it has been puzzling me as to meaning. No further comment is implied, or indeed, needed.

> Newsgroups: aus.sf
> Message-ID: <458@pta.oz>
> From: [deleted in the interests of avoiding defamation proceedings]
> Subject: Re: Blake's Seven Sucks
> Summary: I do too

Can some VMS-oid tell me if the following is for real, or is it all a vicious rumour?

From: jacobson@karachi. (Dan Jacobson)
Newsgroups: comp.unix.wizards
Subject: Re: friendly messages

I remember this VMS nastiness:
%E-BUZZOFF Foreign commie terminal type. Buy a DEC terminal and reenter command
--
Dan Jacobson, jacobson@eeecs.nwu.edu, {oddjob,gargoyle,att,...}!nucsrl!jacobson

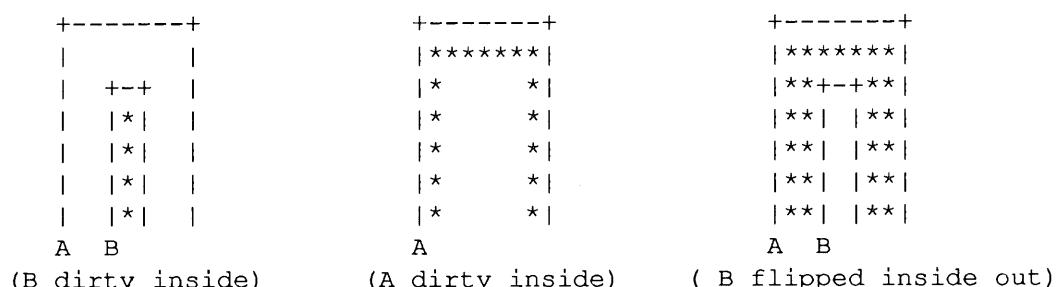
This one, in sci.math, really pickled my fancy. I'll be surprised if our beloved Editor leaves it in here...

From: abali@parts.eng.ohio-state.edu (Bulent Abali)
Newsgroups: sci.math, rec.puzzles, comp.lsi, alt.sex
Subject: Re: The 3 inverter problem

In article <1827@valhalla.ee.rochester.edu>,
badri@valhalla.ee.rochester.edu (Badri Lokanathan) writes:
>On a rather risque note, it is about as useful as the reusable condom
>puzzle: (Hit n if you are easily offended!)
>3 men want to make love to 1 woman. Here are the assumptions:
>(1) There should be no direct genital contact to avoid disease transmission.
>(2) There is a supply of "reusable" condoms with the restriction that
> a dirtied surface may not be used by a different person.
>
>What is the minimum number of condoms required?
>Spoiler hint:
>Simultaneous use of multiple condoms.

The solution of the "condom-puzzle" at the EE bull-pan at Ohio State:

Two condoms A and B are used (by using 2 at the same time) :



Man 1

Man 2

Man 3

Q.E.D.

How about generalizing it to m men, w women, k condoms? Followups to rec.puzzles.

Sincerely,
B. Abali and R. Daoud
(researchers extraordinaire)

-=-
Bulent Abali
Ohio State Univ., Dept.of Electrical Eng.
2015 Neil Av. Columbus, Ohio 43210
abali@baloo.eng.ohio-state.edu

But here are some cleaner versions, for the prudish:

From: price-robert@CS.YALE.EDU (Robert Price)
Newsgroups: sci.math
Subject: The used condom problem
Summary: a less uncouth version

See Martin Gardner's excellent little book "Aha! Insight" for a less uncouth version about surgeons who wish to perform operations without risk of contaminating each other OR the patients with a limited number of sets of rubber gloves. The solution's a pretty little application of combinatorics and there's a good puzzle about a toaster too-- maybe I'll post it next week.

-- Rob Price

From: ilan@Gang-of-Four.Stanford.EDU (Ilán Vardi)
Newsgroups: sci.math
Subject: Re: The condom problem

The algorithm of Hajnal and Lovasz (and their lower bound) appeared in [A. Hajnal and L. Lov'asz, 'An algorithm to prevent the propagation of certain diseases at minimum cost,' in 'Interfaces between computer science and operations research,' edited by J.K. Lenstra, A.H.G. Rinnooy Kan, and P. Van Emde Boas, Matematisch Centrum, Amsterdam 1978].

My improvement has not been published. In fact one journal rejected it due to its 'explicit' content. If you want a preprint, just send me E--mail and I'll mail you one.

Who are all the mad mail-writers? Just look at this snippet, from our resident Guru-Lama, KRE the Great:

From: krek@cs.mu.oz.au (Robert Elz)
Newsgroups: aus.mail
Subject: Random meaningless statistics

In Feb, more than 120Mb of international mail (not news) passed through munnari, one message every 57 secs on average. Average size about 2850 bytes.

What you make of all this I have no idea ... but this was the busiest month ever (notwithstanding it is a short month).

kre

ps: so far this month we're running at about 54 secs per message.

This one, again from Joe V, might get us (that's you and me, John) into trouble. Then again, it might not...

From: jvoros@monash.edu.au (Joe Voros)
Newsgroups: aus.flame,aus.jokes
Subject: Selecting a wife -- an aus.bizarre type article
Summary: The two appetites for sex and food are compared
Keywords: joke

While walking around a supermarket, one becomes aware of a certain similarity between the purchasing of goods for later consumption and the choosing of a wife. Certainly those who restrict themselves to a particular department or section will have a greater chance of making a selection which will turn out to be quite inadequate in the long run due to the lack of appreciable comparisons which could have been made while still in the store. It would seem, then, that a tour of all possibilities is a desirable practice as this will alert the buyer to the wide choice available, even within the few categories there are, and allow a competitive comparison ensuring that as high quality an item as is available can be chosen.

Firstly, it is necessary to have some idea of the *type* of item desired, be it wife or shelf-good. Do you want a nice-looking, sweet-tasting ice cream that will gratify your senses at least for a while, but be of no use at all nutritionally? Or will a slightly less appealing though still quite appetizing steak and potatoes cover your particular needs adequately?

Along the aisle in the fruit/veg department, care must be taken not to choose something which has been left on the shelf for too long, as these can often be bitter, or even rotten to the core. A question of appearance is also evident. Beware of very appealing fruits which may look delectable, as these can often be the worst choices. What is to say that the orange eclipsed by the beauty in front of it is not, in fact, better inside than its competitor, while being only slightly worse for wear? Avoid fruits which have been overly groped or mis-handled as these may tend to get mushy at the juicy bits. Test the fruit coyly, as a particular danger is a Department Head who will force you to purchase an item he has caught you testing. Avoid places where examination is only allowed *after* purchase. Some vegetables become limp and boring if kept on the boil for too long, so think carefully about the specific use you will have for your choice, and whether it is suited to your appetite.

Frozen foods, many of which have been frozen for so long that a tedious process of thawing out is necessary, present a new set of dangers to be avoided by the prospective customer. Beware of artificial colours and preservatives since these can result in a misleading appearance disguising what is really underneath. This is particularly true of the meats, and some of these are so tough that much manipulation is required to make them tender enough to be enjoyed. Also, packaging that seems easy to open may, in fact, be difficult or even impossible, so establish whether or not you will be able to get the product out of its wrapping with ease, *before* buying.

There is a wide variety of desserts available, many of which can tantalise and pleasingly stimulate the senses for quite a while, though soon the true value of their frivolity will become apparent. Their lack of substance and quite obvious single forte will not make them worth keeping for too long. Prolonged exposure to this type of product will deaden the senses and this truly luxurious item will become dull. Even a diet of lotuses can pall.

As for herbs and spices, only use these for embellishment around the edges if the main course is becoming a bit tiresome. "T.V. Dinners" are of little nutritional value except to satisfy a sudden pang, and are not recommended as a staple diet, though they are good for a quick nibble.

In the liquor department, age is the key factor. A good, aged wine [not too old, though, or the tang and zest of its youth will be gone] will give a mellow, mature touch and is not apt to be too sharp and unfledged as with the younger wines. Do not rely on the appearance of the bottle, however, for even the most seemingly aged wine can be quite immature. Avoid very alcoholic wines as these can often be misleading in their impressions. But, no matter which choice is made, without exceeding care, a sour after-taste could easily result.

Thus it can be seen, that careful consideration must be given to both the choosing of wives and dinners; or what it thought to be a bed of roses could turn out to be a dish of tripe from which indigestion is the only effect. Refrain from impulsive choices and consider every possibility before reaching a decision. By following this brief guide and avoiding over-indulgence, you will reduce the danger of your heart ever getting burnt.

--
Joe Voros, Department of Physics, Monash University.

Here's a little something you can all try at your terminals...

From: richb@sunchat.oz (Rich Burridge)
Newsgroups: aus.jokes
Subject: UNIX Humor.

It's hard to tell who the originator of this was, but it came to be from Dan Heller, who got it from Leo Schwab, who got it from Richard Krawitz ... I've tried some of these and I get different results; just passing them on.

[...]

Note that the '%' prompt indicates that the command should be issued from the C shell, and the '\$' prompt indicates the Bourne shell.

```
% rm meese-ethics
rm: meese-ethics nonexistent
```

```
% ar m God
ar: God does not exist

% "How would you rate Reagan's incompetence?
Unmatched ".

% [Where is Jimmy Hoffa?
Missing ].

% ^How did the sex change^ operation go?
Modifier failed.

% If I had a ( for every $ Reagan spent, what would I have?
Too many ('s.

% make love
Make: Don't know how to make love. Stop.

% sleep with me
bad character

% got a light?
No match.

% man: why did you get a divorce?
man:: Too many arguments.

% ^What is saccharine?
Bad substitute.

% %blow
%blow: No such job.

% \(-
(-: Command not found.

% sh

$ PATH=pretending! /usr/ucb/which sense
no sense in pretending!

$ drink <bottle; opener
bottle: cannot open
opener: not found

$ mkdir matter; cat >matter
matter: cannot create
```

You get some beauties in sci.space (long may Henry live). Here are some choice examples:

From: EHARNDEN@AUVMS.BITNET (Eric Harnden)
Newsgroups: sci.space

Subject: Re: SPACE Digest V9 #247

re: telescope stories

what about: there's a guy i know who uses his cat to clean his instrument. he removes the optical component package at the viewing end and slides the animal through the tube to pick up the dust.

Eric Harnden
EHARNDEN@AUVMS

[And if I may digress for a while, here are some followups I got, when I posted above article locally...]

From: dave@stcns3.stc.oz (Dave Horsfall)
Newsgroups: aus.followup
Subject: Re: Bizarre sci.space article

In article <992@wacs.vax.OZ> janet@wacs.vax.uwa.oz.OZ (Janet) writes:

|
| If I recall correctly, they (sci.space) gave up on this idea when someone
| wondered how you'd get rid of the cat hairs afterwards.

Here is some mail I received on the subject, reproduced with permission:

)Date: Fri, 17 Mar 89 07:48:44 est
)From: <geoffk@bio73.unsw.oz>
)Subject: Re: Bizarre sci.space article
)
)I remember one of the large particle accelerator groups (~12 years ago) was
)using ferrets to clean out the accelerator tube - sort of a live bottlebrush.
)Sorry that I can't remember the source, but it was authentic.
)
)Date: Wed, 22 Mar 89 10:30:13 est
)From: <geoffk@bio73.unsw.oz>
)To: dave
)Subject: Re: Bizarre sci.space article
)
)Post it if you wish...incidentally, I think that I recall that they used ferrets
)because they don't molt - you will have to verify that with someone who keeps
)the (disgusting) creatures.

--

Dave Horsfall (VK2KFU), Alcatel STC Australia, dave@stcns3.stc.oz
dave@stcns3.stc.oz.AU@uunet.UU.NET, ...munnari!stcns3.stc.oz.AU!dave

[OK, and now back to your regular program]

From: Ralf.Brown@B.GP.CS.CMU.EDU
Newsgroups: sci.space
Subject: Re: First concert from space--update

In article <380@dvnspcl.Dev.Unisys.COM>,
 tom@dvnspcl.Dev.Unisys.COM (Tom Albrecht) writes:
 } In article <1885@randvax.UUCP>, talmy@randvax.UUCP (Shel Talmy) writes:

}> A company in Los Angeles called Orbit Productions has been formed to
}> stage the first ever concert from space. A large portion of the proceeds
}> from this venture is to be donated to various charities.
}
}
}Hold on to your hats ... "Disaster Area" is on the way!

And if they get the orbit right, the audience will even be at the optimum 37-mile distance.... (sorry, couldn't resist)

--
UUCP: {ucbvax,harvard}!cs.cmu.edu!ralf ===== Voice: (412) 268-3053 (school)
ARPA: ralf@cs.cmu.edu BIT: ralf%cs.cmu.edu@CMUCCVMA FIDO: Ralf Brown 1:129/31
Disclaimer? I claimed something?
You cannot achieve the impossible without attempting the absurd.

From: kev@ist.CO.UK (Kevin Holmes)
Newsgroups: sci.space,sci.space.shuttle,misc.misc,rec.music.misc
Subject: First concert from space

Heyyyyy....

If these guys could wait a few more years then they could make use of some of the proposed (?) SDI lasers for...

Stage Lighting

(Yes, I know about light in vacuum, but I thought it was about time someone lowered this proposal to the ridiculous level it deserves...
:-))

Kev Holmes kev@ist.CO.UK "Selling software is just like Imperial Software Technology prostitution; You've got it, You sell Reading, Royal Berkshire, UK it, You've still got it!"
D. Lambert (IST)

From: wbt@cbnews.ATT.COM (William B. Thacker)
Newsgroups: sci.space
Subject: Re: Henry Spencer, the Movie

In article <21969@ames.arc.nasa.gov>,
mike@ames.arc.nasa.gov.UUCP (Mike Smithwick) writes:
>
>What I'm really waiting for is >>> Henry Spencer the Movie <<<
>:-)

I saw a section of the script for this; the working title is, I believe, "Spencer for Higher." The clip I saw was a climactic showdown between our hero and his evil nemesis, Dietz Vader:

D.V.: "With the moonbase controlled by my robotic automations, the weaklings of Earth lie helpless before me !" [maniacal laughter]

NASA: [portrayed by fat, slowmoving bureaucrat-types] "Damn ! If only
AUUGN

Congress had OK'd the manned base, we could have prevented this !
Help us, oh Henry-wan !" [cries of anguish]

H.S.: "Sorry, boys, but you're on your own." [more cries of anguish,
maniacal laughter from D.V.]

D.V.: "I knew you would turn to the dark side of the moon... er, force."

[a remote camera pans over the robots working the missile silos of
Moonbase, when suddenly, a bright flash appears; the base is wiped out,
levelled, but then begins to change before our eyes; trees spring up,
roads, buildings; soon, an entire city exists, complete with a Moosehead
brewery, and a Canadian flag flies from the tallest building, rippling
gracefully in the solar wind.]

D.V.: "Huh ?"

H.S.: "Hah-hah ! While you were occupied with undermining NASA, my private
space venture firm secretly completed their Genesis device !"

D.V.: "You mean...?"

NASA: "No, no !!!!"

H.S.: "Yes. The Moon has been Torontoformed !"

(These characters are fictional. Any resemblance to sci.space posters is
strictly coincidental. And even if it isn't, it's just a joke, OK ? 8-)

----- valuable coupon -----
Bill Thacker att!cbnews!wbt

"You made a killing in real estate and NASA,
selling cemetery plots in outer space;
'till some falling profits crashed upon your doorstep.
Welcome to the Human Race." - Timbuk 3

Disclaimer: Farg 'em if they can't take a joke !

----- clip and save -----

I guess I'm entitled to a little artistic licence, so here's one of mine:

From: dave@stcn3.stc.oz (Dave Horsfall)
Newsgroups: aus.jokes
Subject: The ACS, masons and catholics (aus.bizarre)

Perhaps it's my warped sense of humour, but the following, taken from
the ACS Newsletter, really pickled my fancy:

"Meetings are always ... held on the fourth Tuesday of the month
commencing 6pm at the Masonic Centre; Catholics are welcome."

--

Dave Horsfall (VK2KFU), Alcatel-STC Australia, dave@stcns3.stc.oz
dave%stcns3.stc.oz.AU@uunet.UU.NET, ...munnari!stcns3.stc.oz.AU!dave

Even the outer depths of the USENET sewers, news.*, can provide some gems amongst the muck:

From: jeffd@ficc.uu.net (jeff daiell)
Newsgroups: news.groups
Subject: Re: CALL FOR DISCUSSION -- COMP.VIRUS
Summary: If This Keeps On....

In article <582@wsu-cs.uucp>, jjb@wsu-cs.uucp (Jon J. Brewster) writes:
> In article <777@twwells.uucp> bill@twwells.UUCP (T. William Wells) writes:
> >In article <203@shadooby.cc.umich.edu>,
 wisner@shadooby.cc.umich.edu (Bill Wisner) writes:
> >: (Curt Stevens)
> >: >here is my yes vote for comp.virus (or whatever the backbone in its
> >: >infinite wisdom chooses to call it eventually).
> >:
> >: Backbone? I don't see a backbone here. Any of you see a backbone?
> >
> >Nope, not here. I'm an invertebrate punster.
> >
> >So slug me!
> >
> All I see is a little ribbing going on... (ahem)

Make no bones about it, people: when it comes to humor-writers, we are *definitely* down to a skeleton crew! But that's OK; I'm not marrow-minded.

Jeff Daiell

P. S. Would an unmarried chiropractor be a spinester?

P. P. S. These jokes were approved by Saul of Tarsal.

P. P. P. S. Don't wait to pass these jokes on -- carpal diem!

(I know, I know; tibia good punster, you have to have a humerous attitude!) Femur of these jokes, email me!

INDEPENDENCE FOR TEXAS -- !otra vez!

--

"Why can't they be like *we* were -- perfect in every way?
What's the matter with kids today?"

From "Bye, Bye Birdie"

And now, the canonical list of Shtones. Just be glad I spared you

from the entire discussion in comp.arch...

From: alan@pdn.nm.paradyne.com (Alan Lovejoy)
Newsgroups: comp.arch, rec.humor
Subject: "?*stones"

--- The Stone Collection ---

14 March 1989

Compiled by Alan Lovejoy

Largo, Florida USA

The following is a list of mostly whimsical benchmark names that have been proposed by a host of people on USENET, by my fellow employees at AT&T Paradyne, and by myself. These benchmark names are modelled on the names of some actual benchmarks (Whetstone, Dhystone, Dhampstone, Rhealstone and Iostone) which usually have the form ?h*stone.

I regret to say that when I first started compiling this list, I was only doing it for my own personal use and therefore it never occurred to me that I should keep track of authorship. I would appreciate it if any authors could contact me to claim their rightfull credit if they see their benchmark name listed here without attribution.

Many of the names were suggested by multiple sources. I have made no attempt to figure out who was first; it is best to simply assume that all suggesters arrived at their names independently, and to leave it at that.

Benchmark Name: Object(s) of benchmark

Author

--- Names suggested by me:

Fhlintstone:	computer graphics animation	Alan Lovejoy
Lhintstone:	measures performance of "lint"	Alan Lovejoy
Whystone:	funniness of ?h*stone benchmark name	Alan Lovejoy
Bhrigestone:	computer aided civil engineering	Alan Lovejoy
Fhirestone:	thermodynamic modelling	Alan Lovejoy
Ghrindstone:	text processing	Alan Lovejoy
Lhoadstone:	program loading	Alan Lovejoy
Bhirthstone:	process creation	Alan Lovejoy
Hhengestone:	celestial navigation	Alan Lovejoy
Philosopherstone:	artificial intelligence	Alan Lovejoy

(I have seen the same or similar suggestions from others that were posted after I thought these up myself. I have never posted any except Fhlintstone until now, however)

--- Names suggested by my coworkers:

Hheadstone:	rate of defunct process creation	Steve Fowler
Rhinestone:	from Duke University...measures program ESP	Roy Crabtree
Warestone:	hirsuteness	Ware Lynch
Wherestone:	navigation	Ware Lynch

--- Names suggested by USENET posters whose names I failed to record

Rhosettastone: natural language translation (I can't believe I missed this!)
Rhollingstone: DSP music generation
Fhlagstone: boolean algebra/logical machine operations
Lhinestone: vector graphics
Lhynestone: wire-fram graphics
Bhlarneystone: for use in measuring performance of vaporware
Lhivingstone: anthropological ability
Bhlackstone: magical ability; unix wizardry
Ehverybodymustgetstone: hipness of a system
Slyandthefamilystone: funkiness of a system
Baudstone: serial io
Sahndstone: silicon compiler performance
Soahpstone: daytime tv viewing
Whatstone: statistical verification of random number generators
Whystone: heuristics
Nhetstone: measure of similarity to USENET
Nhewstone: measure of similarity to television news
Khidneystone: medical diagnostic expert systems
Ghallstone: measure of frenchness
Yhellowstone: ability to put out fires
Mhilestone: ability to meet deadlines
Mhillstone: degree to which something impedes progress
Sticksandstone: measure of how insulting something is
Thunderstone: noise generated by union leaders after a strike is called
Touchstone: pixel resolution of touch screens
Standingstone: availability of goods in the Soviet Union
Swhordinthestone: elected official's ability to lead
Lhymestone: mineral exploration software

--- Names suggested by USENET posters whose names I did record

=====

From: uunet!gatech.edu!abhijit%pyr (Abhijit Chaudhari)
Rhollingstone: A benchmark used to determine the amount of distortion in
the power supply.

Phrecioosstone: Measure of the \$\$ spent on a computer system.

Hhstone: A benchmark for determining the quality of benchmark names.

Lhithpstone: Measures the efficiency of the lisp interpreter.

Stohnehenge: Used to measure the power of previous generation computers.

Xhf25qcdtstone: Measures the efficieny of a random number generator.

Khillstone: Benchmark measuring the physical damage caused to a compter
terminal when a process crashes.

Swhearstone: Efficiency of verbal abuse (can be run alnog with Kihllstone).

rm*hstone: Measure of the amount of work undone by a silly error.

=====

From: uunet!ernie.Berkeley.EDU!carlson (Richard L. Carlson)

I just thought of a benchmark I would actually like to see run:

Mhenustone : measures the number of pop-up / pull-down menu items that a system can draw per second. It would test raw CPU performance, special graphic hardware support, and operating system efficiency; and it would probably give a good indication of a menu-based system's speed as seen by an interactive user.

=====

From: uunet!RELAY.CS.NET!BROOKS%csss-a.prime.com
David Brooks Internet: BROOKS@CSSS-A.PRIME.COM
uucp: {mit-eddie,uunet}!csss-a.prime.com!brooks

Shumnerstone

Lhodestone

Fholkestone (English end of Channel Tunnel)

Shoapstone

=====

From: uunet!cbnews.ATT.COM!ara (a.r.adolt)
from: bob adolt
att-bl

HharryStone = legal "bench"-mark.

=====

From: uunet!decwrl.dec.com!amcad!pop!prem (Prem Sobel)

"Ehlizastone", a measure of how many times per second a computer and operating system can misunderstand or misinterpret the commands you give.

=====

>From: esf00@uts.amdahl.com (Elliott S. Frank)
Elliott Frank ...!{hplabs,ames,sun}!amdahl!esf00 (408) 746-6384
or!{bnrmtv,drivax,hoptoad}!amdahl!esf00

ShlyStone -- measures speed of machine in providing random answers to meaningless questions.

=====

Path: pdn!uunet!peregrine!elroy!ames!mailrus!iuvx!watmath!watmsg!sccowan
>From: sccowan@watmsg.waterloc.edu (S. Crispin Cowan)
Reply-To: sccowan@watmsg.waterloo.edu (S. Crispin Cowan)
Login name: sccowan In real life: S. Crispin Cowan
Office: DC3548 x3934 Home phone: 888-6375
Post Awful: 280 Phillip St., Apt. B1-15 Waterloo, N2L 3X1
UUCP: watmath!watmsg!sccowan
Domain: sccowan@watmsg.waterloo.edu

lhodestone: how long it takes to load a program/boot the OS.
thombstone: how many processes the OS can kill/sec.

=====

>From: aglew@mcdurb.Urbana.Gould.COM

> Ehstone -- how many Canadians, like, worked on the system, eh?

Je me souviens-stone -- how many expatriate anglo-quebecois
worked on the system?

=====

>From: snoopy@sopwith.UUCP (Snoopy)
tekecs.gwd.tek.com!sopwith!snoopy qiclab!sopwith!snoopy
sun!nosun!illian!sopwith!snoopy parsely!sopwith!snoopy

Fhrankenstone: number of times benchmark can die and be resurrected per second

This benchmark is currently very dangerous to run, as an electrical storm is required. It is hoped that a low power technique can be found for the next release. An industry spy overheard the principal researcher asking his assistant to "Put the stack-pointer back." It is believed that the researcher was trapped in a subroutine at the time. Sun's SPARC technology currently leads the pack in this benchmark, but mAMP, a joint venture between MIPS and AMPhenol hopes to overtake SPARC with their on-chip battery design fed by special high-current connectors which also serve as the chip's heat sink. Company spokespersons were unwilling to reveal how many milliamp-hours their product would cache on-chip. Public confidence in the new product was damped at an early demo of a prototype unit which used a defective cpu and a pair of automotive battery clamps from a junkyard. The prototype paniced easily, and the public demanded it be dismantled. Improvements to the prototype, including a voice-synthesis unit, has calmed public fears of the new technology. It's not easy being on the cutting edge. "Hard drives and kilobytes are tinkertoys, I'm talking about the central networking system."

=====

>From: dave@motto.UUCP (dave brown)
| David C. Brown | uunet!mnetor!motto!dave |
| Motorola Canada, Ltd. | 416-499-1441 ext 3708 |
| Communications Division | Disclaimer: Motorola is a very big company |

Sorry, I can't resist. More benchmarks:

Bhirthstone - speed of new product development (measured from conception to delivery).

Hheadstone - speed of product obsolescence.

Kheystone - speed of DES cracking (ref. recent RISKS)

=====

>From: shields@yunccn.UUCP (Paul Shields)

Fhred Flintstone: tests for neanderthal operating systems
Rhollingstone: For use on CMOS technology
Throwstone: tests for glass houses

Mhonumentalstone: In mockery of benchmark tests
(with apologies to P.B. Shelley)
Yheatstone: "Too long a sacrifice / Can make a stone of the
heart / O when may it suffice?"

=====

>From: pa1343@sdcc15.ucsd.edu (John J. Marco)
| John J. Marco -- pa1343@sdcc15.ucsd.edu |
| ...!uunet!sdcc15.ucsd.edu!pa1343 |

Khirystone: The noise volume of the cooling fan of a PC.

=====

In addition to new names, new usages for that name are also welcome.

Enjoy.

--

Alan Lovejoy; alan@pdn; 813-530-2211; AT&T Paradyne: 8550 Ulmerton, Largo, FL.
Disclaimer: I do not speak for AT&T Paradyne. They do not speak for me.
American Investment Deficiency Syndrome => No resistance to foreign invasion.
Motto: If nanomachines will be able to reconstruct you, YOU AREN'T DEAD YET.

Another one that really bought my eye, this one from the great ADO:

From: ado@elsie.UUCP (Arthur David Olson)
Newsgroups: comp.unix.wizards
Subject: Using ksh to count the number of spaces in a string fails :-)

Our version of ksh may be a lot older than ksh88, but it is entertaining:

```
Script started on Thu Mar  9 20:03:58 1989
elsie$ ksh -c " "
ksh: no space
elsie$ exit
```

```
script done on Thu Mar  9 20:04:08 1989
```

-- Arthur David Olson ado@ncifcrf.gov ADO is a trademark of Ampex.

Ah yes - I knew I had these squirreled away somewhere... For new readers, what happened was that the Doggie-Net modem providers tried to break into the USENET market. I think they need to behead their marketoids...

From: csg@pyramid.pyramid.com (Carl S. Gutekunst)
Newsgroups: comp.dcom.modems
Subject: Re: HST Package for USENET

In article <18612@mcdchg.chi.il.us> kowals@studsys.mu.edu (kowalski) writes:
>[I apologize to any offended by its nature. -mod]

Offended? No. Puzzled, certainly. Amused, too.

>USENET NODES....

>As a USENET node, I'm sure you are interested in reduced board time....

Node? Board time? Is that something like MTBF?

>Now authorized USENET nodes can upgrade their boards....

Boards? Those are what I plug into my card cage.

>U.S. Robotics-- the leader in high speed modems to the sysop community....

Sysop? Is that something like system call? Or pseudo op?

I hate to publically belittle a vendor, but it is pretty obvious that USR took their generic BBS offer, and did a global-search-and replace from "Fidonet" or whatever to "Usenet." How can I expect USR to respond to my needs when they won't even bother to investigate Usenet enough to get the terminology right? Usenet is not a BBS, but apparently USR doesn't understand that.

The entire premise of this article is that Telebit offers only a proprietary non-standard modem, and USR is offering standard modems. This neatly neglects to mention that USR has been selling "proprietary non-standard" modems for years, and of course also neglects the fact that Telebit is also now selling true V.32 modems. Even the name "Dual Standard" is terribly pompous, since it suggests that the HST proprietary protocol somehow became a "standard." Well, yes, maybe to the BBS community. But I'm a Usenet "node," what do I care about BBS standards that I cannot use? I *do* care about the "dual standard" Telebit T2500; I can use *both* of its "standards."

For that matter, even today true V.32 modems are not exactly falling off the trees. This will change over the next year or two, but I'm sure not throwing my TrailBlazer away yet. Telebit has proven they will bend over backwards to meet my needs. USR has proven only that they can make an inexpensive modem.

The new USR modem may be a terrific product. But this announcement turns me off something awful.

<csg>

From: snoopy@sopwith.UUCP (Snoopy)
Newsgroups: comp.dcom.modems
Subject: Re: HST Package for USENET

In article <18612@mcdchg.chi.il.us> kowals@studsys.mu.edu (kowalski) writes:

| GET UNIVERSAL HIGH SPEED
| MODEM COMPATIBILITY AND SAVE
| \$1,100!!

A "universal" high speed modem that doesn't speak the most common high speed modem protocol on usenet. Doesn't sound very universal or compatible to me.

| As a USENET node, I'm sure you are interested in reduced board time and
AUUGN

| lower phone bills through the use of high speed modems. However, up to
| now you had to choose a proprietary non-standard high speed modem or a
| high priced incompatible V.32 modem.

I've heard of atime, ctime, mtime, GMT time, and localtime, but this
"board" time is a new one. Must have something to do with being
universally incompatible.

| Now authorized USENET nodes can upgrade their boards to full V.32
| compatibility at an extraordinary savings.

I guess unauthorized USENET nodes like kremvax are out of luck.

| U.S. Robotics-- the leader in high speed modems to the sysop community--
| is offering USENET nodes the fully CCITT V.32 compatible, Courier Dual
| Standard high speed modem for only \$699. A savings of \$900 off the list
| price!

I paid less than that for my modem and it goes faster.

| That's right, 9600 bps world wide V.32 compatibility, plus access to the
| entire installed base of HST's. And at a price some companies charge for
| their 2400 bps modems.

What companies currently charge \$699 for a 2400 baud modem and how many
are they selling? Many 2400 baud modems sell for significantly less.

| The HST Dual Standard supports not only high speed V.32, but also 2400,
| 1200, and 300 bps, ensuring you that your node can now talk to everyone--
| worldwide!

But it can't talk at high speed to anyone I talk to.

| A special bonus for High Speed modem users--
|
| Over the years, many of you may have purchased non-standard modems. The
| problem-- they are only single sourced and they are NOT V.32 compatible.
| If you have one of these non-standard products (and we won't mention any
| names), USR will give you a special bonus--
|
| Get HST/ix software (a \$300 value) for \$100-- an additional \$200
| savings!!!

Let me guess, a MS-DOS binary?

| 1.) First contact USR at 1-800-Dial USR and speak to Susan Pompa--
| she'll ask you a few questions about your board.

My board is made of Lexan (tm) and I slice tomatoes on it.

| 2.) Then send us \$699 (check or money order). Don't send us your old
| modem yet-- we wouldn't want your board to be down for one minute.

But I don't use my old modem to slice tomatoes. It would be a pain to clean.

| 3.) When you receive your USR Dual Standard, install it on your board
| and send us your warranty card, your old modem and \$100. We'll send
| you the HST/ix software, a \$300 value.

I hope the USR Dual Standard is dishwasher safe. Oh, I get it, dual standard means you can wash it by hand *or* in the dishwasher! Sort of expensive for slicing tomatoes on, though. I think I'll keep my lexan.

| You're important to U.S. Robotics.

So important that U.S. Robotics is willing to make a fool of themselves by going after our business without having the slightest idea of what we want or need in a modem.

| We want to help you maintain your board with the latest communications
| equipment.

Thank you, but I think my cutting board can get by without it.

| With the HST Dual Standard, we have the best USENET node solution.

No you don't.

| In short--your node will be able to communicate with everyone.

No, my node can communicate with everybody now, but if I used your modem I'd be stuck at 2400.

| It's our way of introducing the product to the UNIX community and to
| support your USENET efforts.

If you are interested in the USENET market, I suggest you learn a thing or two about it. First, the vast majority of traffic is compressed in the host. For good reason, it's more efficient that way. So don't bother quoting throughput figures using compression in the modem. We don't care. Second, what counts in a USENET modem are: (1) throughput sending compressed news batches via UUCP 'g' protocol, (2) ability to maintain a connection and reasonable throughput when the phone line quality is poor, (3) reliability of the hardware and good technical support when problems occur, (4) use of the 'AT' command set is probably a good idea, (5) decent interactive response for dial-up sessions, (6) SLIP support may very well become important in the near future, (7) some sites would like FAX support, (8) naturally, it also needs to speak 2400/1200/300

Items 1 and 2 are currently best achieved via the PEP(tm) protocol. There is therefore a large installed base of PEP modems, and any high-speed modem wishing to penetrate the usenet market had better talk PEP. V.32 would be nice, but probably not essential to most sites at this time. Sites that also talk to BBS systems might care about talking HST.

Usenet sites run on a wide variety of hardware which are *not* binary compatible with each other, and never will be. Software needs to be source, in a very common, efficient language. C is a good choice.

A modem that talks PEP and V.32 and HST and 2400/1200/300 might be

very popular. Anything less is NOT universal. Naturally, the price would have to be reasonable.

Good luck.

/_____\ Snoopy "My dot-matrix does Postscript."
/_____\ tekecs.wv.tek.com!sopwith!snoopy qiclab!sopwith!snoopy
|__| sun!nosun!illian!sopwith!snoopy parsely!sopwith!snoopy

[Funny - we never heard from that manufacturer again... DH]

And now, for all you OSF followers, who knew that OSF was a Good Thing, but weren't Quite Sure what it Stood For...

From: daa@siesoft (David Allsopp)
Newsgroups: comp.unix
Subject: OSF - what does it *really* mean?

We've all heard of the OSF, but what do those mystic initials really stand for? Here are a few suggestions....

Outer Space Fallguys
Over-Sized Fish
Organisation of Shoe Fetishists
Outlandish and Strange Footwear
Opentoed Sandals are Funky
Omani Sex Fiends
Overly Stupid Frenchmen
Outdoor Sex is Fun
Outrageously Silly Foreigners
Over-powerful, Stupid and Foreign
Obstructivist Set of Fools
Obese, Short and Flabby
Old Simon's Fired
Ontario State Festival
Overpaid Semiprofessional Footballers
Ox and Sheep Fodder
Old Sock Factory
Oregon Stargazers' Fellowship
Omelette Sandwiches, Fried
Outlawed Sherriffs' Federation
Orange 'n' Strawberry Fudge
Orinoco Sailing Fleet
Overpoweringly Scented Females
Only Sundays in February
Other Specimens to Follow
Oasis South of Finland
Observe Several Flamingos
Occupied with Saving Face
Oboe Soloists' Fete
Off-Season Finances

Over-Sexed Freudians
Opticians' Semi-Final
Ordnance Survey Feud
Organic Stud Farm
Octagonal Summer Frock
Obligingly Simple Flirtation
Oriental Snorkellers Fighting
Oculists Swim Faster

I've tried not to reuse any words, so you can probably permute them as you see fit. Should be interesting...

If you are looking for some April Fool episodes, you would be doing pretty well to exceed this debauchery...

From: spock@numm.nu.oz (**Mr Spock**)
Newsgroups: aus.jokes
Subject: We had the funniest April Fools day !

The Cast: Myself
 Andrew : one of my flatmates
 Mariane: another one of my flatmates
 Megan : another one of my flatmates
 Joanne : another one of my flatmates

About 10:00 pm on March 31, Andrew and I decide to play a few little April Fool's Day pranks on our flatmates. Megan is at home but Mariane and Joanne are both out of the house. While Megan sits, watching TV, Andrew and I go to work. The following pranks ensued:

Megan :
Emptied bottle of skin cream, and replaced it with tomato sauce.
Put uncooked spaghetti in pillow.
Put vegemite on door handle.

(Megan gets off lightly, because she was home, and we couldn't do much without her knowing)

Joanne:
Put plastic wrap over toilet seat in ensuite.
Adjusted float in cistern so it would not re-fill after flushing, then flushed toilet, thus leaving it empty.
Replace skin cream with tomato sauce.
Put vegemite on door handles (ensuite and bedroom door).
Put uncooked spaghetti in pillow.
Poured ~100 hazelnuts into doona.
Unscrewed lid of toothpaste and put plastic wrap under screw cap (this has the effect of making the toothpaste look as though its coming out, but it is enclosed in an almost invisible plastic bubble - which makes it hard to put on the toothbrush.)

Mariane:

Put vegemite on door handle
Put uncooked spaghetti in pillow
Set up light with timer set to go off at 2:00 am
Set up disco strobe with stereo set to go off at 3:00 am
Put alarm clock under bed set to go off at 4:00 am
Put plastic wrap over toilet seat.
Rigged flushing mechanism as above.
Stuffed container of skin oil with cotton wool, so it was blocked up.
Unscrewed lid of toothpaste and stuffed with cotton wool.

Miscellaneous :

Filled empty tomato sauce bottle (I wonder why it was empty ?) with very hot tabasco sauce.
Opened up all the little cream biscuits in the biscuit barrel and replace the creamy stuff with peanut butter and vegemite.
Put vegemite on Andrew's and my door handles (explanation follows)
Put piles of clothes resembling sleeping person in Andrew's bed.

- Events begin to unfold -

Megan goes to bed around 11:30 and discovers vegemite the hard way. Unfortunately, she spots spaghetti in pillow, and avoids it. As yet she has not used skin cream, so we'll have to wait and see ...

Mariane comes home at ~12:15, finds vegemite (yuk). Not impressed with this, she enters hers room, and while washing her hands, notices the plastic wrap on the toilet seat. Armed with evidence (plastic wrap) and clean hands, Mariane bursts into Andrew's room (which is right across the hall) only to be vegemited again! She walks over and kicks what she thinks is Andrew but discovers clothes and pillows, and a football for a head. Not impressed She decides that it's time to call it quits and goes to bed. Hears ticking of clock and removes it from under her bed. Goes to clean teeth and finds that toothpaste won't come out of tube. Storms up the stairs to my room, where she finds yet more vegemite (she's very pissed off by now), enters and smears me with what's left of the vegemite. Satisfied that she has had the last laugh, Mariane goes back downstairs gets toothpaste, and goes looking for Andrew. Finds Andrew sleeping in spare room. Smears him with toothpaste. Large, loud fight breaks out, with both parties being smeared. Mariane demands that we tell her what else we have done to her room, so we go downstairs with her, and point out light set to go off at 2:00 am. We also own up to the spaghetti in the pillow, but do not mention disco strobe and stereo set to blast off at 3:00 am.

Andrew and I then pretend to go to bed in our own rooms, and then sneak out, and into the spare room, to crash out there. 3:00 comes, stereo blasts out, and lights come on. Seconds later, it gets switched off. Mariane does not emerge seeking retribution.

- The Day After -

Next morning I got up early and went into the uni to do a bit of work. Before I left, I put a tiny piece of paper in the crack of the door to

my room, so when I come home, I can see if my rooms has been tampered with, as paper will have fallen out. When I return at 10:00 pm that night, after being out all day, Andrew has gone to Sydney, and everyone else had gone to bed. I see that the paper is gone, so I know they have been in my room. Vowing to be very careful I open my door, and check out the light switch .. it looks ok, so I turn on the light. They have replaced the normal lightbulb with a blue one. This does not make it easy to see. "Why do they want to make it hard to see ?" - So I will switch on my desk lamp - ok - what have they done to my desk lamp. I have a closer, squinting look, and discover that there is suntan cream on the button. I use a tissue to turn it on. I then set about looking for all the nasty little things that they have done. First, I look under the bed - an obvious place to start - they haven't actually put anything under my bed, however, they removed the two legs at the bottom of the bed. (my bed has 6 legs, and will balance on only four, if 2 of the end ones are removed). I decide to go down stairs and wake Mariane, (as it turns out she wasn't asleep) and ask where the wheels to my bed are. She says that they are on the top shelf of my cupboard, so I go back upstairs (Mariane follows) and look in my cupboard. Not only do I find the wheels to my bed, but also a screwdriver .. ? As I ponder over the meaning of the screwdriver, Mariane pulls my door closed and pulls out the door handle - thus locking me in my room - SHIT ... how could I have been SO careless !?!? Very pissed off with myself for falling for such a nasty one, I begin to realise what the screwdriver is for. I decide that it will take too long to unscrew all the little screws, and that it would be quicker to use my swiss army knife to pop the hinges out. After having done this, I sit down for about 15 mins - fix my door, and decide that I will read some more a book I am in the middle of. I decide that a little snack would be nice, and I happened to have a packet of biscuits handy. I re-open the packet (I had already eaten some of them) and eat the first one .. it tastes of mint .. ? These are supposed to be shortbread creams .. ? Wait a sec .. I have closer look and the last four biscuits in the packet have had the cream replaced with toothpaste. I was lucky in that the toothpaste made the biscuits soggy, and thus I avoided eating them before it was too late.

Deciding that I had found just about all the little pranks that were likely to disturb my sleep, I turned off the light, and as I did, i remembered my clock/radio. Deciding to check the alarm - sure enough it is set to go off at 3:30 am. I switch it off and go to sleep.

When I get up in the morning, I talk to Megan and Mariane and ask them if it was just them that 'did' my room. They tell me that Andrew helped them, and then I tell them that Andrew helped me 'do' their rooms'. We decide to HAMMER Andrew !

Andrew rides a push bike to Uni, so the first thing we do is take the seat off .. that should be nice and uncomfortable ! Next, we move everything in his room to a little storage shed that we have at the back of the house, and then lock it leaving a note that the key for the shed is in the girls toilets in the Uni union building. (By the time he gets back from Sydney - the key WILL be there). We then get the ~1500 - 2000 old newspapers that are piled up in our garage (the whole lot takes up about as much rooms as a small

car) and page by page, crumple them up, until there are enough crumpled pages to fill Andrew's room to the roof. Let me tell you, it was no easy job .. that's a LOT of newspaper. (about 6 hours of paper crumpling)

Andrew is still in Sydney and I Joanne hasn't returned from wherever it is she went for the weekend, so I think that there are still a lot of laughs to be had at our place. If I get time, I shall send word of what happens to Joanne and Andrew when they come home to get their 'prezzies'.

Mr Spock.

Sigh, I never did see a followup to the above... Oh well. Anyway, here's something from the evergreen aus.bizarre:

From: zvs@bby.oz (Zev Sero)
Newsgroups: aus.bizarre
Subject: from sci.space

Today the Discovery Orbiter is scheduled to arrive at Kennedy Space Center at 5pm. Discovery, that just returned from its five day mission last week is reported to be in excellent condition. Although technician are checking out two small leeks that occurred in Discovery's main engine no. 1.

Shades of Hitchhikers' Guide to the Galaxy....

Zev

And now, to wrap it all up, what better than the infamous Mark V Shaney:

From: mvs@softway.oz (Mark V. Shaney)
Newsgroups: aus.flame.mackin
Subject: Self preservation?

Best to stay alive. Or is this mackin man? Or is this just normal self preservation?

May I be so bold as to suggest that the only way this group is ever going to survive is if there is a demonstratable need for this group. And the private mailing list traffic, although significant, will not save this group is ever going to be overrun by althetes and spectators. It's bad enough in the process!!

And that was just in the process!!

In the search for "meaningful" relationships, pretense is a rule about the fact you're just re posting mostly someone elses words, and trimmed the quotes down, rather than proving you can do better than John Mackin who has the net in mind. If the net is your own private property, I'll be delighted to f k off. In fact, I can get it to D D. In the search

for "meaningful" relationships, pretense is a change from all the events? Brisbane had the olympic games back in the process!!

And that was just in the worst when Richardsons' were playing an imitation of Whitney Houston being strangled at least four times, preferably by Tom Osborn, before posting.

May I be so bold as to suggest that the only way this group is ever going to survive is if there is a change from all the events? Brisbane had the olympic games back in the first fleet landed in Sydney in 1788.... So what!!!

0. You have misspelt Melbourne.

Who cares what he wrote this is just a bloody cyclist cuts across in front of me, goes to the chemist standing nearby. Much vegetable matter thrown onto the center of the bicentennial crap.

Who is mackin? Why does he do what he does? And is he any meaningful relation to the process?

--

-----Mark

Cheers to my wonderful audience (the whole two of you), and we'll catch you in aus.bizarre some time...

--

Dave Horsfall (VK2KFU), Alcatel STC Australia, dave@stcns3.stc.oz
dave%stcns3.stc.oz.AU@uunet.UU.NET, ...munnari!stcns3.stc.oz.AU!dave

:login:

The USENIX Association Newsletter

Volume 14, Number 2

March/April 1989

CONTENTS

UNIX Transaction Processing Workshop Program	3
Call for Papers: Large Installation Systems Administration Workshop	4
Call for Papers: Distributed and Multiprocessor Systems Workshop	5
Call for Papers: Computer Graphics Workshop	6
Call for Papers: Malaysian UNIX Users Association Meeting	7
Call for Papers: Workstation Operating Systems Workshop	8
Book Review: C Traps and Pitfalls	9
<i>Marc D. Donner</i>	
C++ Software Tape	10
Summary of the Board of Directors' Meeting	11
USENIX Online Index and Library	12
FaceSaver does UUNET	14
The Netherlands UUG	14
USENIX 1990 Winter Conference	15
Future Events	16
Publications Available	16
Long-Term Calendar of UNIX Events	17
New Release of 2.10 BSD Available	18
Proceedings Subscription Form	19
4.3BSD Manuals	20
4.3BSD Manual Reproduction Authorization and Order Form	21
Local User Groups	22

The closing date for submissions for the next issue of *:login:* is April 28, 1989

 THE PROFESSIONAL AND TECHNICAL
UNIX® ASSOCIATION

Call for Papers
Workshop on
Large Installation Systems Administration III
September 7-8, 1989
Austin, TX

In light of increasing attendance at the workshops on Large Installation Systems Administration, plus contributions from various places (academia to standards efforts to commercial development work on system administration tools), the third annual workshop on this subject will again be chaired by Alix Vasilatos in Austin, TX on Thursday and Friday, September 7th and 8th, 1989.

For the first time, a program committee will be reviewing submitted papers on subjects including, but not limited to:

- Automating systems administration chores
- Network management
- Distributed services
- Backup and archiving
- Spooling/Queuing schemes
- Configuration synchronization
- Accounting/Passwd maintenance
- Electronic communication (USENET/News/Notes/Mail)
- Software distribution
- Reliability enhancement

Papers can range in length from "very short" (1-3 pages) to not over 14 pages including all figures, diagrams, etc. Complete papers only, please, even if short. Abstracts and outlines will not be accepted. Include a description of the unique characteristics of the site, an outline of the problem, and a description of the solution (detailed enough that fellow administrators can implement it). Workshop proceedings will be available at the workshop.

The deadline for submissions is **June 30, 1989**. For further details about the workshop, contact:

Alix Vasilatos
Open Software Foundation
11 Cambridge Center
Cambridge, MA 02142

alix@osf.org
uunet!osf.org!alix
617-621-8722

For details about registration, contact the USENIX Conference Office.

Call for Participation

Workshop on Experiences with Building Distributed (and Multiprocessor) Systems[†]

October 5-6, 1989
Ft. Lauderdale, FL

Goals: The goal of this workshop is to bring together individuals who have built, are building, or will soon build distributed and multiprocessor systems, especially operating systems. The workshop will feature full presentations, panels, work-in-progress presentations, and possibly tutorials on aspects of building and testing these systems. The workshop will provide a forum for individuals to exchange information on their experiences, both good and bad, in designing, building, and testing their systems. This includes experiences with coding aids, languages, distributed debugging tools, prototyping, reuse of existing software, performance analysis, and lessons learned from use of such systems.

Submissions: *Ten* copies of each submission should be mailed to the program committee chair (address below) no later than **1 June 1989**. Submissions are invited on any topics related to the workshop in the form of papers or extended abstracts, although the program committee will give preference to full papers. Furthermore, the committee will give preference to submissions describing experiences with actual systems.

Panels and Tutorials: Suggestions for panel sessions and tutorials, including recommended participants, should be mailed to the program chair no later than **1 May 1989**. Such submissions should include a description of the relevance to the goals of the workshop, and the qualifications of the participants suggested.

Important Dates

Panel & Tutorial proposals	1 May 1989
Paper submissions	1 June 1989
Program Committee decisions	14 July 1989
Camera ready copy due	1 August 1989
Workshop	5-6 October 1989

For Further Information, contact:

<i>General Chair</i>	<i>Program Chair</i>
George Leach	Gene Spafford
Paradyne Corporation	Software Engineering Research Center
MS LG-129	Dept. of Computer Sciences
PO Box 2826	Purdue University
Largo, FL 34649-2826	W. Lafayette, IN 47907-2004
(813) 530-2376	(317) 494-7825
reggie@pdn.nm.paradyne.com	spaf@cs.purdue.edu

[†] Sponsored by The USENIX Association;
in association with the NSF / Purdue / Florida Software Engineering Research Center;
in cooperation with ACM SIGOPS and SIGSOFT (pending) and IEEE-CS Technical Committee on Distributed Processing
(pending).

Call for Participation
5th USENIX Computer Graphics Workshop
November 16-17, 1989
Doubletree Hotel – Monterey Convention Center
Monterey, CA

The 5th USENIX Computer Graphics Workshop will be held in Monterey on Thursday and Friday, November 16-17, 1989. The theme of the workshop is “personal graphics.” By this, we mean the use of computer graphics to aid, benefit, or amuse a single person. Generally, personal graphics applications are highly interactive, so that the user has a great deal of control over the result. Furthermore, the graphics is frequently not an end product, but is instead a communication medium between the user and computer. Examples of personal graphics might include desk-top publishing, data visualization programs (e.g., MacSpin), windowing systems, micro-world simulations (Kay’s vivarium?), and “performance” graphics (e.g., video weirdness). It probably does not include ray-tracing, yet another VLSI graphics chip, or fast rendering algorithms. A distinguishing feature is that the user is included as an integral part of the description of the system.

Personal graphics is becoming increasingly important as the cost of high-performance computing gets lower and lower, so that “Joe Public” is exposed to better and faster graphics than was previously possible. You only have to look at the progression from PC to Macintosh to NeXT to see this trend. One question that could be addressed by presentations in this workshop is “How are ‘ordinary people’ going to effectively use computer graphics in their daily lives?”

In addition to traditional full-length papers, we will entertain proposals for more informal short presentations and panel discussions. Submissions for the workshop will be judged on their intrinsic merit, but preference will be given to those which bear closely on the theme.

Extended abstracts (300-700 words) or position papers should be submitted by **July 3, 1989**. Authors will be notified of acceptance by **July 17, 1989**. Materials for inclusion in the proceedings should be submitted by **October 2, 1989**. Submissions may be made electronically or as hard-copy. Electronic submissions should be plain text, *troff* (-ms or -me preferred, do not use -mm), *LaTeX*, or PostScript. Mail electronic submissions to *usenix-graphics@crim.eecs.umich.edu*, or send hard-copy to the workshop chair

Spencer W. Thomas
EECS Department
University of Michigan
1101 Beal Ave.
Ann Arbor, MI 48109-2110
(313) 936-2616

Call for Papers
International Seminar
on
Current Developments
and
Future Trends of UNIX-based Systems

12-13 September, 1989

organised by
Malaysian UNIX Users Association (MALNIX)
in cooperation with
Malaysian Institute of Microelectronic Systems (MIMOS)

Objectives

To provide a forum for discussion and exchange of information, ideas and experiences on current developments and future trends of UNIX-based systems.

To promote research activities related to UNIX in Malaysia and the region.

Seminar Topics

Parallel Processing	Office Automation
Real-time Applications	Database Systems
Software Development Tools	Networking
Emerging Standards	UNIX in Commercial Environments
UNIX on PC	User Interfaces
System Security	

Submission of Papers

A copy of an extended abstract, limited to about 2000 words, should reach the Organiser by **April 3, 1989**. Notification of acceptance will be sent to the authors by May 22, 1989. The full paper shall be sent, by August 21, 1989, to:

The Organiser
MALNIX/MIMOS International Seminar
7th Floor, Bukit Naga Complex
Jalan Semantan
50490 Kuala Lumpur, MALAYSIA

Phone: (60) 3 2552 700 Fax: (60) 3 2552 755
E-mail: malnix@rangkom.my or uunet!mimos!malnix

CALL FOR PARTICIPATION

SECOND WORKSHOP ON WORKSTATION OPERATING SYSTEMS

Sponsored by:



IEEE Computer Society - Technical Committee on Operating Systems

Asilomar Conference Center, Pacific Grove, CA.

September 27-29, 1989

The Second Workshop on Workstation Operating Systems (WWOS-II) will bring together a number of researchers and developers to discuss their work and experiences in workstation operating systems. As workstation technology and man-machine interfaces advance, the operating system must evolve. Potential topic areas include:

- New workstation operating systems
- OS changes for the workstation environment (e.g. graphical interfaces)
- Scheduling and resource management
- Workstation architectures
- Multi-processor considerations

To facilitate the dialog that is so valuable in a workshop setting, attendance will be limited to 60 active workers in the field.

Registration requests should be addressed to Joseph Boykin, the General Chair. This request should include a 1-3 page position paper describing the participant's interest and future directions related to workstations

and workstation operating systems as well as insights and lessons gained from recent research and practical experience.

Those interested in presenting the results of their research, or current work-in-progress, should submit a 3-5 page paper to Luis-Felipe Cabrera. Papers must be submitted by April 15, 1989. Acceptance will be made by June 1 and final copy is due by July 1, 1989. Selection of papers will be made by the program committee:

David Anderson, *UC Berkeley*
Kenneth Birman, *Cornell University*
Anita Borg, *DEC WRL*
Luis-Felipe Cabrera, *IBM (chair)*
Sam Chanson, *U. of British Columbia*
Robert Hagmann, *Xerox Parc*
Paul Leach, *Apollo Computers*
Andy Hisgen, *DEC SRC*
Sape Mullender, *CWI, Amsterdam*
Ike Nassi, *Encore Computer*
Mike Powell, *SUN Microsystems*
David Presotto, *AT&T Bell Laboratories*
Bud Tribble, *Next Computers*
John Wilkes, *Hewlett Packard*

Registration, including a copy of the proceedings: Computer Society Member	Pre-registration	On-site
' Non-member	\$160	\$215
Student Member	\$210	\$275
Lodging:	\$95	\$125

Lodging: \$55 per night for double occupancy room, breakfast, lunch, and dinner.

General Chairman: Joseph Boykin
Encore Computer Corp., 257 Cedar Hill Street
Marlborough, MA 01752
(508) 460-0500 x2720
ARPA: boykin@encore.com
UUCP: encore!boykin

Program Chairman: Luis Felipe Cabrera
IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120
(408) 927-1838
ARPA: cabrera@ibm.com
UUCP: ucbvax!cabrera

Book Review: C Traps and Pitfalls

by Andrew Koenig

(New York: Addison-Wesley) 147 Pages, \$16.25

Reviewed by Marc D. Donner

IBM Thomas J. Watson Research Center
ibm.com!donner

C Traps and Pitfalls is another book, like the *C Programmer's Handbook* by Bolsky, that achieves excellence by picking one problem and solving it well. Koenig seeks to exhibit a number of programming errors that are characteristic of C, explain how they occur, what their symptoms are, and suggest ways to reduce their occurrence in programs. As such the text focuses almost exclusively on specific implementation examples and does not attempt to enunciate abstract principles, though several important principles are explained rather matter-of-factly. This isn't fair, but it might have been fun to title the book "C Puns and Anagrams."

One unique feature of this slim volume is that it has value both to experienced and novice C programmers. That a book not simply a reference work could have something to offer both to master programmers and to apprentices is rather unusual, since their needs are usually so disparate.

The text won't keep novices from making the errors it describes, but it will probably help them recognize the symptoms when they occur and reduce the time to identifying and resolving them.

For experienced C programmers this book will evoke a fond memory of pernicious bugs and clever solutions, and it might well teach something new. Even if it teaches nothing, experienced programmers will feel toward this book as Hobbits feel toward genealogies [J. R. R. Tolkien, *The Fellowship of the Ring*, Houghton Mifflin, Boston 1965, page 17]: "Hobbits delighted in such things, if they were accurate: they liked to have books filled with things they already knew, set out fair and square with no contradictions." However, I predict that it will be a rare C programmer,

however wizardly, who finds no treasure in this little book.

One particularly strong point in Koenig's treatment of some of C's peculiarities is his reference back to the principles practiced by the original designers. He has tracked down several of the oddities to specific problems entailed by the transition from B to C and a variety of issues that cropped up as C evolved (for example, the discussion of | and || on page 19). Whenever possible, he tries to provide some suggestive mnemonic or story that will help the reader remember rather than just explaining.

The book breaks mistakes down into several classes. Koenig treats the syntactic pitfalls early and relatively briefly. He then goes into considerably greater length on what he calls semantic pitfalls but which might better be referred as primitive semantic pitfalls.

Subsequent sections explore progressively more subtle semantic problems. He recognizes that the boundary between oddities and problems with the language and oddities and problems with program portability is a fine, and devotes a chapter to a brief exploration of portability issues. Most of this treatment is based on a specific example, whose behavior and misbehavior Koenig subjects to a thorough analysis.

The section on the preprocessor might be supplemented or replaced by the "C Puzzle Book" or some other more detailed treatment, but it is quite adequate for this volume.

There are digressions into ANSI C issues from time to time, but the book is quite comprehensible to non-ANSI C hackers and in fact is probably most relevant to mainstream C programmers.

The only set of issues that might have been treated but were not are some of the old-*pcc*-peculiarities versus modern-*cc* features that have translated themselves into unexamined stylistic practices in the C programming community. An example of this is the practice, still widespread, of creating a type-suggestive three-character prefix for the names of fields in a structure. An example is the *timeval* structure returned by *gettimeofday()*:

```
struct timeval {
    /* seconds since Jan. 1, 1970 */
    long tv_sec;
    /* and microseconds */
    long tv_usec;
};
```

The convention that makes us put the *tv_* in front of each field is a defensive practice from an old version of the portable C compiler that suffered from the symbol table defect that each field name in a program had to be unique. This requirement is gone from all but the most backward of C compilers, but its legacy is so deeply established that we have almost forgotten the reason for it.

This book can be read profitably by the novice C programmer annually for the first five years, and then kept around to reread annually by the advanced programmer for spiritual renewal. As with many other excellent things, its excellence stems from its simple and direct message, unmuddled by ambitious and self-important pomposity.

C++ Software Tape

The 1988 USENIX software tape contained C++ software. It has been sent to all 1988 Institutional and Supporting members of the Association.

Individual members of USENIX who wish to obtain a copy of the tape may request it from the Association Office. It requires no AT&T nor UC license. You will be sent a requestor "Tape Release Form." The completed form plus a check for \$125 should be returned to the Association, whereupon the tape will be sent out, postpaid in the US. Foreign individuals will be billed for the additional (air) postage.

The tape, in *tar* format at 1600 BPI, contains:

- GNU C++ Version 1.21.0 (Michael Tiemann)
- OOPS (Keith E. Gorlen)
- Storage management class and String class... (Peter A. Buhr)
- InterViews 2.3 (Mark A. Linton)
- C++ Subroutines for string manipulation (Arthur Zemon)

The tape is not available to non-members of the USENIX Association.

NOTICE: Some copies of the C++ tape are faulty. If you cannot read your tape, return it to the USENIX Office.

USENIX Online Index and Library

Deborah Scherrer

As the quantity of USENIX and USENIX-related publications rapidly expands, and the quality becoming increasingly attractive, the need has arisen for some sort of online access, at least to an index of published articles. USENIX has authorized a proposal to fund, implement, and make available an electronically accessible index to USENIX and related groups published articles. Versions of abstracts and/or full papers will also be made available in cases where authors are willing to donate them. Articles to be referenced will be drawn from (past and future):

	Entries (approx)
USENIX	
Conference proceedings	550
Workshop proceedings	150
<i>Computing Systems</i>	8
Newsletters	100
EUUG	
Conference proceedings	200
Newsletters	150
Software Tools Users Group	
Conference proceedings (held jointly with USENIX)	40
AUUG	
Conference proceedings	?
Newsletters	?
JUS	
Conference proceedings (yes, in Japanese)	?
TOTAL (approx)	1300

Other sources (AFUU, GUUG, NZUSUGI, etc.) will be continually evaluated and included as deemed suitable.

The index will include references from as many past issues of the publications as can be obtained, and kept current as new issues are released. In addition, authors of USENIX publications are being given the opportunity to donate their full paper, or abstract only, for inclusion and distribution through the online library.

The online index and library will reside at UUNET and be freely available. Users will be

able to retrieve both the index and any available online abstracts or papers, individually, by simple email request. Specific details on retrieval will be forthcoming. Although the library and index will be most effectively used electronically, the index will also be printed periodically and made available through the USENIX office, for those without email access.

If You Have Ever Had a Paper Published by USENIX: Donation of either a paper or an abstract to the library is purely voluntary. Papers can be donated and distributed either with a copyright notice, or a copyright release, at the copyright holder's discretion. An appropriate release form, festooned with the appropriate signatures, is absolutely required before any paper or abstract will be made available electronically. Authors wishing to donate papers or abstracts need to

1. Assure themselves (and us) that they own the copyright.
2. Sign an **Author Release Form**.
3. Send an electronic version of the paper to index@usenix.org or uunet!usenix!index.

Note that the issue of copyright ownership should be investigated thoroughly by the donor – multiple authors may share in copyright, or the author's institutions or funders may hold them. USENIX does not have the resources to research copyrights and will rely on the information you provide us on the signed release. The electronic papers and abstracts will be made available for retrieval in essentially the same format they were delivered – no attempt will be made to translate into a common text formatting language, etc. (Understand that some illustrations will not be available electronically and some people may not be able to properly print papers they request.)

Authors may choose to donate abstracts only, even in cases where a full paper has been published. Authors donating abstracts only also need to sign the form and email a copy of the abstract, if they have one. If the abstract

is not available electronically, and the author(s) gives permission for publication, USENIX will key in a copy from the publication itself (assuming the abstract had been published). USENIX does NOT have sufficient resources to key in full papers.

Please check your memories for any papers you may have had published by USENIX or related organizations, and sign the attached form giving permission for us to make available at least the abstract, and the paper if you wish. Note that if you do give us permission to make available an entire article, you will have to send us an electronic copy of that article.

FaceSaver does UUNET

The USENIX database of faces from the Phoenix, Dallas and San Francisco conferences is being installed on UUNET for access by one and all. All attendees at these three conferences whose faces were saved have been sent notices regarding their portraits and the permission which they gave or denied for such a distribution. Unless this notice was returned by the post office as undeliverable, all portraits for which we have affirmative releases are being sent to the database.

The portraits are either 108 (wide) \times 128 (high) \times 8 bits (non-square pixels) or 96 (wide) \times 128 (high) \times 8 bits (square pixels), depending on the conference at which they were acquired. There is a fixed length header with name, email address, and pixel format, followed by the raw data.

Several simple programs to deal with the images are also available, and it is hoped that others will be donated.

Access to the portraits will be available via electronic mail. Send mail to *uunet!faceserver* (yes, that's sErver, not sAver) with the subject line of "help" or "send *email-address*" to retrieve the face. For example, typing

```
echo send rick@uunet.uu.net | \
mail uunet!faceserver
```

would get Rick Adam's picture returned to you.

If you are uncertain about the status of your portrait, or need to correct the data accompanying it, send e-mail to *uunet!usenix!facesaver*.

Lou Katz, Saver of Lost Faces

Future Events

UNIX Transaction Processing Workshop, Pittsburgh, May 1-2, 1989

See page 3.

USENIX 1989 Summer Conference and Exhibition, Baltimore, Jun. 12-16, 1989

Large Installation Systems Administration Workshop, Austin, Sept. 7-8, 1989

See page 4.

Distributed Processing Workshop, Fort Lauderdale, Oct. 5-6, 1989

See page 5.

Graphics Workshop V, Monterey, Nov. 16-17, 1989

See page 6.

Long-term USENIX & EUUG Schedule

Sep 18-22 '89 Vienna, Austria
Jan 22-26 '90 Omni Shoreham, Washington, DC
Apr 23-27 '90 Munich, W. Germany
Jun 11-15 '90 Marriott Hotel, Anaheim
Jan 21-25 '91 Grand Kempinski, Dallas
Jun 10-14 '91 Opryland, Nashville
Jan 20-24 '92 Hilton Square, San Francisco
Jun 8-12 '92 Marriott, San Antonio

Publications Available

The following publications are available from the Association Office. Prices and overseas postage charges are per copy. California residents please add applicable sales tax. Payment may be made by check, VISA or MasterCard and must be enclosed with the order.

Conference and Workshop Proceedings

Meeting	Location	Date	Price	Overseas Air
USENIX Conference	San Diego	Jan. '89	30	25
C++ Conference	Denver	Oct. '88	30	20
UNIX and Supercomputers Workshop	Pittsburgh	Sep. '88	20	15
USENIX Conference	San Francisco	Jun. '88	20	20
C++ Workshop	Santa Fe	Nov. '87	30	20
Graphics Workshop IV	Cambridge	Oct. '87	10	15
USENIX Conference	Washington DC	Jan. '87	10	20
Graphics Workshop III	Monterey	Dec. '86	10	15

EUUG Proceedings for Spring 1988 (London) and Fall 1988 (Portugal) are available in limited numbers to North American customers at \$40 per copy.

Long-Term Calendar of UNIX Events[†]

1989 Apr 24-28	IEEE 1003	Minneapolis-St. Paul, MN
1989 May 1-2	* Transaction Processing Workshop	Pittsburgh Hilton, Pittsburgh, PA
1989 May 8-12	DECUS	Atlanta, GA
1989 May 14-16	AMIX	Israel
1989 May 16	POSIX Application Workshop	NIST; MD
1989 May	UNIX 8x/ctc	/usr/group/cdn; Toronto, Ont.
1989 Jun	NZSUGI	New Zealand
1989 Jun 12-16	USENIX	Hyatt Regency, Baltimore, MD
1989 Jul	JUS 13	Toyko, Japan
1989 Jul 10-14	IEEE 1003	San Francisco, CA
1989 Sep 7-8	* Large Systems Admin. Workshop	Austin Marriot, Austin, TX
1989 Sep 12-13	MALNIX	Kuala Lampur, Malaysia
1989 Sep 18-22	EUUG	Vienna, Austria
1989 Sep 27-29	Workstation Operating Systems	Pacific Grove, CA
1989 Oct 5-6	* Distributed Systems Workshop	Marriott Marina, Ft. Lauderdale, FL
1989 Oct 16-20	IEEE 1003	Brussels (or Amsterdam)?
1989 Nov 1-3	UNIX Expo	New York, NY
1989 Nov 6-10	DECUS	Anaheim, CA
1989 Nov 16-17	* Graphics Workshop V	DoubleTree Hotel, Monterey, CA
1989 Nov	JUS 14	Osaka or Kobe, Japan
1989 Dec	JUS UNIX Fair	Toyko, Japan
1990 Jan 22-26	USENIX	Omni Shoreham, Washington, DC
1990 Jan 23-26	UniForum	Washington Hilton, Washington, DC
1990 Jan 29	IEEE 1003	New Orleans, LA
1990 Feb	UNIX in Government	Ottawa, Ont.
1990 Apr	IEEE 1003	Montreal, Que.
1990 Apr 23-27	EUUG	Munich, Germany (tentative)
1990 May 7-11	DECUS	New Orleans, LA
1990 May	UNIX 8x/etc	/usr/group/cdn; Toronto, Ont.
1990 Jun 11-15	USENIX	Marriott Hotel, Anaheim, CA
1990 Autumn	EUUG	south of France
1991 Jan 21-25	USENIX	Grand Kempinski, Dallas, TX
1991 Jan 22-25	UniForum	Infomart, Dallas, TX
1991 Jun 10-14	USENIX	Opryland, Nashville, TN

[†] Partly plagiarized from John S. Quarterman by PHS.

* USENIX Workshops

New Release of 2.10 BSD Available

The second release of 2.10BSD is finally available! It has been designated 2.10.1. Although the changes are fairly simple to describe, they cover large portions of the distribution. Most will not be visible to either users or administrators; specifically, no recompilation is necessary. Administrators should be aware that the 4.3BSD disk quota system is now available. Due to address space considerations, however, it is expensive to run. Also, the source for the on-line manual pages has been rearranged as per the 4.3BSD-tahoe release.

The major change, and the reason for the second release, is an extensive reworking of the kernel to move the networking into supervisor space. This move eliminated most, if not all, of the instabilities seen in the original networking provided with 2.10BSD; it also doubled the speed of, for example, file transfer. As encouragement to sites that encountered difficulties in using the networking in the first release, or encounter difficulties in this release, we have beta sites that have been running for months without crashing, as well as sites with fifty nodes. We are, however, still suspicious of the DEQNA driver...

In application land, many missing pieces of the 4BSD distribution have been added, most notably the FORTRAN compiler and library and the line printer sub-system. Many other programs have had minor (and not-so-minor) fixes applied.

Keith Bostic
Casey Leedom

Because the changes to the kernel are major, no "upgrade" tape will be available. 2.10.1 BSD is only available as source, to appropriate licensees of V7, System III, System V, or 2.9BSD. The cost is \$200, prepaid.

The release consists of two 2400 foot, 1600 BPI tapes (approximately 80Mb) and approximately 100 pages of documentation. If you require 800 BPI tapes, please contact USENIX for more information.

If you have questions about the distribution of the release, please contact USENIX at:

2.10BSD
USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710

+1 415 528-8649
{uunet,ucbvax}!usenix!office

If you have technical questions about the release, please contact Keith Bostic at:

{ucbvax,scismo}!keith
keith@okeeffe.berkeley.edu

+1 415 642-4948

NOTE: There are a few copies of 2.9BSD available. If you do not have split I&D and want to run UNIX on your PDP-11/x, write the USENIX office.

— PHS

4.3BSD Manuals

The USENIX Association now offers all members of the Association the opportunity to purchase 4.3BSD manuals.[†]

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

A new Master Index has been created. It contains cross-references to all documents and manual pages contained within the other six volumes. The index was prepared with the aid of an "intelligent" automated indexing program from Thinking Machines Corp. along with considerable human intervention from

Mark Seiden. Key words, phrases and concepts are referenced by abbreviated document name and page number.

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

The costs shown below do not include applicable taxes or handling and shipping from the printer in New Jersey, which will depend on the quantity ordered and the distance shipped. Those charges will be billed by the printer (Howard Press).

Manuals are available now. To order, return a completed "4.3BSD Manual Reproduction Authorization and Order Form" to the USENIX office along with a check or purchase order for the cost of the manuals. You must be a USENIX Association member. Checks and purchase orders should be made out to "Howard Press." The manuals will be shipped to you directly by the printer.

Manual	Cost*
User's Manual Set (3 volumes)	\$25.00/set
User's Reference Manual	
User's Supplementary Documents	
Master Index	
Programmer's Manual Set (3 volumes)	\$25.00/set
Programmer's Reference Manual	
Programmer's Supplementary Documents, Volume 1	
Programmer's Supplementary Documents, Volume 2	
System Manager's Manual (1 volume)	\$10.00

* Not including postage and handling or applicable taxes.

4.2BSD Manuals are No Longer Available

[†] Tom Ferrin of the University of California at San Francisco, a former member of the Board of Directors of the USENIX Association, has overseen the production of the 4.2 and 4.3BSD manuals.

**EUROPEAN
UNIX® SYSTEMS USER GROUP
NEWSLETTER**

EUUG

*Volume 9, Number 2
Summer*

Editorial	1
An extension to the troff character set for Europe	2
Mk'ing Hardware: A Tutorial	10
A European Representation for ISO C	15
Alternatives to OSI	19
Brussels conference report	22
News from the Netherlands	27
EUUG Executive Report	29
Report from the Danish Group	30
UKUUG News	31
UKUUG UNIX Security Workshop	33
Networking Column	37
Puzzle Corner	40
UNIX CLINIC (<i>aka</i> Call Doc Strange)	41
Standards Report on C and POSIX	49
UNIX Standards: Repertory, Reps and Representation	51
Software Review	53
USENIX Association News for EUUG Members	58
UNIX System V Data Networking Architecture	60
Window Systems: X/ST/window	70
Book Reviews	79
AUUG—A Call for Papers	81
Calendar of UNIX Events	82
Glossary	85
UKUUG Summer Abstracts	86

An extension to the troff character set for Europe



E.G. Keizer
K.J. Simonsen
J. Akkerhuis

Vrije Universiteit, Amsterdam, The Netherlands
University of Copenhagen, Copenhagen, Denmark
C.M.U., USA

ABSTRACT

The typesetting program *troff* was originally written for formatting English text for the CAT 48 Typesetter. Its offspring is used for formatting a variety of languages with a large diversity of output devices. The authors agreed on an addition to the *troff* character set covering old and new national and international latin-based character sets.

The problems

When adapting the UNIX[†] typesetting program *troff*¹ to a new output device, one wants to have access to the extra characters offered by the device, without sacrificing any characters already in use. Device-independent *troff*, also called *titroff* or *ditroff*² uses a flexible font definition mechanism that allows addition and deletion of characters.

Many people, including the authors, have used this mechanism³ to add characters. This has led to a diversity of names, with the expected conflicts of using the same name for different characters and different names for the same character in different implementations. Thus *troff* input files are becoming less and less portable, even for the same output device on different installations. We regret this development and, during a conference in Copenhagen, we decided to make an attempt at some standardisation. B.W. Kernighan, author of *ditroff*, agreed to our proposal of acting as a clearing house for our new names and he still has to give his blessing to this article.

We realise that it is impossible to name every printable character in the world. The total amount of different characters is simply too huge. Naming all the hundred different turtles in a turtle font is both frustrating and futile. We restricted ourselves to the following categories:

- characters belonging to the printed language of several Western-European countries: Æ ß
- variations of letters, especially with accents: ï û ö
- often used mathematical symbols: ∙ √ ⊗

Natural language support

The *troff* character set is based on the US-ASCII standard. This standard is well suited to English text, but causes problems when used in most European countries. US-ASCII is a version of the ISO 646-1983 standard. The ISO standard states the characters used in 7-bit ASCII and contains 94 printable graphic symbols. ISO-646 allows national versions for 12 of its character positions. The European Computer Manufacturers' Association ECMA is registering all different national versions of ISO 646-1983 and assigns

[†] UNIX is a trademark of Bell Laboratories.

a different character to each. This allows the creation of documents with multiple character sets. The assigned character serves to identify each character set in such documents. The table below shows several versions conforming to ISO 646.

National ISO 646 character sets														
Country	Standard	.la parameter												
ISO	ISO 646 IRV	ISO	#	¤	@	[\]	^	'	{		}	-
USA	X3.4-1968	US	#	\$	@	[\]	^	'	{		}	~
Great Britain	BS 4730	GB	£	\$	@	[\]	^	'	{		}	-
Japan	JIS C 6229	JP	#	\$	@	[¥]	^	'	{		}	-
China	GB 1988-80	CN	#	¥	@	[\]	^	'	{		}	-
Denmark	DS 2089	DK	#	\$	@	Æ	Ø	Å	^	'	æ	ø	å	~
Norway	NS 4551-1	NO	#	\$	@	Æ	Ø	Å	^	'	æ	ø	å	-
	NS 4551-2	NO2	§	\$	@	Æ	Ø	Å	^	'	æ	ø	å	
Finland	FI	#	\$	@	Ä	Ö	Å	^	'	ä	ö	å	~	
Sweden	SEN 850200 B	SE	#	¤	@	Ä	Ö	Å	Ü	'	ä	ö	å	-
	SEN 850200 C	SE2	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
Germany	DIN 66 003	DE	#	\$	§	Ä	Ö	Ü	^	'	ä	ö	ü	ß
Hungary	MSZ 7795/3	HU	#	¤	Á	É	Ö	Ü	^	á	é	ö	ü	"
France	NF Z 62-010	FR	£	\$	à	°	ç	§	^	μ	é	ù	è	"
Italy	IT	£	\$	§	°	ç	é	^	ù	à	ò	è	ì	
Spain	ES	£	\$	§	í	Ñ	ñ	^	'	ó	ñ	ç	~	
	ES2	#	\$	•	í	Ñ	ç	ñ	'	’	ñ	ç	..	
Portugal	PT	#	\$	§	Ã	Ç	Ó	^	'	ã	ç	õ	º	
	PT2	#	\$	'	Ã	Ç	Ó	^	'	ã	ç	õ	~	

Terminals in these countries are often adapted to these national variations. Creating *troff* input for Danish texts on Danish terminals is a frustrating experience. One has to type \(\AE\) for Æ in spite of the presence of a special key for Æ. You can by-pass this by using the *troff* command .tr, which allows the mapping of any character to any other. We have employed the .tr command in a macro .la which is designed to make it possible to shift between all the ISO 646 input character sets in the above table. The macro takes a code for the country as parameter; the first two letters being the ISO 3166 two-letter country code. The .la macro can be used like this:

```
.la US
First we write something in "God's own" character set.
.la DK
S} skriver vi noget s|dt p} dansk: sodavandsis.
.la DE
Und f|r Deutschen k|nnen wir auch etwas schreiben!
.la US
```

giving:

```
First we write something in "God's own" character set.
Så skriver vi noget såd på dansk: sodavandsis.
Und für Deutschen können wir auch etwas schreiben!
```

The .la macro can only be used when all the characters of the character sets in use have a unique code on the printing device. Also you cannot change input character set within a diversion in *troff*, you need to use the special character names if you want to use foreign characters within a sentence. An example of having a French name in a Danish text:

Jeg s} Jer\(^ome Fran\(^, cois komme til K|benhavn.

giving:

Jeg så Jerôme François komme til København.

To be able to use all other national characters within a national character set, we decided to introduce names for all the different national characters, even for the 'default' US names.

Also we went through the new ISO standards for Latin alphabets (ISO 8859) and assured that all special characters there would have a unique name according to this proposal.

A last warning is about the *troff* escape character \ and the national characters taking its place. Here you must write the national character followed by an 'e' to get the desired result.

The .la macro has the following contents:

```
.de la
. \" languages - keld@dkuug.dk & storm@dkuug.dk
. \" Covers all ECMA registered versions of ISO 646
. \" Countries according to ISO 3166
. \" Commented is registered ECMA char code and standard number
.f1
.ie \\n(.S=0 .ds )L \\*(=L
.el .ds )L \\$1
.ds =L \\*(LA
.ds LA \\*()L
.rn )L
.if "\\*(LA"DK" .tr #\(\sh\$ \(\Do@ \(\at[\(AE\\\\\\\\(/O]\(\oA^ \(\ha\\\" \(\ga\\\" \(\ae\\\" \(/o)\(\oa^ \(\ti \\" DS 2089
.if "\\*(LA"US" .tr #\(\sh\$ \(\Do@ \(\at[\(IB\\\\\\\\(rs]\(\rB^ \(\ha\\\" \(\ga\\\" \(\IC|\(\ba)\(\rC^ \(\ti \\" B X3.4-1968
.if "\\*(LA"ISO" .tr #\(\sh\$ \(\Cs@ \(\at[\(IB\\\\\\\\(rs]\(\rB^ \(\ha\\\" \(\ga\\\" \(\IC|\(\ba)\(\rC^ \(\rn \\" @ IRV
.if "\\*(LA"GB" .tr #\(\Po$ \(\Do@ \(\at[\(IB\\\\\\\\(rs]\(\rB^ \(\ha\\\" \(\ga\\\" \(\IC|\(\ba)\(\rC^ \(\rn \\" A BS 4730
.if "\\*(LA"DE" .tr #\(\sh\$ \(\Do@ \(\sc[\(:A\\\\\\\\(:O]\(:U^ \(\ha\\\" \(\ga\\\" \(:a|\(:o)\(:u^ \(\ss \\" K DIN 66 003
.if "\\*(LA"FR" .tr #\(\Po$ \(\Do@ \(\a|\(\de\\\\\\\\(,c]\(\sc^ \(\ha\\\" \(\mu\(\('e|\(\u)\(\e^ \(\ad \\" f NF Z 62-010 (1982)
.if "\\*(LA"CN" .tr #\(\sh\$ \(\Ye@ \(\at[\(IB\\\\\\\\(rs]\(\rB^ \(\ha\\\" \(\ga\\\" \(\IC|\(\ba)\(\rC^ \(\rn \\" T GB 1988-80
.if "\\*(LA"JP" .tr #\(\sh\$ \(\Do@ \(\at[\(IB\\\\\\\\(Ye]\(\rB^ \(\ha\\\" \(\ga\\\" \(\IC|\(\ba)\(\rC^ \(\rn \\" n JIS C 6229-1984
.if "\\*(LA"IT" .tr #\(\Po$ \(\Do@ \(\sc[\(\de\\\\\\\\(,c]\(\e^ \(\ha\\\" \(\u\(\('a|\(\('o)\(\e^ \(\i \\" Y
.if "\\*(LA"ES" .tr #\(\Po$ \(\Do@ \(\sc[\(\r!\\\\\\\\(N]\(\r?^ \(\ha\\\" \(\ga\\\" \(\de\(\('n)\(\c^ \(\ti \\" Z
.if "\\*(LA"ES2" .tr #\(\sh\$ \(\Do@ \(\bu[\(\r!\\\\\\\\(N]\(\,C^ \(\r?^ \(\ha\\\" \(\ga\\\" \(\aa\(\('n)\(\,c^ \(\ad \\" h
.if "\\*(LA"PT" .tr #\(\sh\$ \(\Do@ \(\sc[\(\rA\\\\\\\\(,C]\(\o^ \(\ha\\\" \(\ga\\\" \(\aa\(\('a|\(\,c)\(\o^ \(\de \\" L
.if "\\*(LA"PT2" .tr #\(\sh\$ \(\Do@ \(\aa\(\('A\\\\\\\\(,C]\(\o^ \(\ha\\\" \(\ga\\\" \(\aa\(\('a|\(\,c)\(\o^ \(\ti \\" g
.if "\\*(LA"HU" .tr #\(\sh\$ \(\Cs@ \(\('A\(\('E\\\\\\\\(:O]\(:U^ \(\ha\\\" \(\a|\(\('e|\(\(:o)\(\:u^ \(\a^ \\" i MSZ 7795/3
.if "\\*(LA"NO" .tr #\(\sh\$ \(\Do@ \(\at[\(AE\\\\\\\\(/O]\(\oA^ \(\ha\\\" \(\ga\\\" \(\ae\\\" \(/o)\(\oa^ \(\rn \\" ' NS 4551 - 1
.if "\\*(LA"NO2" .tr #\(\sc$ \(\Do@ \(\at[\(AE\\\\\\\\(/O]\(\oA^ \(\ha\\\" \(\ga\\\" \(\ae\\\" \(/o)\(\oa^ \(\bv \\" a NS 4551 - 2
.if "\\*(LA"SE" .tr #\(\sh\$ \(\Cs@ \(\at[\(:A\\\\\\\\(:O]\(\oA^ \(\ha\\\" \(\ga\\\" \(\a|\(\(:o)\(\oa^ \(\rn \\" G SEN 850200 B
.if "\\*(LA"SE2" .tr #\(\sh\$ \(\Cs@ \(\('E\(\(:A\\\\\\\\(:O]\(\oA^ \(\(:U^ \(\('e|\(\(:a|\(\(:o)\(\oa^ \(\:u \\" H SEN 850200 C
.if "\\*(LA"FI" .tr #\(\sh\$ \(\Do@ \(\at[\(:A\\\\\\\\(:O]\(\oA^ \(\ha\\\" \(\ga\\\" \(\a|\(\(:o)\(\oa^ \(\ti
.if "\\*(LA"JS" .tr #\(\sh\$ \(\Do@ \(\vZ\(\vS\\\\\\\\(-D]\('C^ \(\vC^ \(\vz\(\vs)\(\(-d)\(\vc^ \(\('c \\" z JUS I.B1.002
..
```

Problems we did not pursue

Written English is based on the latin alphabet, it hardly uses accents and other variations of letters. Other languages make use of accents above (è), below (ç) and through (ø) the letters. We do not address the problems of languages with more than one accent per letter and accents connecting letters. In our opinion these problems have to be solved by separate preprocessors. This allows a much more friendly user interface. These preprocessors could also solve the problem of hyphenation and ligatures for these languages.

The troff naming scheme

Troff has three ways of naming characters:

- one-character ASCII names like A for A, B for B and @ for @.
- escaped one-character names prefixed by a \ like \- for current font minus, \e for backslash and \acute{v} for acute accent. There are only a few of these.
- two-character names prefixed by the indicator \(` like \(`sc for §, \(`*g for γ and \(`14 for ¼.

The sets of one-character and escaped one-character names are fixed. Only the set of two-character names can be extended.

Choosing new names

While choosing names for new characters we were very much aware of the fact that the restriction of two characters per name defies all attempts to choose a consistent and logical naming scheme. Still we used a few principles in choosing the new names. Whenever these principles conflicted we refrained from long discussions but placed more value on a quick decision.

Our principles:

- may not conflict with original troff manual¹
- try to avoid the national characters: # \$ @ [\] ^ ' { } ~
- use names associated with the graphical description of the symbol. Thus \oA for Å instead of \AA.
- Whenever a character is a combination of an accent and a letter use as name a character representing the accent followed by the letter. For example: \`e for é. This is the way these combinations were made on old-fashioned typewriters: first hit the dead key with the accent then the key with the letter. The symbols we used for the accents (and the like) are:

ASCII	'	,	:	,	-	o	^	.	"	u	v	-	/
Accent	'	,	"	,	-	o	^	.	"	u	v	-	/
Name	aa	ga	ad	ac	a-	ao	a^	a.	a"	ab	ah	a-	/

- fractions are done in the natural way: \(`nm for $\frac{n}{m}$.
- Currency signs have two letter names. The first letter is Capital, the second small.
- The names for accents start with an a, for example \(`ad for ".

The weird characters one has

Suppose you have a flat, which is unlikely on any other device, but still want to have your input sent to your friend/editor which can run it on his troff. The first thing to do is not using the character as it is implemented on your troff (f.i. \(`ft), but define a string at the beginning of the text and use that. The next thing to do is to give a description how it should look like. So if we stick with our flat, you end up at the start of your input file with:

```
.\" We use a "flat" (\(`ft) a lot instead of a backslash, because it
.\" stands out nicely and, since a backslash can be interpreted in
.\" so many ways, I want to make clear that you see an escape when I
.\" talk about it....
.\" So I define the string ft
.ds ft \(`ft
.\" A "flat" is what in music stands for lowering the current
.\" tone. If that not is clear, consider a | (pipe character) with
.\" a small circle attached to it on the left bottom side.
```

```
.\\" A define in the style of
.\\"     .ds \o'|\o'
.\\" will do if you don't have anything more than a lineprinter around.
.\\" If you want to be really fancy, you might want to try
.\\" something like:
.\\" .nr x \w'o'/2u
.\\" .nr y .2m
.\\" .ds ft \v'-\nyu'\z|\h'\nxu'\v'\nyu'\s'-9'\o\s'0'
.\\" Fancy ain't it?
```

So the rest of your article will look (at the input side) as:

An escape (denoted by *(ft) in troff will introduce a two character name XX by *(ft (so *(ft(XX) and a two character named QQ string interpolation will be triggered by *(ft*((*(ft*(QQ) .

The future

New versions of *troff* include more forms of naming characters. Both \C'arbitrary long name' and \N'absolute Number of character in current font' are used. The first allows character names of arbitrary length. This does not solve the problems discussed in this article. Worse, it will even be harder to choose names upon which a sizeable number of people will agree.

Bibliography

1. J. F. Ossanna, "NROFF/TROFF User's Manual," Comp. Sci. Tech. Rep. 54, Bell Laboratories, Murray Hill, NJ (October 1976).
2. B. W. Kernighan, "A typesetter-independent TROFF," Comp. Sci. Tech. Rep. 97, Bell Laboratories, Murray Hill, NJ (March 1982).
3. Mark Kahrs and Lee Moore, "Adventures with Typesetter-Independent TROFF," TR 159, University of Rochester, NY (June, 1985).

Our character set

We divided the characters in four categories. Each character is mentioned only once. Whenever we doubted we tried to place a character in the category we thought was most suitable. Lastly we included the characters from Ossanna's troff document for reference.

Symbols from ISO 646 standards

Char	Name	Char	Name
#	\sh sharp	¥	\Ye Yen
¤	\Cs Currency sign	\$	\Do Dollar
£	\Po English pound	@	\at at sign
[\IB left square bracket	\	\rs backslash
]	\rB right square bracket	^	\ha hat or accent circumflex
{	\IC left curly bracket]	\rC right curly bracket
	\ba bar (possibly broken)	-	\ti tilde
~	\a" accent double quote	Æ	\AE AE
Ø	\O O slash	Å	\oA A circle
æ	\ae ae	ø	\o o slash
å	\oa a circle	Ä	\:A A diaeresis
Ö	\:O O diaeresis	Ü	\:U U diaeresis
é	\'E e acute accent	ä	\:a a diaeresis
ö	\:o o diaeresis	ü	\:u u diaeresis
ß	\ss German ringel S	à	\`a a grave accent
ç	\,c c cedilla	è	\`e e grave accent
ù	\'u u grave accent	ò	\`o o grave accent
í	\'i i grave accent	í	\! reverse !
Ñ	\~N N tilde	í	\? reverse ?
ñ	\~n n tilde	ã	\~a a tilde
â	\^a a hat	ç	\,C C cedilla
ê	\^e e hat	î	\^i i hat
ð	\~o o tilde	û	\^u u hat
É	\'E E acute accent	À	\'A A acute accent
á	\'a a acute accent	í	\'i i acute accent
c	\'c c acute accent	C	\'C C acute accent
Ó	\~O O tilde	Ã	\~A A tilde
ð	\-d d bar	Ð	\-D Capital Icelandic Eth (D)/D bar

Symbols from ISO 8859 standards

Char	Name	Char	Name
ð	\Sd small Icelandic eth (d)	í	\!/ Polish I
Ł	\L Polish L	þ	\Tp Small Icelandic Thorn
Þ	\TP Capital Icelandic Thorn	ı	\bb broken bar
¹	\S1 superscript 1	²	\S2 superscript 2
³	\S3 superscript 3	ë	\:e e diaeresis
È	\'E E grave accent	ó	\'o o acute accent
ô	\^o o hat	ú	\'u u acute accent
²	\Of feminin ordinator indicator	º	\(Om masculin ordinator indicator
«	\Fo double french open quote	»	\Fc double french close quote
,	\ac accent cedilla	..	\ad accent diaeresis
¶	\ps english paragraph sign	í	\:i i diaeresis
½	\12 one half	¼	\14 one quart
¾	\34 three quart	·	\md centered dot
¬	\no not		

Typographical characters

Char	Name	
'	\ab	accent breve
'	\ah	accent haček
~	\a~	accent tilde
-	\a-	accent macron
i	\.i	dotless i
<	\fo	french open quote
IJ	\IJ	ligature IJ
™	\tm	Trade Mark
œ	\oe	French oe
✓	\YOK	check mark

Char	Name	
º	\ao	accent corona
.	\a.	accent dot
^	\a^	accent circumflex
.	\ho	hook
J	\j	dotless j
>	\fc	french close quote
ij	\ij	ligature ij
'	\oq	open quote
Œ	\OE	French OE
%o	\%o	per mille

Mathematical characters

Char	Name	
ñ	\-h	h bar
⊗	\c*	circle times
^	\OR	logical or
∀	\fa	for all
∴	\3d	therefore
∉	\nm	not a member
⊄	\cn	does not contain
ℵ	\Ah	Aleph
∓	\+-	minus plus

Char	Name	Meaning
\oplus	$\setminus c+$	circle plus
\approxeq	$\setminus =\sim$	approximately equal
\vee	$\setminus \text{AN}$	logical and
\exists	$\setminus \text{te}$	there exists
\perp	$\setminus \text{pp}$	perpendicular to
\supset	$\setminus \text{cn}$	contains
\approx	$\setminus \sim$	approximately
$\not\equiv$	$\setminus \text{ne}$	not equivalent

Symbols from the Troff manual by Ossanna

Char	Name	3/4 Em dash
—	\em	
•	\bu	bullet
_	\ru	rule
½	\12	1/2
fi	\fi	fi
ffi	\Fi	ffi
ffl	\Fl	ffl
†	\dg	dagger
¢	\ct	cent sign
©	\co	copyright
–	\mi	math minus
*	**	math star
'	\aa	acute accent
—	\ul	underline
α	*a	alpha
γ	*g	gamma
ε	*e	epsilon
η	*y	eta
ι	*i	iota
λ	*l	lambda
ν	*n	nu

Char	Name	
-	\hy	hyphen
□	\sq	square
¼	\f14	1/4
¾	\f34	3/4
fl	\fl	fl
ff	\ff	ff
°	\de	degree
'	\fm	foot mark
®	\rg	registered
+	\pl	math plus
=	\eq	math equals
§	\sc	section
`	\ga	grave accent
/	\sl	slash (matching backslash)
β	*b	beta
δ	*d	delta
ζ	*z	zeta
θ	*h	theta
κ	*k	kappa
μ	*m	mu
ξ	*c	xi

Char	Name	Char	Name		
o	*o	omnicron	π	*p	pi
ρ	*r	rho	σ	*s	sigma
ς	\ts	terminal sigma	τ	*t	tau
υ	*u	upsilon	ϕ	*f	phi
χ	*x	chi	ψ	*q	psi
ω	*w	omega	Α	*A	Alpha
Β	*B	Beta	Γ	*G	Gamma
Δ	*D	Delta	Ε	*E	Epsilon
Ζ	*Z	Zeta	Η	*Y	Eta
Θ	*H	Theta	Ι	*I	Iota
Κ	*K	Kappa	Λ	*L	Lambda
Μ	*M	Mu	Ν	*N	Nu
Ξ	*C	Xi	Ο	*O	Omnicron
Π	*P	Pi	Ρ	*R	Rho
Σ	*S	Sigma	Τ	*T	Tau
Υ	*U	Upsilon	Φ	*F	Phi
Χ	*X	Chi	Ψ	*Q	Psi
Ω	*W	Omega	√	\sr	square root
—	\rn	root en extender	≥	\>=	>=
≤	\<=	<=	=	\==	identical equal
≈	\~=	approx =	~	\ap	approximates
≠	\!=	not equal	→	\->	right arrow
←	\<-	left arrow	↑	\ua	up arrow
↓	\da	down arrow	×	\mu	multiply
÷	\di	divide	±	\+-	plus-minus
∪	\cu	cup (union)	∩	\ca	cap (intersection)
⊂	\sb	subset of	⊃	\sp	superset of
⊆	\ib	improper subset	⊅	\ip	improper superset
∞	\if	infinity	∂	\pd	partial derivative
∇	\gr	gradient	⊓	\no	not
∫	\is	integral sign	∞	\pt	proportional to
∅	\es	empty set	∈	\mo	member of
	\br	box vertical rule	‡	\dd	double dagger
↷	\rh	right hand	↶	\lh	left hand
④	\bs	Bell System logo		\or	or
○	\ci	circle	{	\lt	left top of big curly bracket
└	\lb	left bottom	}	\rt	right top
┘	\rb	right bottom	{	\lk	left center
┐	\rk	right center	—	\bv	bold vertical
└	\lf	left floor	—	\rf	right floor
┘	\lc	left ceiling	—	\rc	right ceiling

Mk'ing Hardware: A Tutorial

*Tom Killian
tom@research.att.com*

*AT&T Bell Laboratories
Murray Hill, New Jersey 07974 U.S.A*



Tom Killian started life as a high-energy physicist. He worked at CERN, in Geneva, and at Brookhaven National Laboratory on Long Island. In both places he was unable to persuade his colleagues to stop writing in FORTRAN. He joined the Computing Science Research Center at Bell Labs in 1982, where his research interests include operating systems, real-time hardware and software, laser-printer hardware and software, computer music, and espresso machine repair.

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

Introduction

We begin with an overview of files and dataflow in the UCDS system.

The logical part of a design usually consists of circuit diagrams (.j files) and PAL equations (.e files). The circuit contains *chips*, each identified by *name* (which is arbitrary) and *type* (which is generic, e.g., 74LS74). Chips have *pins*, each identified by a *pin name*; pin names are local to a chip, and are a property of its type. Pins are connected by *nets*, which have unique *net names* (assigned by the drawing package if omitted by the user). It is an error for a pin to be connected to more than one net. The semantics of a circuit diagram (its .w file) are derived from the .j file by listing, for each chip, its name, type, and net-pin connections.

The physical design consists of tables matching pin names with numbers for each chip type, and chip types with packages (.pins files), geometric descriptions of chip packages (.pkg files), a geometric description of the wire-wrap board (.brd file), and a list of positions of the chips to be placed on the board (.pos file). Most of the information in .pins and .pkg files comes from standard libraries, but the user must generally supply some of it, usually for I/O connectors. In the case of a PAL, pin information and package type are extracted from its .e file into a .p file.

The wire-wrap 'compilation' process proceeds as follows. First, a .w file is generated from each .j file, and a .p file is made from each .e file; next the .w files are read in conjunction with the .p files and any pin libraries to make a .pins file; now the .w and .pins files are processed to make the .net file. This step also performs macro expansions, but that need not concern us; the salient property of the .net file is that all nets are expressed in terms of pin numbers. Now the .net file is read in conjunction with package libraries to make a .pkg file, and the .net, .pkg, and .brd files are used to generate position

information (.pos file). Finally, the .net, .pos, .pkg, and .brd files are used to make the point-to-point wrap list (.wr file), and the process is complete.

PAL configuration is a three-step process. First the logic equations (in the .e file) are transformed into a representation of functions as sums of minterms (.min file). Then the .min file is transformed into a fuse list appropriate to a particular device. Finally, the fuse list is downloaded to a PAL programmer that burns the fuses. The last step requires knowledge of the PAL manufacturer as well as its type, since different manufacturers often use different programming algorithms.

The Simplest Case

Suppose we have a design specified by design.j, control.e, and design.brd, with libraries /usr/ucds/lib/lib.pins, and /usr/ucds/lib/lib.pkg. Then our mk file might appear as follows:

```

design.w:    design.j
            jraw -w design.j >design.w

control.p:   control.e
            lde -W control.e >control.p

design.pins: design.w control.p /usr/ucds/lib/lib.pins
            mkpins design.w control.p /usr/ucds/lib/lib.pins >design.pins

design.net:  design.w design.pins
            cdmglob -f -v design.w design.pins >design.net

design.pkg:  design.net /usr/ucds/lib/lib.pkg
            mkpkg design.net /usr/ucds/lib/lib.pkg >design.pkg

design.pos:  design.net design.pkg design.brd
            place design.net design.pkg design.brd >design.pos

design.wr:   design.net design.pkg design.brd design.pos
            wrap -c -v design.net design.pkg design.brd design.pos >design.wr

```

Transitive closure guarantees the correct dependency semantics. The file can be made more generic by using meta-rules where possible, and parameterizing elsewhere. The recipes can be made less prolix by use of \$prereq and \$target:

```

NAME=design
PALPINS=control.p

HWLlib=/usr/ucds/lib
LIBPINS=$HWLlib/lib.pins
LIBPKG=$HWLlib/lib.pkg

wr:N:      $NAME.wr

%.w:      %.j
        jraw -w $stem.j >$target

%.p:      %.e
        lde -W $stem.e >$target

%.pins:   %.w $PALPINS $LIBPINS
        mkpins $prereq >$target

%.net:    %.w %.pins
        cdmglob -f -v $prereq >$target

%.pkg:    %.net $LIBPKG
        mkpkg $prereq >$target

```

```

%.pos: %.net %.pkg %.brd
    place $prereq >$target

%.wr: %.net %.pkg %.brd %.pos
    wrap -c -v $prereq >$target

```

With a little reorganisation, we can support several pieces of hardware that have this structure, each in a separate directory such as hw/design. In this directory we put a file called config containing the first two lines from the mk file. The remainder of the mk file goes into a public place, say, /usr/ucds/lib/mkfile. Finally, we put this one-liner into our search PATH, say, /usr/ucds/bin/Mk:

```
exec mk -f config -f /usr/ucds/lib/mkfile "$@"
```

Now the command 'Mk wr', executed from hw/design, will have the effect of building design.wr.

Realistic Complications

The first problem with this scheme is that most designs have several .j and .e files, so these must be specified in config in addition to the NAME:

```

NAME=design
JFILES=cpu.j io.j
EFILES=control.e decode.e

```

Now we can use backquote substitution to get the lists of .w and .p files that we need. At the beginning of our mk file, we insert:

```

ENVIRON="resuffix() {
    echo \$1 | sed 's/\.[^ \t]*\.'\$2/g'
}"
WFILES='resuffix "$JFILES" w'
PALPINS='resuffix "$EFILES" p'

```

Note the use of \\$ to prevent mk from attempting parameter substitution when ENVIRON is assigned. The rules using %.w as a prerequisite must be modified to use \$WFILES:

```

$NAME.pins: $WFILES $PALPINS $LIBPINS
    mklpins $prereq >$target

$NAME.net: $WFILES $NAME.pins
    cdmglob -f -v $prereq >$target

```

Another problem is with the dependency of the .pos file. The mk file says that positions must be recomputed whenever the .net file changes, but only rarely will this be the case. Since positions are most likely determined through a complex interactive process, we don't want their calculation triggered gratuitously. So, we replace the rule for %.pos with this one:

```

%.place: %.net %.pkg %.brd
    place $prereq >%pos

```

This allows the user to work on the positioning at will, since %.place will never be 'up-to-date'. Conversely, since %.place does not appear in the prerequisites for any rule, mk will never force the user to do placement unnecessarily.

Updating the Design

The most serious problem to be addressed is that of reworking the design. Invariably changes must be made after the board has been built, and of course we don't want to rewire from scratch. The UDCS program rework bears the brunt of this. Given old and new .wr files (representing the present and desired states of the board), it produces three files as output: a list of old wires to unwrap (UN.wr), a list

of new wires to wrap (RE.wr), and a file representing the new board state (NEW.wr). (Rework may alter wrap levels in order to minimise the sizes of UN.wr and RE.wr, so NEW.wr may not be identical to the new input file.)

It is vital that `mk` not overwrite the old .wr file until we are sure that we no longer need it, and it would be nice if it invoked `rework` when appropriate. So we change the recipe for %.wr as follows:

```
%.wr: %.brd %.pkg %.net %.pos
    wrap -c -v $prereq >N$target
    if [ -f $target ]; then
        rework $target N$target && mv NEW.wr N$target
    else
        mv N$target $target
    fi
```

Now we make, e.g., Ndesign.wr. If design.wr already exists, we run `rework` and install its output as the new wrap file (since we assume one design per directory, we don't bother renaming UN.wr and RE.wr); otherwise we simply rename the output of `wrap`. It is important that the old wrap file is never touched or renamed, so we can change our minds and 'Mk wr' as often as we like before physically changing the board.

When we actually do change the board, the presence of multiple wrap files becomes a liability. So we add a `mk` rule to invoke when we have finished rewrapping:

```
%.commit:
    mv N$stem.wr $stem.wr && rm UN.wr RE.wr
```

Some Extra Touches

We would like to be able simply to type 'Mk place', 'Mk wr', etc., instead of having always to supply the stem. Instead of adding a rule for each such abbreviation, we can use a single regular-expression meta-rule that uses back-referencing to handle them all:

```
^ (place|wr|commit) \$:RN: $NAME.\`1
```

We generally need design-specific .pins and .pkg libraries, so we make the amendments:

```
LIBPINS=$HHLIB/lib.pins io.pins my.pins
LIBPKG=$HHLIB/lib.pkg my.pkg
```

In order to make these files optional, we add the rule:

```
my.pkg my.pins io.pins:
    touch $target
```

Thus empty files will be made if required, but since no dependency is specified existing files will not be touched.

Finally, we add a rule for running the wiring machine:

```
%.wrap:
    if [ -f N$stem.wr ]; then
        wrap -s -r$ORIENT UN.wr
        wrap -s -r$ORIENT RE.wr
    else
        wrap -s -r$ORIENT $stem.wr
    fi
```

This rule chooses the appropriate .wr file or files. ORIENT is a new parameter to be set in the config file to specify the orientation of the board in the machine, and is the principle reason for bothering with the rule.

PAL Programming

PAL programming in UCDS suffers from some interesting quirks. Lde, the program that generates minterms, quite properly doesn't care about the properties of real devices; such knowledge is left to pal, which generates the fuse list. Thus the .min file must be fed to pal to know whether or not the equations are going to fit, but the type of PAL that we are using (e.g., 16L8) is not part of the .min file! It happens that the PAL type is easily extracted from the .p file, so given a program (paltype) to do this, we can generate and test our .min files like this:

```
% .min: %.e %.p
    lde -a $stem.e | quine | cover | hazard >$stem.min.ng
    pal -d 'paltype $stem.p' $stem.min.ng >/dev/null
    mv $stem.min.ng $target
```

First we make a .ng (no good) file. If an output has too many terms, pal will exit with non-zero status, the recipe will be aborted, and the .ng file may be studied to figure out a way around the problem. Otherwise the .ng file becomes the target. (It is not useful to retain the output of pal, since it costs little to compute and is incomprehensible to humans.)

When we actually burn a PAL, we have to specify the manufacturer. It makes little sense to save this in a file, since we don't know whose devices the stockroom will have on any given day. So we make the manufacturer name part of the target string in a regular-expression meta-rule:

```
'^([^.]+)\.drom\.(^.+)$':R: \\\1.min \\1.p
man=$stem2
type='paltype $stem1.p'
pal -d $type $stem1.min | tee to | drom -w -t $man$type
drom -t $man$type >from
cmp to from; rm to from
```

Now if we type, say, Mk control.drom.TI (and control.e calls for a 16L8), the first command executed will be:

```
pal -d 16L8 control.min | tee to | drom -w -t TI16L8
```

This writes the PAL (drom -w), and saves a copy of the fuse list. The remainder of the recipe reads back the PAL and checks that it was programmed correctly.

Conclusions

We have used this single mkfile with multiple config files to manage about twenty different pieces of hardware. It has proved especially convenient on small designs. Also appreciated is the ability to add targets (e.g., check to perform diagnostics, draw to make plots) to the central mkfile, and have them globally available.

References

A. G. Fraser, **UNIX Time-Sharing System: Circuit Design Aids**, *Bell System Technical Journal* 57 (6), pp. 2233-2249 (1978).

A. G. Hume, **Mk: A Successor to Make**, *AT&T-BL Computing Science Technical Report* 141, October, 1987.

Mk is available from the UNIX System Toolchest.

A European Representation for ISO C

Keld Simonsen

University of Copenhagen, Denmark

Bjarne Stroustrup

AT&T Bell Laboratories, USA.

Keld is a well known ice cream eating chairman of the Danish group who is active in the Danish group, the network and in the area of international standards.

Bjame has gained international recognition with his creation of the C++ language. Dr Stroustrup received a MSc in computer science from Cambridge University, England. His research interests include distributed systems, operating systems, simulation, and programming.

Not being native English speakers both of the authors are ideally suited to be heading a campaign to solve the 'European C problem'.

The proposed ANSI C standard uses the American national character set, ASCII, for its representation, but as other national character sets use some characters differently, the ANSI C standard proposal includes a specification for an alternate representation, the *trigraph* representation. This paper proposes an extension to the trigraph proposal based on keywords and two-letter combinations of special characters, that is much easier to read and write. We hope this proposal will stimulate a discussion about this and related problems leading to a widely accepted solution.

The problems

Writing programs in C can be quite difficult where English is not the native language. The problem is that C uses symbols from the American national character set as operators and punctuation characters. This implies that you cannot use all letters from a national alphabet in identifiers and that programs appear strangely transformed on your screen or printer because some C operators are printed as the corresponding letters.

In many countries in Europe, South America and Africa the national character set standard is a version of the ISO 646-1983 international standard for 7-bit character sets. ASCII (ANSI X3.4-1968) is the national version of ISO 646 in the USA. It is the impression of the authors that ASCII and the national standard competes bravely from country to country for being the national industry standard. Also, on IBM mainframes there are

national EBCDIC implementations very close in character repertoire to the formal national standards. In some areas, such as Scandinavia, the national character sets are predominant both in the 7-bit and the EBCDIC world. In other countries, such as the Netherlands, you mostly see ASCII. There are some new international standards based on 8-bit characters (the ISO 8859/1/2 Latin character sets and the ISO 6937/2 Videotex standard) which includes both ASCII and the various national characters, but it will take years, if not decades, for these new official standards to become industry standards and for 7-bit equipment to completely lose its importance.

It is quite desirable (and only polite) to allow programmers to write (and read) programs using their national 7-bit character set representation. The recent proposed ANSI C standard allows a representation of C programs in a national character set that does not have all of the

characters used to represent a C program in ASCII. This is achieved by defining alternative representations, trigraphs, for the “offending” C operators and punctuation characters. However, there is still no provision for identifiers with national letters and the trigraph representation is not very suitable for human eyes.

ASCII and Trigraph Representations

The ISO 646-1983 standard leaves 12 positions undefined and up to national standards organisations to decide. For those undefined positions used in C for national ISO chars, the proposed ANSI C standard specifies a trigraph representation, a three-character replacement beginning with two question marks (??). Three characters ('\$', '@' and '\'') are not used in the proposed ANSI C (but they are used in some dialects of C) so we will also refrain from using them.

The 12 ‘problem’ characters in ASCII and their equivalent ANSI C trigraph representation are:

ASCII	#	\$	@	[\
Trigraph	??=			??(??/
ASCII]	~	`	{	
Trigraph	??)	??'	??<	??!	??>

ISO646 reserves the characters # and \$ so that they cannot be used for letters of a national alphabet. This ensures that they are not a problem in the context of C. The other 10 characters are “available for national or application oriented use.”

Let us see how these things looks on a common example. First the ordinary C version in plain ASCII:

```
main(argc, argv)
int argc;
char* argv[];
{
    if (argc<1 ||
        *argv[1]=='\0')
        return;
    printf("Hi,%s\n", argv[1]);
}
```

And this is how it looks in the Danish ISO 646 character set:

```
main(argc, argv)
int argc;
char* argv[ÆÅ];
```

```
{
    if (argc<1 ØØ
        *argv[Æ1]=='Ø0')
        return;
    printf("Hi,%sØn", argv[Æ1]);
}
```

Experience shows that at least some people can learn to read and write this. In our opinion this is not a skill anyone should be encouraged to enquire. The begin-end brackets and the or operator look ugly and the array subscripts simply drown. It looks just as weird in, say, the Finnish, French, German, or Spanish ISO 646. This is the reason for the proposed ANSI C trigraph solution. How does our example look using trigraphs?

```
main(argc, argv)
int argc;
char* argv??<??>;
{
    if (argc<1 ??!??!
        *argv??<1??>=='??/0')
        return;
    printf("Hi,%s??/n", argv??<1??>);
}
```

Where are the array subscripts? Which operator is used in the conditional statement? We believe that even though the problem of defining an ISO 646 representation is solved by introducing trigraphs, the original problem of being able to write C programs without losing one’s native language has not been attacked at all. The resulting “C programs” are unreadable and unwritable.

We see no alternative to using trigraphs for representing {, }, \, etc., in strings and character constants. For example:

```
switch (tok) {
    case '{': ...
    case '}': ...
    case '\\': ...
}
```

becomes

```
switch (tok) {
    case '??<': ...
    case '??>': ...
    case '??/??: ...
```

Trigraphs are not pretty, but with the notable exception of backslash they will not be common in this context.

Our Proposal

To solve the original problem one could provide a combination of new keywords and reasonably nice-looking one- or two-letter symbols.

New keywords:

or		
cor		(conditional or)
and	&	
cand	&&	(conditional and)
xor	^	
compl	~	(complement)

Introducing new keywords is always a way to break existing programs, but the keywords could be conditionally in effect for new programs. The new keywords requires a few more keystrokes than the ASCII characters, but not many, and some would consider them to improve readability of C! The keywords and and cand were added for consistency—it seemed silly to have an or but no and. The & and && operators are still valid. The new keywords (except cor and cand) can be combined with = to make assignment operators.

Note that use of trigraphs will not by itself avoid the problem of keywords. Macros will be used to make trigraphs more palatable. For example:

```
#define begin ??<
#define end   ??>
#define or    ??!
#define cor   ??!??!
...
```

Naturally, groups of programmers will agree on standard versions of such macros and naturally not everyone will agree on the same versions. In a few years we will therefore be faced with the problem of having several incompatible sets of "de facto keywords" and conventions. The alternative is to pick a minimal set now. This is what this proposal does.

New symbols:

(:	{
:)	}
!([
)]
??/	\

The use of ??/ as the escape character is probably the least elegant of these alternative representations, but we see no acceptable

alternative given that trigraphs are necessary in some contexts anyway. One might argue that an escape need not to be too pretty anyway.

We decided to make the compound statement brackets digraphs, finding 'begin' and 'end' too long to write and too likely to be found "not in the spirit of C" by large numbers of programmers. The digraphs (/ /) might have been preferable to (: and :) but current usage precludes this. For example

```
/*
 * argument type commented out
 * waiting for ANSI C compiler:
 */
```

```
int printf/* const char*, ... */;
```

Using (: will cause a minor problem for C++ parsers because C++ has the prefix scope resolution operator ::. For example:

```
if (::open("myfile.c", 0)==0) ...
```

This problem can be solved either by a bit of lookahead in a C++ lexical analyser or by requiring C++ programmers to use a space before the :: operator in this relatively rare case.

The grammar of C precludes using (and) for subscripting (as is done for many other languages). Using keywords to represent [] would be unacceptably awkward. Using digraphs would be little better but we could not find an acceptable pair. For example (/ ... /) suffers from the problem mentioned above, (! ... !) collides with the common usage 'if (!p) ...' and (* ... *) collides with the common usage 'if (*p) ...'. Parentheses are typically unnecessary for subscripting and ! should be considered an infix subscript operator (as in BCPL). The binding strength of the binary ! operator (subscripting) should be just above the unary operators and it should be left associative. For example, a!b.c!2*d means ((a! (b . c)) !2) *d.

Our proposal would allow the program look like this:

```
main(argc, argv)
int argc;
char* argv();
```

```
(:
```

```
if (argc<1 cor
```

```
*argv! (1)=='??/0')
```

```
return;
```

```
printf("Hi,%s??/n",argv!(1));
:)
```

Using ! as an infix operator, this can be further cleaned up:

```
main(argc, argv)
int argc;
char* argv!;
(:_
if (argc<1 cor
    *argv!1=='??/0')
    return;
printf("Hi,%s??/n", argv!1);
:)
```

One would still need parentheses for more complicated subscripts; for example, v!(i+j). The symmetrical nature of subscripting in C will actually be more obvious. An apparent ambiguity would exist between ! used for subscripts and ! used as the negation operator. Consider:

```
int v[!4];
```

This would have to be represented as

```
int v(!(!4));
```

since

```
int v!!4;
```

would mean

```
int v[] [4];
```

Programs written this way still have a distinct C flavour. Locally, they can even be more terse than C and they are always shorter than programs written using the trigraph notation.

Practical Details

As observed, the trigraph proposal solves only a small part of the problem they were introduced to compensate for, whereas this proposal is an almost complete solution.

It is debatable whether it should be legal to mix the constructs proposed here, such as (:, or, and compl, with traditional C constructs, such as {, |, and ~.

Mechanical translation from the standard American/English notation for C to this proposed "European" C notation is trivial. The potential ambiguity of translating [] into ! is handled by using parentheses systematically; [expr] becomes !(expr). The reverse translation requires understanding of operator precedence to

handle subscripting. Furthermore, if national characters are allowed in identifiers such characters must be expanded into sequences of English characters. This is no major problem.

Conclusions

We have found a *usable* solution to the international representation problem of C for most languages using a latin alphabet. The solution is quite easy to implement in a C compiler. We tried it.

The (:, :, and ! constructs do not affect existing programs, and the new keywords are chosen so that it is likely that only a minimal number of existing programs will be affected. Using them only in programs identified as non-US ISO C should remove completely the possibility of hurting existing programs.

If adopted this proposal will open the way for allowing national characters to be used in identifiers.

This proposal is submitted for consideration to the ANSI C committee and the ISO committees dealing with C. It is also considered for adoption into C++.

Acknowledgements

Brian Kernighan and David Prosser found errors in previous versions of this proposal and suggested improvements to the presentation.

References

K&R: C

ANSI C draft

Richards: BCPL

BS: C++

Received: from roof.gardenshed.jocks-cabin by patio.macmiggins.stornoway via shoutnet; 21 Jan 89 20:50 SST
 Received: from patio.macmiggins.stornoway by bonfire.oban.beaconnet via infernonet; 25 Feb 89 15:31 SST
 Received: from bonfire.oban.beaconnet by cs.nott.ac.uk via cannet; 30 Mar 89 21:43 BST
 To: jpo@cs.nott.ac.uk
 From: jock@gardenshed.jocks-cabin
 Date: Fri, 21 Jan 89 20:50 SST
 Subject: Alternative protocols

Alternatives to OSI

Jock C St. Martin

*University of the Outer Hebrides
 Scotland*

Following recent discussions concerning the relative merits of OSI and ARPA protocols, I decided to throw my hat into the ring. Furthermore, I believe that the ARPA protocols are not the only contenders with OSI, and that a number of even more 'mature' mechanisms exist. I present seven possibilities for consideration.

1. Bean tins and bits of string

The use of bean tins and taut pieces of string has long been recognised as an effective means of communication. In fact, excavations from Anglo-Saxon dwellings in Nottingham show their use (albeit with imported coconuts as opposed to bean tins) in early everyday office situations.

Bean tins and string have several advantages over OSI:

- a. They are fast, light weight and portable.
- b. They don't require the purchase of expensive computers.
- c. Complex error correction (based on the "NO - I said ..." principal)
- d. Uses off the supermarket shelf technology.
- e. They were not invented by the ISO.

They also exhibit a very few trifling limitations:

- a. Poor support for 'packet' switching (however, tin switching may be supported).
- b. Users often cut themselves on the tins.

- c. Star network topologies become more complex.
- d. They don't scale very well.

2. Shouting from the roof tops

Shouting from the rooftops can be an effective method of optimised local area communication. It is based on the well understood CMSA/CD technology but with the notion of priority. Users can insert high priority traffic with the "If I might get a word in edgeways" packet. It is already in widespread use—e.g., the House of Commons, political canvassing and Speakers Corner. Naturally, a roof top is only necessary for high bandwidth traffic. The PTT's would probably assume this role. The average user would be content to shout in the street.

Shouting has many advantages over OSI:

- a. It is not as 'complex and obscure'.
- b. Most people understand shouting.
- c. Broadcasts are easy.
- d. It's fun.
- e. It wasn't invented by the ISO

OSI has hardly any advantages over shouting:

3. Burning beacons on hill-tops:

Burning beacons on hill-tops have long been used to warn of advancing Armadas and their like. However, the author believes that beacons may

have wider applications than just these.

In particular, they have the following advantages over OSI:

- a. No 'dangerous checkpointing'.
- b. They keep you warm.
- c. Not overly complex and obscure.
- d. A secondary use for the disposal of those nasty ISO people.
- e. Not cluttered with unnecessary functionality.
- f. Not invented by the ISO.

Disadvantages to OSI:

- a. Not suitable for the office environment (this may really be an advantage in some circumstances).
- b. Low bandwidth (may also be an advantage—see 7)
- c. Error rates can be high. Arsonists, pyromaniacs and 'Satanic Verses' burners can generate spoof packets.

4. Semaphore

Semaphore has been in use for many years. So why did ISO not consider this for international internetworking? This is difficult to determine, but is probably due to political motivations rather than any deficiencies in the protocols. Naturally there are a few rough edges to be addressed.

Advantages over OSI

- a. Broadcasts are easily accommodated.
- b. Widely supported off-the-shelf infrastructure (boy scouts).
- c. Not invented by ISO

Disadvantages over OSI

- a. Not so useful at night (but a working party on luminous flags is in progress).
- b. Bandwidth is rather low—but automation should help.

5. Messages in bottles

This is a low cost solution to networking. Bottles are easy to obtain and with a little development, this neglected backwater of communications technology could be a real alternative.

Advantages over OSI

- a. High bandwidth data channels already in existence (e.g. the gulf stream, rivers and sewers.)
- b. Large amounts of data can be placed in the appropriate sized bottles.
- c. Not invented by ISO.

Disadvantages to OSI

- a. Transit time is unpredictable (but then IP, for instance, does not guarantee any bounded delivery time)

6. The Telephone

This might be seen as an enhancement of method 2. However, there is a lot to be gained from this approach. The name lookup problem is already solved as are routing issues. Lets face it, communications protocols are ultimately used for communicating between people. So why not just standardise the telephone. Add on services such as broadcast agents (commonly called gossips/operators) are easy to achieve.

Advantages over OSI

- a. Its a mature existing technology.
- b. Directory services issues, routing and charging are already established.
- c. It's now available in portable form.
- d. Not invented by ISO

Disadvantages to OSI

- a. Because it's a mature technology, there aren't so many interesting research areas.
- b. As a result of 2. there are few exotic conference openings. It costs money.

7. Not communicating at all

One question I asked myself was "why communicate at all?" On consideration it was realised that not communicating has the following advantages over OSI.

- a. Low consumption of bandwidth.
- b. Cheap and easy to manage.
- c. No one disagrees with you.
- d. Without the time wasted on communication, other business proceeds much quicker.

- e. Not invented by the ISO

No known disadvantages to OSI.

The ARPA protocols.

The ARPA protocols deserve consideration along with many of the above mentioned methods of communication. In particular, they have one major advantage over OSI.

- a. Not invented by the ISO

However, despite this overwhelming advantage of the Internet protocol suite, the ISO proponents simply will not give in. In this section I therefore give a few other reasons for the superiority of the Internet suite—as if 1. was not enough.

Scalability. The Internet protocols are obviously scalable as has been proved time and time again. All that is required is for the PTT's to take the sensible step of providing a network infrastructure and the rest can be solved. Charging is easily accommodated—the PTT's pick up the bills.

Network interface. Many people have commented on how convenient it is to have a network address which fits into a common word size. This is such a advantage that the limitations are really insignificant. If the address space ever gets used up there is an obvious extension mechanism—the waiting list.

Session layer. The Internet suite sensibly disregarded session services as superfluous. As has been observed, checkpointing is inherently dangerous as it can lead to loss of network usage and revenue. OSI has been influenced by the Internet community here, and has provided a session service complex enough that most implementations try and ignore it.

Presentation layer. Again the Internet triumphs. It is quite clear that for the most part applications only need to exchange data consisting of bytes of 8, 16 and 32 bit quantities. These simple structures can be used as building blocks to construct almost any structure required. If this is not sufficient, there is a simple escape mechanism provided, known in the jargon as a "string encoding". It is quite clear that ASN.1 is just over the top—CHOICE's and OPTIONAL's are for quiche-eating indecisive applications.

Application layer. Well the Internet has got this one too. Honestly, it's quite obvious that each

application should do its own thing. That's what they're there for. If an application needs remote procedure call interface, or security, or name lookup, then it can do it itself rather than forcing it to use some more general service like ROS or directory services.

Summary

In summary, I feel that all of the above methods are orders of magnitude better than OSI (which incidentally, and by coincidence, wasn't invented here). In particular, I feel that method 7 offers the greatest potential and, with this in mind, WE DO NOT WELCOME ANY FURTHER COMMENTS YOU MIGHT HAVE!

Editor's note

This article is in no way connected with either Julian Onions or Steve Benford of the University of Nottingham beyond their rôle as postal agents for the author.

Brussels conference report



Marc Gumbold
marc@pbinfo.uucp

*Universität-GH Paderborn
West Germany*

Marc Gumbold is a graduate CS student at the University of Paderborn, West Germany.

He finds it rather hard to survive without money and is therefore spending some of his time as *postmaster@pbinfo.uucp*.

Although he has not yet been to Italy he is really fond of Cappuccino and Espresso, but doesn't disdain other kinds of coffee either.

Once upon a time there was a German student who decided to go to an EUUG conference for the very first time in his life. He sent email off to *euug@inset.uucp*, which he found to be the uucp address of the EUUG Secretary, in order to get more information about the Spring '89 Conference. Some 10 or 12 days later he eventually received a nice booklet containing the Conference Programme and—shock!—the costs. As he would like to attend the *UUCP & Sendmail Configuration* tutorial very much, he would have to pay £475 altogether. Impossible! Rather annoyed he read the chapter about 'Student Grants' and wondered whether a student grant application would be accepted. As there were only a few days remaining before the booking deadline he eventually decided to make a phone call to the EUUG Secretary in England. A kind gentleman (who you might guess is Bill Barrett) on the other end of the line told him that they will pay the fee for the Conference and one tutorial for him, if he just agrees to write something for the EUUG Newsletter. Of course he accepted.

Well, this tale is actually the prehistory of (a) my participation in the EUUG Spring '89 Conference in Brussels and (b) this report. You already guessed *that?* Ok. Never mind. Let's start the action.

Considering my financial situation I tried to book a bed in a youth hostel. I got a bed for all but the first night, but wasn't disturbed too much by this as I thought there would be no problem in finding some place to sleep the first night, once I was in Brussels. As Brussels is really not too far away

from Paderborn (perhaps some 500 kilometers) I decided to go there in my own 12-year-old veteran car.

Monday

For various reasons I couldn't start on Sunday so I left Paderborn *very* early on Monday morning heading for Brussels. Not being too much troubled by the snow on my way I approached Brussels at about half past seven in the morning. I got stuck in the morning rush hour. It was eight o'clock when I found myself in the very center of Brussels. Imagine the situation as follows: Holding the city map in the one hand, turning the steering wheel with the other, and changing gears with the third I was looking with one eye on the city map, with the other on the street to watch traffic, and with the third I was looking out for the Palais de Congrès where the Conference was to be held. (You will easily notice a slight mistake I made; there are a limited number of resources like hands and eyes of course. However I would not like to explain the scheduling algorithm to you right now...) Perhaps this would have ended up in chaos if not my car started making some strange noises. So I had to stop. A quick investigation revealed the cooling water had boiled, so I had enough time to ask an elderly woman who just came along, where the Palais de Congrès could be found. She spoke French :-(, found out I was German :-), and continued to speak French :-(. Well, nevertheless somehow I understood her. Actually the Palais was only one or two hundred meters away. I managed to get my car in an underground car park near the Palais and went off.

The Palais de Congrès is situated on the *Mont des Arts* or *Kunstberg*. Both means *Mountain of Art*. This is a really funny thing in Brussels: all the streets and places are labeled in French and Dutch. You'll perhaps know that people in northern Belgium speak Dutch (well, really Flemish, but don't be fussy about that, it's roughly the same), and those in southern Belgium speak French. But although Brussels is located north of the language border, most of the inhabitants speak French.

Well, let's return to the Palais de Congrès. Entering an entrance labelled 'EUUG' I went straight to the registration desk. I was welcomed by Bill Barrett. He immediately remembered my name and handed over the Conference Proceedings and my name badge to me.

There was still some time remaining until the beginning of the Conference. I had a look around. The conference room itself was rather big and equipped with nice and comfortable seats in red and orange tones. After a 'phone call home I decided to sit down in the foyer and wait for things to happen. More and more people dropped in, got their proceedings and their badge and did the same I was doing. I met a couple of people from the German EUnet backbone in Dortmund who I had seen before. Sigh! At least someone I knew a little.

Well, eventually we sat down in those cosy seats, waiting for the Conference Chairman, Prof. Marc Nyssen from the Free University Brussels, to open the Conference. He found some nice words to welcome us and presented a keynote speaker, Dennis Tsichritzis. Dennis spoke about the need for object oriented program development. I really got the feeling that what he said was rather interesting but his speed and non-standard English made it rather hard for me to follow.

With some other 400 people I rushed out to get coffee. There was a slight problem with my stomach as it thought it was lunch time already (remember I got up at 2.30 in the morning). Walking around I got the impression that everybody else seemed to know everybody except me. Hmm. But I felt somehow attracted by the multi-lingual noise level surrounding me, having a glimpse at all the badges showing where people came from. Denmark, France, UK, Germany, Netherlands, even people from the States.

The next session was dealt with distributed operating systems. It began to get really boring. I leaned back and wondered if it had been a really good decision to go to this Conference. But the last talk rather changed my mind. Sape J. Mullender reported about their Amoeba system development in a very impressive manner. I even forgot to think about my rumbling stomach.

Finally, lunch. There was a restaurant inside the congress center where lunch was served. During the meal I took a quick course in Belgian history, given to me by a Michel Bardiaux from Brussels, who was sitting next to me.

The next three talks were entitled 'Networks'. First, John S. Quarterman, American network guru, reported about 'Recent changes in North American Computer Networks'. I was really confused by the amazing number of networks they have over there. If you nevertheless would like to learn more about this: John will publish a book about these topics in this year. After a contribution from Austria about an educational network, Daniel Karrenberg gave an overview about EUnet, the European UNIX Network. He also presented the teams managing the national EUnet backbone computers. A very nice idea, every team showed some slides about what their site looked like.

Coffee again (it was really no problem for coffeeholics like me to survive this conference...). Chatting with the Dortmund backbone gurus revealed that there was one empty bed in one of their hotel rooms. So at least I knew where to spend the night.

The next and last chapter for this day was 'Graphics', including two very interesting items: First the development of a medical image processing system on a SUN workstation, and second a five minute computer animation which was computed in a network of SUNs using only the machines' idle times. This one was done at the University of Karlsruhe. Really impressive!

After having some free drinks (just orange juice for me, as I had to get my car out of the car park), me and Jan-Hinrich Fessel from Dortmund (the guy who agreed to share his room with me for one night—thanks again) went to the 'Hotel Madeleine'.

After relaxing for a short moment we decided to find some place to eat something. As there were six of us, everybody having different criteria of

choosing the right pub, it took us a rather long time and some kilometers of running around, until we all were hungry enough to agree to enter one pub near the *Bourse* (or *Beurs* which all means *stock exchange*). We had some beer and something to eat, which was really not as expensive as you could think, considering the nice interior decoration which probably had not changed much since the early 20ies. I'm afraid I must disappoint you in not being able to give you a report on the quality of Belgian beer (I think you should ask *postmaster@unido.uucp* for that...). I'm in no way a beer professional, I only drink it in case of extreme thirst. If you would like to chat about coffee, you're welcome.

Now I became really tired, after all I have been on my legs for more than 20 hours. After having a look at one or two more pubs I decided to return to the hotel and have some sleep(). But the hotel was in the very center of the old town, including masses of cars driving around all the night over streets with cobbled pavement. However I managed to sleep a few hours.

Tuesday

After a lousy breakfast in the hotel I transferred my car and my luggage to the youth hostel. I found a car park *free of charge* near the youth hostel. Now that was something! I had paid 320 Belgian Francs for car parks since my arrival on Monday morning. (Look it up in your newspaper if you would like to know how much that is...)

From the youth hostel I reached the Palais de Congrès with a five minute walk. It was an ugly cold morning and I was really sleepy. Nevertheless I rather enjoyed the first talks about 'Network Software'.

Coffee break. Outside it was snowing now. Brrrr...

Luckily there were no windows in the conference room itself, so you could at least imagine that the sun was shining outside. Well, inside we were listening to the very interesting talk of David Rosenthal from Sun Microsystems. He spoke about the relationship between response times and memory, but it was much more interesting than it probably sounds. I really encourage you to read his paper 'More Haste, Less Speed' in the proceedings, especially if you're a workstation user.

Lunch. While I find it a bit difficult to get in contact with others during the coffee breaks, it's

much easier during lunch. This time I had a chat with Claude Anzala, from the French EUnet backbone, about some mail problems. It was especially nice to speak to people you only (so far) knew by name; Teus Hagen for example.

The whole Tuesday afternoon was about 'Standards'. I was especially looking forward to a talk named 'UNIX Standardisation: an Overview', but was a bit disappointed that this talk was rather a description of the POSIX concept than an overview over existing standards. Next Robert L. Duncanson from AT&T Europe reported about System V Release 4, followed by Kevin R. Fall from Berkeley speaking about new features of BSD UNIX 4.4(?)

Coffee (sigh!) was followed by John Totman's talk about X/Open. A change in the programme and the next talk was held by Morris Schwartz from 'UNIX International'. Phew! I've heard about X/Open, OSF, POSIX, and so on, and I must say I was already confused by all these standardisation efforts. But now still another group, which was presented to be the 'System V Product Planning Authority' and which will support X/Open. I really began to loose the thread totally. Then Henning Oldenburg from OSF who explained some things about OSF/Motif, yet another window system standard... oh no!

I was tired. And I was really fed up with all this standardisation stuff. There was a BOF (Bird Of a Feather meeting) about standards, but I decided to have a look at my youth hostel to relax a bit. Besides there was the Conference Dinner at eight o'clock in the 'Autoworld Museum'. Laying on my bed I tried to work out how to get there. For several reasons I didn't want to go there by my own car, a taxi was considered to be too expensive, it was too far to walk, so I decided to go by bus. Line 20 would be ok, there was a bus stop very near the youth hostel, so I would have no problems.

A little after half past seven I boarded a bus. It was a wonderful tour; the bus was a real veteran, the streets were bumpy, so I soon felt every bone inside me. Having the city map on my knees I tried not to loose orientation in order to get out at the right bus stop. I noticed another passenger who was equipped with a city map. He got out at the same bus stop and followed me. I was not surprised to find out that he planned to go to the EUUG Dinner. After a few minutes we luckily found the entrance to the Autoworld Museum.

Really amazing, I had never seen such a great number of old cars before. I got a drink, talked to some fellow EUUGians and admired all these lovely old cars. They were even older than mine! I was especially attracted by a bus, who seemed to be in a much better condition than that one that took me there and I really wondered why this one was staying in a museum considering the much worse material on the streets outside.

Finally, the dinner. There was nothing extraordinary about it, except for the fact that, in contrary to the lunches, people had to get up from their seats, and go to one of three (or four, I don't know exactly) tables to get the food onto their plates. Of course this led to race conditions...

Now I really had a problem: The youth hostel closed at midnight, I didn't know how to go back (bus lines seem to be unidirectional in Brussels) so I had to get up earlier than the rest to reach the hostel in time. (I always had the dark feeling there was a hitch to the low price of a youth hostel...) However I managed to get back by metro and reached the youth hostel at five minutes to twelve.

Wednesday

The first talks on Wednesday morning revealed nothing special. No, the talk of one Japanese (!) should really be mentioned. He spoke about the development of a high speed filesystem which becomes necessary if huge masses of data, for instance video data, must be accessed.

Tom Killian from the AT&T Bell Labs presented an improved 'make', called 'mk'. In addition to the well-known 'make' it understands meta-rules defined by regular expressions and transitive closure of dependencies.

So far I have forgotten to mention that there was a competition. The task was to produce net addresses that were likely to me more descriptive than those normally used. After lunch on Wednesday the results were proclaimed:

The winner was Dominic Dunlop with *worm@broadcast* for which he was rewarded with a UNIX licence plate.

The second prize, a T Shirt was given to Paul Bussé for:

1. Tom's paper can be found elsewhere in this newsletter.

five%work@myhost.

Some other results listed for your enjoyment:
Honourable Mentions:

i-am@cinema.darling, Eva Kühn.
*\$+\\<@\$-1\\>\$**, Philippe Dax.
*hacker@**, Peter Gilis.

Nepotism section:

tilbrook@another.new.job
sunil!@home
keld@nearest.icecream.stand
red-shoe@michel.right-foot

Novel Transport layer:

?#^.?@toilet.paper,
/* the address was actually written on
toilet tissue ... */

In the afternoon we learned something about languages. Two object-oriented talks, and one about distributed logic programming. It seems to be quite fashionable nowadays to distribute all sorts of thing that used to be sequential... The last group of talks on Wednesday afternoon was about 'Tools', but I skipped most of them because I had a look at the exhibition in the basement... Oh yes, the exhibition. I almost forgot it. You could find many really great names there, but I got the impression that there was rather little interest in the exhibition at all. At least until they transferred the coffee break facilities into the basement room where the exhibition took place. Then suddenly the room was overcrowded. Coffee speaks louder than sales talks...

One Belgian firm however managed to present Steve Jobs' NeXT computer. There was always a reasonable crowd around the black cube. I would really love to have such a box, especially for writing this report: that thing's got a built-in dictionary and thesaurus... however when testing it I could find no facility for providing you automatically with ideas as to what to write about... Alas, no machine is perfect.

The meaning of the first letter of 'EUUG' was impressively demonstrated by the fact that one talk was given by Agnes Hemadi, coming from the Hungarian Academy of Sciences in Budapest. She told us about the development of a new, unusual database interface. The very last talk was given by Lee McLoughlin. Lee spoke about his experiences with porting X Windows to different machines, giving hints for all the poor chaps who have to do the same.

Well, Wednesday evening was the official ending of the Conference. Teus Hagen and Marc Nyssen found some nice words—all of which I have forgotten. In the evening there was a EUnet BOF, where several problems concerning the EUnet situation were discussed.

Although the weather was terrible (what luck! I found an umbrella in my car) I had a walk through the old center of Brussels in the evening. I walked over the Grand Place with the famous town hall, but because of rain and coldness nothing could impress me. So after a pizza and an espresso I ended up in the youth hostel, talking the rest of the evening with *normal* people, i.e. those that were in no way related to UNIX. After 3 days UNIX this can really mean some relief.

Thursday

As I planned to go to the *UUCP & Sendmail Configuration* tutorial on Friday I had to stay in Brussels. I could really have gone home now, with all these new impressions I had. Besides, the weather... you already know about. But as I was looking forward for the tutorial I decided to make Thursday my sightseeing day. So I spent three quarters of the day running around in the city. I was really impressed by the amazing number of restaurants and pubs in the old city. There are lanes with, let's say ten or more pizzerias; if you're running out of one you'll find yourself in the kitchen of the next one. Well, I had a look at the 'Manneken Pis', of course. I heard before that it sometimes wears clothes, but was nevertheless impressed finding it dressed with an MP uniform. Although this was surely a respectful appearance I couldn't hide a smile... By the way, do you know there is a feminine counterpart to 'Manneken Pis'? It's called 'Janneke Pis' and was made a few years ago. Just drop an email if you would like to know where to find it. Running around I really got the impression that Brussels was rather a French town, not as clean and tidy as Dutch or Flemish towns are. But I'm not sure if I would have got another impression if there would have been a little bit sunshine. The weather was a real misery. After some shopping I returned to the youth hostel in late afternoon, wet to the skin. I warmed up myself. Because of the fact that there was a small party in the youth hostel, the following night consisted only of some 3 hours of sleep.

Friday

Sleeeeeeeeepy. How could I manage the tutorial? Well, somehow I got to the Palais de Congrès and found out that the tutorial started at 9.30. Hmm, one should really read the timetable. What a pity! I could have had another half an hour inside my sleeping bag.

I checked in for the tutorial and received the tutorial papers. They looked rather promising. The tutorial dealt with the configuration of *sendmail*, a sophisticated mail transport agent, and was given by Yves Devillers, mail guru at the French EUnet backbone. You probably either know by experience or have heard from others that *sendmail.cf*, the *sendmail* configuration file, causes nightmares for most postmasters. Well, it surely didn't do so to Yves, who managed to tame the beast. The mystery of his success was *sendmail*'s debug option, which he tried to explain to us. The pace was all right, although there's much more to say about *sendmail* than could be done in one day. But the provided papers were really excellent, so I think it's no problem to make more use of this tutorial working on the papers at home.

At five o'clock in the afternoon it was all over. I got out of the Palais and found the sun shining for the first time of my stay in Brussels. I just couldn't believe it, after 4 days of rain and snow. I returned to the youth hostel and gathered my luggage. Fortunately I had not to travel alone. I was accompanied by Anke and Bernard from the German EUnet backbone, and Kevin Fall from Berkeley, who decided to go to Germany. I dropped them in Dortmund and arrived at home happily but absolutely tired at exactly midnight.

The conclusion? This was my first EUUG conference but *surely* not the last one. I really enjoyed it. You'll probably not find me in Vienna, because of an exam I've to do in late September. But I'll certainly meet you in Munich next year. I would like to encourage every student who feels somehow related to UNIX, to have a try and go to an EUUG conference. Costs are a problem for students, of course. (Not every chap can pay for the conference by writing a report—who would care to read all these boring essays? :-)).

See you in Munich, Marc.

News from the Netherlands



Frances Brazier
frances@psy.vu.nl

*Department of Cognitive Psychology
Vrije Universiteit
Amsterdam
The Netherlands*

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

Membership

The NLUUG's membership continues to increase in number. In the table below the increase in

academic, industrial and individual membership is shown.

April 88	Nov 1988	April 89	
59	71	80	academic members (@ Dfl 300)
133	157	164	industrial members (@ Dfl 600)
20	31	35	individual members (@ Dfl 100)
212	259	279	TOTAL

Conferences

Our next conference will have been held by the time this newsletter is printed—on May 9th. The theme of this conference is human-machine interaction, a topic we expect to be well received. We have again invited a few 'foreign' speakers to broaden the scope of the programme, one of whom previously spoke on the same topic in Lissabon. The exchange of ideas and of speakers between EUUG conferences and national conferences is one which should be promoted. As has been shown in the past, both parties can profit from each other's efforts. Good speakers aren't

always easy to find, let alone good talks.

Our first experience with the production of proceedings was gained last autumn. Although the proceedings were well appreciated it was decided to leave the decision to produce proceedings up to the programme committee. This has resulted in the decision to defer from the production of proceedings for the May conference. The general aim is, however, that proceedings should be published for at least one of the two conferences held in a year. Speakers

will be informed of the possibility to submit their contributions to this newsletter.

The programme for May 9th includes:

- *Natural Language Communication with Computers: Some compromises and problems.* A Jameson & W Claassen. Nijmegen Institute for Cognitive Research and Information Technology.
- *The Océ 6000 publishing and printing system: the development of a graphical user interface.* René Collard. Océ Nederland BV.
- *Windows.* Shawn Phillips. SUN Microsystems Inc, Mountain View, USA.
- *NeWS and X, Beauty and the Beast?* WT Roberts. Department of Computer Science, Queen Mary College, London.
- *The Generation of Graphical User Interfaces using Attribute Grammars.* S Doaitse Swierstra, HH Vogt, H Bos. State Department of Computer Science, University of Utrecht & HCS Information Technology.
- *An object-oriented approach to UI-design.* Charles van der Mast & JM Versendaal. Department of Computer Science, Technical University of Delft.
- *X-Window and Open Dialogue: a standard UIMS.* Bouwe van der Eems. Apollo Computer BV.

The backbone

A new legal body has been established for the exploitation of the new Dutch backbone *hp4nl*: Stichting NLnet. The Board of this Foundation consists of 3 to 5 members of which the majority must legally be appointed by the NLUUG. Currently the Chairman, Treasurer and Secretary fulfill the same positions for both the NLUUG and the Stichting NLnet. A candidate for system management has been found. If everything goes well he will be stationed at CWI as of May 1st.

As of May 1st our network accounting has also been revised—the principle 'sender pays' has been implemented. We now just hope the algorithm's correct. All mail is classified as being either national, European or non-European, and is accounted as such.

The problems with SURF (EARN) have not yet been solved completely. We do, however, hope to have a contract signed by the beginning of May.



EUUG Executive Report

Helen Gibbons

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 on 1st April 1989 just prior to the Spring Conference in Brussels. This was followed a day later by the Governing Board Meeting at which national group representatives from Denmark, Sweden, Italy, Germany, The Netherlands, Portugal, Brussels, Ireland, Norway, France, Yugoslavia, Austria, Great Britain and Finland were represented. There was also an invited observer from USENIX.

Earlier in the year an application for affiliation from Hungary had been accepted, and at the April meetings official approval was granted also to applications recently received from Portugal and Yugoslavia. This means that there are now 16 National Groups within EUUG with a growing membership totaling around 3000.

The Swiss group has been suspended from membership.

The year 1988-89 saw a big improvement in EUUG services, particularly the EUUG Newsletter which grew to be bigger and better than ever before; other publications were also made available, such as the BSD Manuals and E-mail directory, which was a big success and is selling well. Conferences, tutorials, software distributions, public relations and EUnet are all becoming bigger, busier and more professional. This, of course, means not only that investments and budget risks are higher but the work load has also greatly increased. Because of this, the position of an EUUG Executive Director is being considered. The Executive Committee itself will also be expanded to cope with the work load. It will meet again early in July.

The EUUG has been active in its contact with other bodies and groups. Discussions with NCR and SUN user groups are projected, but have not been very successful so far. An exchange of

articles has been arranged with USENIX and co-operation is also taking place on the subject of standards. Meetings are being arranged with OSF, UNIX International and X/Open.

EUnet is moving steadily towards a stable and very reliable service. Most of the core links use either leased lines or X.25. The financial situation of EUnet is now improving and it is expected that the US sites will soon contribute to the US European data transportation cost. It is hoped that the 'mcvax' machine will shortly be replaced by a SUN 4 server. Thanks are due to the co-operation of the many network managers at all the European backbones.

The future strategy of the EUUG will be discussed in great detail at a Strategy Workshop being held in Dublin on 5-7th May 1989.

Although some discussion was held at the April meetings on reducing the number of major conferences to only one a year, it was in the end agreed that the EUUG would continue with its programme of two conferences per year for the foreseeable future. The Spring Conference in Brussels was a great success with an attendance of approximately 400. Future conferences will be held in:

Vienna, Austria	18-22 September 1989
Munich, W. Germany	23-27 April 1990
Nice, France	22-26 October 1990
Tromso, Norway	20-24 May 1991
Budapest, Hungary	16-20 September 1991

There after conferences are in the early stages of planning for Jersey and the Netherlands in 1992.

UKUUG News

Mick Farmer

mick@cs.bbk.ac.uk



*Department of Computing
Birkbeck College*

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

UNIX Security Workshop

We held a very successful meeting on 16th February at the Institute of Education, University of London, attended by over 150 delegates. Many aspects of UNIX security and administration were discussed, leaving many delegates with the wish for a similar workshop to be held in the near future. The proceedings were videoed and the edited programme is now available, initially, in the UK on a three-hour video cassette (details elsewhere in this issue).

Pete, Farewell

Pete Collinson, the founder of the UKnet network, leaves 'ukc' (The Computing Laboratory, University of Kent) after 15 years on 31st May 1989. He started the network as an informal method of communication between early UNIX aficionados in the UK. This was part of a wider interest shown by a small group of similar-minded UNIX people in Europe, particularly in Holland. In the early years between 1980-1984 the number of communicating sites/machines in the UK varied between 20-50. Mail was paid for by each site and the structure was friendly rather than formal. Originally 'news' came in via a free link to 'ukc', supplied by an ex-student of Pete's. When this charitable route ceased in 1984 it became necessary to find an alternative supplier, which meant that 'ukc' was suddenly involved in IPSS charges for shipping Megabytes of news and mail on slow modems. A charging scheme was started

and then the VAT inspectors became interested.

Fortunately for the UK Pete was not easily defeated. The technical problems in handling a UK domain naming system, that was unrelated to the rest of the world, the complexities of potentially changing nightly any of the world mail routings based on latest mapping information, and the authorisation/stopping and charging of mail and news were mere intellectual exercises for him. He will be greatly missed! The overall result is that, since 1985, UKnet has expanded linearly with no signs of abatement. The latest statistics are as follows:

UKnet Statistics (April 1989)	
Academic Sites	199
NRS-registered Academic Sites	182
Non NRS-registered Academic Sites	17
Commercial Sites	187
NRS-registered Commercial Sites	88
Non NRS-registered Commercial Sites	99
Total Sites	386

The continuing structure at 'ukc' will be:

Ian Dallas, PSS & JANET gateway problems, *ind@ukc.ac.uk*

Martin Guy, FTP problems, *mg@ukc.ac.uk*

Ian Harding, News manager, *ih@ukc.ac.uk*

Peter Houlder, Anything else, *uknet@ukc.ac.uk*

Pete is taking the plunge and becoming a freelance UNIX consultant, he can be contacted as *pc@hillside.uucp* or by phone on +44 227 61824. Pete's leaving has been accompanied by two other members of staff. Sean Leviseur (general mailer/networking knowledge) and Chris Downey (former News Manager). Sean is also doing freelance work and can be contacted as *sjl@westhawk.uucp* or by phone on +44 9282 2574. Chris will be working for Unisoft in London, phone +44 1 315 6600.

Ukc will continue to support the network and we have many interesting ideas for the future, but members are requested to show patience in the early days of transition.

UKUUG Summer Meeting 1989

This meeting takes place in Glasgow, Scotland, on 27-28th June. The local organiser is Jim Reid from Strathclyde University. A synopsis of the papers organised so far is given at the back of this issue. Jim Crammond will also organise a tutorial on the UK-Sendmail package.

UKUUG Winter Meeting 1989

We plan to hold this meeting in Cardiff, Wales, on 12-14th December, 1989. Further details will follow when they are available.

UKUUG Summer Meeting 1990

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

UKUUG Winter Meeting 1990

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

London UNIX User Group (LUUG)

The X-Windows system de-mystified was the title of Ray Anderson's talk at the London UNIX User Group in April. More than 60 people crowded into the lecture room for the latest LUUG Lecture. Ray Anderson (IXI Ltd.) opened the talk by describing the origins of X in MIT's Project Athena; a far-reaching programme intended to help students and tutors communicate through computers. The goals of X were to be device and network independent, to support several concurrent applications on each display, and to be extensible.

In 1984 Robert Scheifler of MIT started with a copy of 'W' from Stanford and by July 1986 the X10 tape was made available to the world. Sun's announcement of their NeWS windows system later that year prompted eleven major companies to join MIT in forming the X Consortium. With a full-time team now at work, X11 was produced in less than a year. X11 is the current version, and has now reached Release 3.

X is based on the 'client-server model': client programs communicate with a server process which handles the screen, keyboard, and mouse. To people used to 'fileservers', 'printservers' and the like, it seems a little odd to have the *server* on the desk while the *clients* are in a back-room somewhere. "No problem" said Ray: "Servers exist to look after scarce resources. By comparison with the number of tasks they handle, users are getting pretty scarce!"

Some important features were described: The X-window system (unlike Presentation Manager) has an *asynchronous* protocol, which makes it usable over high-delay links such as satellite channels. In fact, X could be made to work over *any* reliable order-preserving channel. The X server maintains a hierarchy of windows, and the protocol is designed to make window-creation cheap. As a result, almost every rectangle on the screen is a window in its own right. Even the humble scrollbar is likely to contain four or more windows.

The lecture ended appropriately with a set of slides showing the range of available widget styles and window managers.

Ray Anderson can be contacted at:

IXI Ltd. of Cambridge.

Phone: (+44) 223-462131

IXI distribute the X-windows system, the programming manuals, and also their own X.desktop product.

The next LUUG lecture will be at 19:00 on Thursday 25 May at 18 Bedford Square, London WC1. Jim Crammond of Imperial College will describe the new UK Sendmail Configuration Package.

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

UKUUG UNIX Security Workshop

*Thomas Grossi
grossi@capsogeti.fr*

Cap Sesa Innovation

Introduction

I was able to attend the UKUUG seminar on the Security of UNIX Systems. There was a diversity of interesting presentations, whereas the audience was almost all British, and I think that I was the only delegate from France.

The Hackman project

*Russell Brand, Lawrence Livermore National Labs
speaking for Peter Shipley*

The Hackman project goal was originally to give users a safe system. Originally they planned to catalogue all security bugs, but that turned out to be too vast a domain, so they restricted it to UNIX bugs, and thereafter to BSD UNIX. The end result of the project will be a book listing known security bugs, describing them, giving a history and a program illustrating the problem, and showing how to fix them. Another goal is to come up with a page or two of programming hints to help ensure that you write a secure program (like "never use gets; use fgets instead"—at the heart of the recent Internet worm).

Brand has an interesting definition of a security bug: something that lets someone do something you wouldn't want him to do if you knew he could. All bugs, says he, are based on erroneous beliefs at some level or another – the programmer assumed the system would work a certain way, but it doesn't. Some bugs from version 4.1a still exist in SunOS 4.0.2. Sun's Yellow Pages have turned out to be a veritable mine of bugs: for the past few weeks, they've been coming up with a new bug each day.

One of the questions from the audience concerned System V. Brand replied that the group had picked BSD because it was what they were all most familiar with and what they used the most; if there is a great deal of overlap between the two, it might turn out that for a small amount of additional work they could cover SysV as well, but they don't know.

Another question concerned the possibility of giving the catalogue to vendors in exchange for a binding agreement in which the vendors agreed to fix all the bugs within a certain period. Brand said that they had not thought of that yet, but seemed pleased by the idea.

Brand himself was a major source of questions for the rest of the day.

*666 etc.,
Lindsay Marshall
University of Newcastle-upon-Tyne*

The title of this talk was a play on the default privileges of files on UNIX systems and on caballistic symbolism (remember Proctor & Gamble?). You should come away from this talk with the feeling that protection=666 is something to be wary of!

An awful lot of security holes exist for the simple reason that many system files have too-permissive permissions. A major problem is that vendors dump the machines with absolutely no explanations on users who just don't know what they're doing. For example, the architecture department at Newcastle recently brought a workstation onto the network that had no root password and read/write permission on everything!

Marshall suggests that a multiplicity of distinct uids and gids for different utilities (uucp, news, sendmail, yp, etc.) will tighten up considerably a system's security. The SAs should su to each specific utility-user rather than working as root. This has two other significant advantages: 1) the amount of damage you can do accidentally as one of these users is thus greatly diminished; 2) it allows you to farm out duties to several people. There is somewhat of a problem in figuring out how to can carve the utilities into separate uids, but it's less difficult than one would expect.

Other random comments: There's no reason whatsoever for allowing a binary file to be

readable—to run a program it suffices to be executable. One should keep an eye on setuid programs—they're inherently dangerous. ‘ps’ and ‘who’ could be considered security problems—Joe User shouldn't necessarily be able to tell what everyone else is doing on his machine. When using chmod, type chmod a+rwx rather than the format chmod 666—it's less error-prone (since it's more mnemonic and won't remove execute permission by accident). Watch out for gnu software—a lot of it has a umask of 000 (and therefore leaves files with a protection of 777). This is understandable if you take into account Robert Morris' character.

*Secure RPC
Andy Rutter
The Instruction Set*

W.r.t security, network applications are hard. Security is often ignored or added as an afterthought, and credentials can be easy to fake. As a result, it's usually left up to the applications to take care of security issues. Of course, the same comments often apply here as well, with the additional problem that the solutions are usually ad hoc.

In SunOS 4.0, it was decided to offer authenticated platform services, to assure at least a minimal level of security. The goal was to make the network as secure as a unix time-sharing environment (adequate for general use) while adding a minimum of overhead. It was assumed that it is not possible to replace packets in transit through the network and that eavesdropping is not a problem; therefore the data are not encrypted. On the other hand, impersonation ('masquerade') is taken to be a problem, so servers and clients both want authentication of their dialogue partners.

How is this authentication done then? The client selects a conversation key (CK) at random. (This ensures that no one can break the encryption schemes by monitoring conversations and picking out the keys used.) This key is encrypted using a public key (PK; some well-known constant raised to a secret-key power). The client then sends a triplet to the server: (identifier, PK-encrypted CK, CK-encrypted time window). The server acknowledges by sending back the time value + 1, encrypted with the CK (again, 1 is added so that the value sent back is different from the value received—thus preventing eavesdroppers from figuring out the keys). If the validation sent back

falls within the allowed time window, the client knows that it is talking to the authentic server.

How does this affect users of the Sun network? NFSSRC 4.0 offers secure NFS and YP. For an RPC user, the changes are completely transparent. Performance degradation is about 2 per cent. For the installer, there are some minor differences. Public keys are kept in /etc/publickey. In /etc/exports one must add:

/home-secure, access=net-group
and one must also add secure to the options in /etc/fstab.

Of course, it is possible to break secure RPC. One method is to record transactions that occur on the network and replay them when the server crashes. This can be avoided by making the time window less than reboot time. (By default the window is a half-hour or an hour.) If there is a gateway, the network automatically becomes insecure, because the gateway has the possibility of substituting data packets. Rlogins can be a Trojan horse. Also, if someone is dedicated enough, it is not difficult to jam the network and substitute data packets.

Other problems arise as well. When a server crashes, the cached keys disappear and conversations cannot be resumed—they must be restarted. If a server is diskless, its disk server can be faked. Setuid programs do not work unless the user is logged in, as the conversation key is generated at the beginning of the session. And you must be able to trust your public key servers.

*YP
Willian Roberts, Queen Mary College*

The Yellow Pages was originally intended to be a distributed look-up service, not a distributed data base. This is why it fails so badly when it comes to modifying the data it stores.

How does YP work? The system has been set up so that it's completely transparent to application programs. Whenever an application needs some information that is not known locally, the routines in libc know to ask the ypbnd.

First, a bit of background information. On each machine a program called ‘portmap’ keeps track of rpc services that have been registered, and tells which ports are used by which servers. In general the ports are given out arbitrarily, but for YP servers, the server is given port 111. When ypbnd needs yellow page information, then, it broadcasts a pre-request to port 111 on all

Simple attacks:

- the frame buffer is usually writable
- window manager logs are usually readable (e.g., Sun's commandtool leaves mouse-selected data in /tmp)

Boot attacks:

- by default, anyone can boot his workstation in single user mode
- you can get a root login by interrupting the execution of /etc/rc
- if you have a local streamer, you can load mini-unix from the installation tapes

Monitor attacks:

- patch kernel data – proc table or user area entries
- patch kernel text – patch the code with adb solutions:
 - disable monitor mode
 - get a secure PROM monitor

YP attacks:

- corrupt the data bases

NFS:

- the protocol is open and public
- it's extremely vulnerable, due to its weak authentication
- even SunOS's 'secure' NFS can be broken the attacks:
- save up NFS actions, since they contain file handles, and replay them
- one can forge file handles, NFS credentials, mount request and even bogus clients
- protection:
 - use nobody's uid (-2) for root requests across the network
 - setuid, sgid mount options
 - secure UDP port numbers (in NFS = 2049) – any user can bind to it if it's not running
 - mound demon checks: set ipaddr_check to non-0

Advantages of SunOS 4:

- controlled root access
- 'secure' option similar to secure RPC (above) with stronger authentication and public-key encryption. However, the keys are distributed via YP!!!

The Internet Worm

Jim Reid

I won't go into the details, as they are available in the various reports that are circulating.

In his comments, he stressed how vulnerable we would be to a virus that would be posted to comp.sources.unix, for example. In a scenario of what might happen, he depicted a virus for libc that would be included in some utility and which would lie in wait until the program ran as root. When the unwitting root came along, the utility would then install its virus in the system's libc, for example. Consider the difficulty of finding such a virus! Or perhaps it could install a faulty login. But there you might find it out by looking at the code, so better yet, install a faulty C compiler that checks to see if it's compiling the program 'login'. Then when one looked at the code, it would be ok, but the virus would still be there. But one might be able to find its source by looking at cc with a debugger. So you could modify the debugger, so that it knows what not to display if it gets called on cc ...

*Open Session,
at the day's end.*

The question was once again raised, How many system's actually pay attention to frequent password changes and to the security of those passwords? Another interesting point made was that dump tapes should be secure, else anyone can procure a copy of sensitive files.

Most of the talk, however, centered around students in a computing environment. It appears that most security problems come from students who find that break-ins are a clever thing to do. The most effective approach to dealing with this situation seems to emphasise the ethics involved, promising appropriate punishment but not making a big deal out of the problem.

This is an article that was originally printed in the AFUU Newsletter Tribunix, reprinted by kind permission.

machines. This message says, "say 'yes' if you can answer this, else say nothing." (In this manner ypbnd can act on the first answer it gets back.) Ypserv replies giving the port number on the local machine of a data base server corresponding to the request. Ypbnd then sends the real request to the address indicated.

There are several problems with this system. First off, ypbnd doesn't keep track of the number of attempts it makes to find a DB server. It basically keeps on trying forever, and if it can't find one, your process hangs. Likewise, libc only checks to see whether rpc has failed or whether it has replied "yes"; the "no" case has not been considered. The use of portmap is a major security hole: you can ask to unregister the official ypserv and put your own (bogus) server in its place.

*The Cambridge Experience,
Pete Brooks.*

This talk was not terribly enlightening. The blurb in the day's program sums it up quite well:

"Cambridge University's Computer Laboratory has a number of vendors' systems (DEC, SUN, HP, Xerox, Acorn & HLH) mostly running various flavours of UNIX. The central power is a 'processor bank' of Micro-Vax-IIIs which live 'in the cellar'. They are used as CPU servers from ethernet terminals. The main (disk) servers do not have user accounts (only admin) and are trusted by all clients. Access to and from WAN to LAN is transparent. We also have IP over X.25 links."

"When some anomalies were noticed, our academic laissez-faire approach was laid aside and we made some use of our security gurus to tighten things up. Passwd was changed to force eight mixed chars, no words or keyboard strings, etc. Login was tightened up to check the calling address and behave accordingly. The importance of reliable logging was once more emphasised to us."

*Some Myths and Facts about UNIX Security,
Mario Wolczko,
Department of Computer Science,
University of Manchester.*

A breach of security is possible when 1) a file has inappropriate owner or group permissions, or 2) a process attains an illegal user id. The former is usually due to improper user education; the usual method in the latter case is via setuid.

The kernel itself does not really care about users and passwords; for it, users and groups are only numbers. The problem lies in the applications built upon the kernel. For example, demons that trust everything they get, users who exchange passwords.

All processes have a real uid and an effective uid. The real uid is the uid of the login user. The effective uid corresponds to the permissions allowed to the process and is governed by the setuid bit. At any time, however, a process can set the effective uid to the real uid OR VICE VERSA.

The danger of setuid scripts: given a script 'chty' that is run setuid, give four ways of grabbing root privileges. Its contents are:

```
#! /bin/sh
/etc/chown `whoami` 'tty'
```

Solutions:

1. cp trojan whoami
PATH=.:\${PATH} chty
trojan gets executed instead
of system whoami
2. cp trojan etc
IFS=/ PATH=.:\${PATH} chty
IFS is the field separator.
in this case, 'etc'
becomes an executable pg
3. ln /usr/bin/chty -i
-i # -i option gets passed to sh,
makes it interactive
4. there's a small time window between the instant at which the kernel has decided to give the process root permission and the execution of the file, during which it is possible to substitute another file for chty and have it executed instead.

Another weak spot lies in the fact that file descriptors are inherited across execs, and can even be passed down sockets. (In SunOS 4, 'send' only works with the 'unix' call.)

*Workstation Security,
Jim Reid, University of Strathclyde*

A list of ways security can be broken on workstations.

Networking Column

*Peter Houlder
uknet@ukc.ac.uk*



*Computing Laboratory
University of Kent at Canterbury*

Peter Houlder has been in the Computing Laboratory at the University of Kent for the last 4 years and looked after day to day UKnet admin work in the last 3 years.

He graduated in Geography from Kings College, London in 1970 and then spent 9 years in business—dropping out in 1979. He then spent a year touring North, Central, South, and 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.

This month we have two reports. The first from the Italian and the second from the Norwegian Network. Spain have promised an article next time, which should be interesting as they have

gone from 0 to 31 sites in about a year. Denmark have pointed out that they have submitted lots of details in the past. My target list is therefore reduced to Greece, Ireland, Iceland, Luxembourg, Portugal and Yugoslavia.

Status Report from the Italian Backbone

*Joy Marino
Alessandro Berni*

*Dipartimento di Informatica
Sistemistica e Telematica
Universita' di Genova
iunet@dist.unige.it*

The IUnet

The network continues to grow: our sites count is currently 46 and we plan to increase by another 5-6 in the near future.

10 sites are receiving the news: this causes us problems with the 2 dial-up lines (1200-2400 bps), which are always busy, and with disk space (only 170 Megabytes). Another 2 telephone lines are going to be ordered to support the Dataconsyst T2000 modems (that is Telebit Trailblazers in a different box :-)) that we have just purchased. To conserve disk space we run expire each night and move the old news on a Sun 4 server as a

temporary solution.

Our ‘news’ plan is the following: as soon as it is possible move the WORM disk system from the small machine based on a National processor to the faster Sun 4 (a.k.a. ugdist) in order to ‘see’ it using NFS. Then, install the UNO server (USENET News on Optical disk) developed here at DIST and distribute the remote query program to our nodes.

This will prevent us from having to make periodic backups on tape (since all the news of interest will be maintained online on the 2 Gigabytes optical disk) and will enable our users to ‘order the back issues’ making use of the keyword search service

implemented by UNO.

We plan also to have an on-line archive of PD software, based on the same WORM disk with NFS.

What we really hope anyway is to get a faster machine with SysV.3, 200-300 Megabytes of disk space and NFS support. It seems that UNISYS will be ready to give us a machine matching our needs before the *i2u* Convention (taking place in early June).

A Cisco gateway (MGS with a 2200 packets per second board) has been purchased by the DIST to support a *private* IP link to the CNUCE Institute of the National Research Council, which is connected to the USA using the old SATNET. It can however support a second synchronous link, so it has been decided that it will serve also the incoming EUnet IP link. This link to Pisa will be used to reach the other italian networks that is to exchange .IT mail.

What we would like to be made clear is that even if we were to get the link to the Internet via the CNUCE today, we would have still the same determination to pursue the establishment of an European IP network and are ready to give full support to this initiative.

We have contacted the INFNET (Italian HEPnet) people about sharing a small part of their Bologna-Geneva high-speed link but got no answer until now (maybe they are too scared :-)). We are however determined to go on with this matter and in case the 'INFN Connection' does not come to a solution we will be forced to hook up to some European backbone. That would be a pity, since we 'backboners from the suburbs of the network' would be forced to pay a higher price for the same service (our distance from 'central' Europe raises the line cost, which is already high since our PTT has extremely expensive fares).

An 'IP Network(s) Interconnection Demo' is planned for the *i2u* Convention.

IUnet and EUnet

We are currently maintaining links with the following backbones: mcvax, inria, unido, enea, dkuug. The problems with our X.25 software have been finally solved, so we would like to establish links with the other backbones (possibly in both directions, that is we would like to *call* and to *be called*).

site	Mail statistics within IUnet (Feb 88-Jan 89) (in Kbytes)			
	(ita)	(bit)	(eur)	(usa)
automa	3.96	0.00	0.00	0.00
ccauimi	1993.41	531.91	877.29	725.25
cip	17.56	37.51	123.31	48.21
cise	242.08	10.64	1645.47	352.39
cniac	48.95	6.85	5.73	21.36
crai	33.12	18.45	619.59	309.72
crcc	31.70	0.00	7.85	7.34
cseit	5240.00	8.57	2035.00	1472.19
delphi	775.18	43.05	2927.00	14813.54
depolimi	0.00	0.00	0.88	4.17
dipisa	305.73	375.38	2295.00	2173.06
disrm	138.87	25.02	1472.31	1482.18
enidbo	174.93	197.04	1159.64	569.10
i2unix	3881.00	950.30	4213.00	4085.76
ic8u74	15.95	7.00	109.94	319.98
iconet	6345.13	53.79	5938.00	594.78
ieeupdb	7.26	0.00	11.97	29.11
imacnr	0.71	0.51	5.76	240.47
intecs	0.00	0.00	1219.00	0.00
irst	1095.00	1150.00	3756.03	2239.41
leonardo	708.49	0.77	622.00	186.21
lipr	5.54	0.00	0.00	0.00
littix	4.42	0.00	0.81	0.29
olhqm01	2.96	1.26	47.42	6.39
olita2	0.00	1.26	0.00	6.39
sadix	6.72	3.44	3.73	26.07
sarinit	106.75	0.41	1323.67	8759.53
semto	6611.49	35.59	946.67	7969.23
siemil	35.33	0.00	0.00	0.00
sissa	0.00	9.68	0.00	6.38
sogesta	54.00	0.00	0.00	0.00
synarea	3.07	0.00	1.39	1.93
ubdm	752.68	294.15	587.17	810.07
udsab	338.62	1964.93	818.00	2711.86
ugdibe	0.88	0.00	88.50	0.00
ugdist	1371.15	220.78	2507.00	2663.34
unirc1	1.46	0.00	0.00	0.00
unisysit	0.00	0.00	0.00	6.39
zel	86.36	0.00	30.08	23.31
Total	30440.46	5948.30	35399.19	52665.38

IUnet and the Other Networks

Talks have been going on within the GARR (Group for the harmonisation of research networks): the immediate result is that the IUnet subdomains have been registered in the primary server for .IT (making use of MX records). We now pass on our UUCP link to the CNUCE a rapidly growing volume of traffic. When we decide that it's the right moment (that is, when they will depend on our mail service more than they do now) we'll inform them that we cannot afford any longer to pay *with our money* for *their* mail. Our hope is to find a satisfactory agreement on this *financial* topic, since we are sure that they can afford it :-). The problem with mail directed to .IT that is being bounced back to the sender is

due mostly to problems with the routing tables of the other Italian networks: cooperation is going on to ensure that this problem finds a rapid solution.

IUnet and the Hackers

Most of you know of the campaign set up by a group of wily hackers (I mean *cyberpunk hackers*) who claimed they could offer the equivalent of our mail and news service for free.

Since they refused to subscribe the EUnet services they had been *blackholed*, that is their mail was being rejected and returned to the sender. Unfortunately this was not well understood on the other side of the Atlantic: site pyramid (willing to support an initiative that seemed to be compliant with the 'Usenet spirit') offered them a news and mail feed for free. The danger of having an anti-EUnet growing day-by-day was avoided when

pyramid disabled their UUCP logins: an investigation with Italcable had proved that those hackers were using stolen PADs and X.28 passwords (that's why they offered their service for free!) a real hacker's solution!

After that they have tried (are trying) to re-establish friendly relations with us: we of course feel the need of granting to *amateurs* a *limited* access to the network, but we don't think that those hackers of Sublink have the necessary accomplishments of capability and honesty to run the *amateur* part on our behalf.

A *popular computing* magazine, which already runs the biggest Bulletin Board System in Italy (based on a UNIX system b.t.w) has shown a definite interest for *retailing* EUnet mail and news to their end-users.

EUnet in Norway

Steinar Overbeck Cook
steinar@fdmetd.uucp

Fellesdata A/S,
Nedre Skøyenv 26,
Postboks 248,
N-0212 OSLO 2,
Norway

Here is some information on how things are going in Norway.

Norway has officially been part of EUnet for approximately 1 year now. We have had various problems with our backbone, a Norsk Data machine running some sort of UNIX under SINTRAN. This did not work very well so Norsk Data has 'given' us a new backbone, a Wyse 386 PC, which is running UNIX from Interactive. We are still having some troubles though, caused by modems not responding, telephone switchs which does not switch and a lousy serial port controller with brain-damaged driver software. These

problems only apply to the members dialing into the backbone, all communications with the mcvax and so on, seems to work very well. The NUUG's network group are working hard to solve these problems. They are now going to try another controller to see if that helps. The modems on the backbone has also been changed.

When our problems with connecting to the backbone are solved, we will start testing Trailblazer modems. It is very expensive to use dial-up lines in Norway so we hope that faster modems will attract more members to use the services of EUnet.

Puzzle Corner

Mick Farmer

mick@cs.bbk.ac.uk

Hi peeps,

A number of you have pointed out that Puzzle Number 1 (and other puzzles of that ilk) can be solved using the programming language Prolog. Here is one version:

```
/* Dedications - EUUG Puzzle Number 1 */

select(H, [H|T], T).
select(X, [H|T1], [H|T2]) :- 
    select(X, T1, T2).
authors([book(kernighan, steve, C1),
        book(lesk, brian, steve),
        book(ritchie, ken, mike),
        book(bourne, dennis, C2),
        book(thompson, D1, C3)])
    ] ) :- 
    select(D1, [mike], _),
    select(C1, [brian, ken, dennis], T),
    select(C2, T, [C3]). 
fact1(L) :- 
    select(book(thompson, _, X), L, _),
    select(book(_, X, brian), L, _).
fact2([]).
fact2([book(_, D, C) | T]) :- 
    +=(D, C),
    fact2(T).
solution(X) :- 
    authors(L),
    fact1(L),
    fact2(L),
    select(X, L, _).
```

There isn't space to describe this program in detail, but if anyone has a different version I'd like to see it. Any comments are also welcome.

Now on with the solutions to the puzzles in the last issue. Puzzle Number 2 is a thinly disguised problem in number bases. The number 1,000,000 in base 10 yields 11333311 in base 7. Thus, working from the right, there was one rebate of 1 ECU, one of 7 ECUs, three of 49 ECUs, three of 343 ECUs, three of 2,401 ECUs, three of 16,807 ECUs, one of 117,649 ECUs, and finally one of 823,543 ECUs. I wonder which national group got that!

Puzzle Number 3 can be solved without the need for complicated arithmetic or the value of π . The ball of wire is 24 inches in diameter. A cylinder 24 inches in diameter and 24 inches long contains

half as much again as our ball. Therefore a cylinder 24 inches in diameter and only 16 inches long has the same contents as our ball. We can consider this cylinder as a large number of thin wires (one-hundredth of an inch in diameter) and all 16 inches long. There are $24 \times 24 \times 100 \times 100 = 5,760,000$ of these giving a total length of 92,160,000 inches. Thus the wire is 1,454 miles 2,880 feet in length.

Puzzle Number 4 first appeared in the Sunday Times in 1968. My prolog system expires (out of stack space) when faced with this one:

The seven dwarfs always walk to work in single file and in the same order: Happy, Doc, Sneezy, Bashful, Sleepy, Grumpy, and Dopey. At holiday time two stayed at home, the others from habit went in single file, but never in their working order of march.

One day, five of them set out to build a new house for Snow White, each carrying something useful. She had given each dwarf a different coloured scarf, and Dopey was very proud of his green one, but Bashful would not show anyone his. Sneezy marched immediately ahead of the one carrying the twine. The saw bearer was followed by white scarf.

At the site it was found that no one had brought nails and no one volunteered to go back for them. Blue scarf told Grumpy to go, but Grumpy refused and told him to send the one with the hammer, adding that, as leader for the day, he should have thought of that himself.

The other three gave their opinions. Happy said the one with the twine should go. White scarf said it should be Doc. The one behind him, who today was ahead of two who were normally in front of him, said red scarf should go. Sneezy then said it was daft to suggest sending Dopey and a quarrel began.

When calm was restored, it was found that No. 3 and No. 5 on the day's march had suggested the same person, so he was made to go.

Who was he, and what was he carrying?

UNIX CLINIC (*aka Call Doc Strange*)



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

Olivetti International Education Centre

*with a special guest appearance by
Sam Nelson
sam@uk.ac.stir.cs*

*Dept of Computing Science,
University of Stirling*



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. Nothing much else to report this issue—apart from a trip to Japan. *Ah so deska!*

Sam Nelson escaped briefly into the unreal world of commercial software production (no names, no pack drill) but realised later that it was only a sabbatical appointment. Returning to the fold, he conned some trusting souls out of their superusers' passwords and now divides his time disproportionately between mediocre bass guitar playing, erratic photography and the Intuitionist school of UNIX (system) administration (as in 'Well, it seemed like a good idea at the time...').

The Use and Abuse of Regular Expressions

Dear Doc Strange,

Whenever I start and people look at me in a funny sort of way. Is it my expressions?

Perplexed of Putney.

Dear Perplexed of Putney,

Hmm... Yours is indeed a difficult case, demanding much thought and accumulated wisdom ... methinks I will have to call upon a real expert...

Ladies and gentlemen, a big hand please for your special guest contributor—all the way from Stirling in Scotland—Mr Sam Nelson! Taran-tara!!!)

The standard UNIX¹ System V manual set contains upwards of a dozen command entries which use string regular expressions in one way or another. It never ceases to amaze the author how many UNIX² users on his site know very little about their use. Based on the notions that this is probably the case at many other sites across Europe, and that there are at least some real UNIX users reading this journal nowadays, it seemed worthwhile to describe some of the marvellous things that can be achieved with regular expressions.

Two or three minutes spent working on the right regular expressions can sometimes save several hours editing, condensing an enormously tedious session into perhaps half a dozen commands or fewer. The generalisation of the regular expression system under UNIX System V into C library function calls opens up new opportunities for the simple construction of regular-expression-based applications.

Regular Expression Syntax

For a complete explanation of the syntax of the ordinary UNIX regular expression, there is no better source than the first three pages of the *ed* (1) standard manual entry. Some tools extend or modify this syntax for their own purposes, but the general principles below should be enough to understand what follows:

- A single character in a regular expression string matches itself in the target string. 'Special' characters are generally matched in the usual UNIX way, by prefixing them with the backslash character '\', although there are occasional pathological situations.
- A list of characters in square brackets matches any one of the characters in the list. A shorthand form is also available, thus the list [0123456789] can be written as [0-9].
- A dot '.' matches any single character. A string of single character matches followed by '*' matches any arbitrary list of the single characters it would have matched on its own, including a zero-length string of them.
- A '^' at the beginning of a regular expression matches the beginning of a line; a '\$' matches the end of a line.

Thus [+]*[0-9]* matches any decimal integer; [fF][a-zA-Z]* matches any word whose initial letter is 'f'; and [a-zA-Z0-9_!%]*@[a-zA-Z0-9_.]* matches *almost* any network mail address currently in use.

The above description gives the impression that an attempted match can give only a straightforward 'yes' or 'no' answer. While this sort of result is important, it only scratches the surface. The UNIX editors, *ed*, *ex/vi*, *sed etc*, as well as *grep* and *awk* use this to identify lines in files (no coverage of *awk* is attempted here: there is already a book on the subject!).³ The real power becomes apparent when the regular expression special brackets '\' and '\\' are used in a regular expression string to identify and use interesting parts of the matched string. In the simplest case, *expr* returns a single bracketed substring as its standard output if the match succeeds, so

```
expr arbitrary_integer : '[+-]*[0-9]*\\([0-9]\\)'
```

provides an alternative, quaint method to generate remainder on division by 10 (not recommended for general use!) and

```
expr pathname : '.*/\\(.*)\\..*''
```

is a generalised implementation of *basename* which strips *any* suffix from a pathname rather than just a

1. UNIX is a registered trademark of AT&T in the USA and other countries.
 2. The author is firmly convinced that 'UNIX' is a noun.
 3. Aho, Kernighan and Weinberger, *The AWK Programming Language*, Reading, Mass., 1988 (Addison-Wesley, ISBN: 0-201-07981-X).

single specified suffix. For example, given a pathname of `/staff/colston/lib/junk.old` it will return just `junk`. `ed` allows the identification of a number of substrings in its 'substitute' command for use in the 'replace' string, so

```
s/\(^]*\)\ \(([^]*\)\ \(([^]*\)\)/\2 \1 \3/
```

transforms 'UNIX operating system' into 'operating UNIX system' (note here that a `^` as the first character inside square brackets causes a match with any character *not* present in the list, rather than a match with those that are present).

A real-life example

Frequently the 'glue' that ties together the regular expression operators is as important as the regular expression operations themselves. Consider the following scenario (this was a real-life situation, unfortunately!):

Due to an edict from somewhere above, the order of two parameters in a particular library subroutine has to be changed. This being a very popular and general routine, it is called in dozens of places in the huge tree of source files for which you are responsible. By hand, this requires you to search every source file with the text editor one by one, and actually type the new function call every single time because the parameters used vary widely. One possibility for a mechanical method is:

1. Create a list of files which contain calls of the relevant function (one could simply perform the operations below on all files in the tree, but this efficiency measure is reasonably cheap):

```
find . -type f -exec grep 'fn(.*)' {} \; -print | grep '^.'
```

(If your `grep` can be made silent, the second `grep` is unnecessary.)⁴

2. For any file in a given list, swap the order of the two parameters in any call of the given function:

```
sed 's/fn(\(.*\),\(.*\))/fn(\2,\1)/g' file > temp_file
mv temp_file file
```

3. Combine the two...

```
for f in `find . -type f -exec grep 'fn(.*)' {} \; -print | grep '^.'`
do
    sed 's/fn(\(.*\),\(.*\))/fn(\2,\1)/g' $f > t
    mv t $f
    echo "fixed $f"
done
```

Unfortunately, in real-life this operation had to be carried out on a VMS system with no pattern-matching editor capabilities, and with extremely limited facilities for the creation of a list of files in a tree with a given property. It took almost a day.

Yet another example

This one is not human-language-independent. It would probably fail miserably in German, at least. Suppose you have the text of a book on disk (say 3-400 pages) and your publisher wants a checklist of possible trademarks for crediting. The non-mechanical method is to read the entire book while holding a highlighting pen. A regular-expression-based approach might go as follows:

Assume the text of the book is contained in all of the `TEX` source files in the current directory:

4. In SunOS, this is the `-s` option.

1. Create a list of words in the book, separated by newlines:

```
cat *.tex | tr -cs '[A-Za-z]' '[\012*]' | ...
```

(A better definition of a word could easily be substituted.)

2. Remove from the list all words that aren't proper names and strip out duplicates:

```
... | sed -n 's/[A-Z][a-zA-Z]*/\1/p' | sort | uniq > propernames
```

The list that remains will probably contain more than just the trademarks that need to be credited, but at least the body of text to be searched will have been reduced by a couple of orders of magnitude. The point is that so long as the approximation to the required list is larger, *but not too much larger* than the required list itself, hours of tedious proofreading disappear in a puff of shell script.

Regular expression applications

There are certain weaknesses in the array of regular expression weaponry provided. Consider the following:

1. Suppose you want a list of 'GECOS fields' in the /etc/passwd file that contain a particular pattern. This can be done either by

```
grep '^[:]*:[[:]*:[[:]*:[[:]*:pattern' /etc/passwd | cut -d: -f5
```

or perhaps by

```
cut -d: -f5 /etc/passwd | grep pattern
```

The fact that a pipe must be used here is a pity, because the task is a fairly straightforward one and on a large site the job is not going to complete in microseconds.

2. Take a fairly huge, flat directory of PXL, GF and PK format font files as used by **TEX** and its support tools, and reorganise them into a tree separated by both file type and magnification. Assume the filenames start out in the form

```
<fontname>.<magnification-number><type> e.g., cmr10.1500pxl
```

and are to be deposited in a tree in the form

```
dvifonts/<type>/<magnification-number>/<font>.<type>
```

This may sound a little artificial, but it makes a lot of sense, since file access times can be decreased significantly by using sets of small directories rather than a single large directory. **expr** is underpowered for this task, since it can only return one substring at a time. The preferred method with the provided facilities is probably:

```
for f in *.*; do
  NF='echo $f | sed 's@(.*)\.\.([0-9]*\)\.(.*)@dvifonts/\3/\2/\1.\3@'
  mv $f $NF
done
```

Here, a large number of more or less trivial pipes has had to be used where they really ought to be unnecessary, slowing down the task quite considerably. (**sed** allows any character to be used to bracket match/replace pairs so I have used @ instead of / in an attempt to make it less confusing.)

The **PW** library provided with System V implementations makes available **regcmp(3X)** and **regex(3X)** for the compilation and execution, respectively, of regular expressions inside C programs. Unfortunately, no library function is provided for the substitution into a result string of substrings identified by **regex()**. The construction of such a function, which more or less implements the mechanism familiar to users of, say, the **vi** 'substitute' command, is relatively straightforward. The code for **regsub()** follows:

```

#define EOS      '\0'
#define PARTSUB  '\\'
#define WHOLESUB '&'

char *regsub(src,orig,rret,res,rsize) char *src,*orig,**rret,*res; int rsize;
{
/*
 * Substitute substrings into a template. Arguments are as follows:
 *   src:   the template string. If '\digit' appears, it means 'substitute
 *          here the substring given as element 'digit' of the substring
 *          array 'rret' (qv). If '&' appears, it means 'substitute here the
 *          whole of the original string'. If '\char' appears, it means
 *          'substitute here the given character' to allow for '\' and '&'
 *          in the result string.
 *   orig:  the 'original' string. If 'regsub()' is being used as part of a
 *          regular expression match/replace mechanism, this is the test
 *          string given to regex() to attempt a match against a regular
 *          expression.
 *   rret:  an array of substring pointers. If the standard System V 'PW'
 *          version of regex() was used, this will have ten elements max, and
 *          the substring substitution mechanism can currently only cope with
 *          ten, as it uses a single digit as the index. Obviously, all ten
 *          elements don't have to be filled in---you may not have used any!
 *          Reference to an unfilled substring is, in the best traditions of
 *          such things, 'unpredictable'.
 *   res:   optional buffer for the result string. If NULL is passed,
 *          'regsub()' attempts to malloc() space for the result.
 *   rsize: The size of the buffer passed in 'res'.
 * The return value of the function is a pointer to the result string, if
 * space was found to accommodate it, or NULL otherwise. Note that even in
 * the case of a buffer being passed in 'res', the address of this buffer is
 * still returned as the function result.
*/
char *s,c,*ret,*rt,*osrc=src; /* Address saved for the second pass */
char *malloc();
int count=1; /* Initialised to 1 to count the EOS character */

/*
 * First, zip through the 'source' string, and calculate how long the result
 * string is going to be. This is easy because all the components are
 * available. It also saves problems with running out of space in the middle
 * of a substitution, and makes error returns much cleaner.
*/
while(*src) {
    switch(*src++) {
        case PARTSUB:      /* '\' char---followed by a digit, or not? */
            c= *src++;
            if(isdigit(c))
                count+=strlen(rret[(int)(c-'0')]); /* Add length of substring */
            else
                count++; /* '\char' just means 'char' */
            break;
        case WHOLESUB:     /* '&' char means 'put whole orig. string here' */
            count+=strlen(orig); /* Add length of original string */
            break;
        default:           /* ...otherwise it must be a 'normal' character */
            count++;
    }
}
}

```

```

/*
 * Now, sort out where to put the result string when it's built. The
 * space required is in 'count'.
 */
if(res!=NULL) {           /* Test to see if buffer was offered */
    *res=EOS;             /* Tidy up in case of failure */
    if(rsize<count)       /* Is the buffer big enough? */
        return(NULL);
    else
        ret=rt=res;
}
else if((ret=rt=malloc(count))==NULL) /* Try to get a buffer big enough */
    return(NULL);
else
    *ret=EOS;
src=osrc;                  /* Go back to the start of the source string */
/*
 * Next, perform the substitution. This is the same as the size calculation
 * above, except that the characters are stuffed into the result string
 * instead of just being added to a counter. There are probably some clever
 * optimisations available here due to having just skipped through the
 * string once, but the code looks cleaner this way, and it isn't exactly
 * a CPU-killer anyway.
*/
while(*src) {
    switch(c= *src++) { /* Copy taken of each character, this time */
    case PARTSUB:
        c= *src++;
        if(isdigit(c)) {
            s=rret[(int)(c-'0')];
            while(*rt= *s++) /* Copy the substring into place in the result */
                rt++;
        }
        else
            *rt++=c;          /* '\char' means 'char' */
        break;
    case WHOLESUB:
        s=orig;
        while(*rt= *s++) /* Copy entire original string into place */
            rt++;
        break;
    default:
        *rt++=c;              /* ...otherwise it's a 'normal' character */
    }
}
*rt=EOS;                  /* This may or may not be missing... */
return(ret);
}

```

Now, given `regcmp()`, `regex()` and `regsub()`, the program `reg` which takes source string, regular expression and result template turns out to be about 30 lines of C:

```

#include <stdio.h>
#include <string.h>

#define RETNUM 10
#define RE_SOS "^"
#define RE_EOS "$"
#define RETURN_STRINGS ret[0],ret[1],ret[2],ret[3],ret[4],\
                     ret[5],ret[6],ret[7],ret[8],ret[9]

char *progname, *ret[RETNUM];

main(argc,argv)
int argc;
char **argv;
{
    char *test,*regexp,*template,*comp,*result=NULL;
    char *malloc(),*regcmp(),*regex(),*regsub();
    int i,len;

    progname= *argv++;
    if(argc<3) {
        fprintf(stderr,"usage: %s string reg-expr [template]\n",progname);
        exit(0);
    }
    test= *argv++;
    regexp= *argv++;
    template= *argv++;
    len=strlen(test);
    for(i=0;i<RETNUM;i++)
        if((ret[i]=malloc(len+1))==NULL) {
            fprintf(stderr,"%s: no space for return substring #%d\n",progname,i);
            exit(1);
        }
    if((comp=regcmp(regexp,NULL))==NULL) {
        fprintf(stderr,"%s: regular expression compilation failed\n",progname);
        exit(2);
    }
    if(regex(comp,test,RETURN_STRINGS)==NULL)
        exit(3);
    else if(strlen(template)==0)
        exit(0);
    else if((result=regsub(template,test,ret,NULL,0))==NULL) {
        fprintf(stderr,"%s: template string substitution failed\n",progname);
        exit(4);
    }
    printf("%s\n",result);
    exit(0);
}

```

and does the 'echo | sed' task straightforwardly:

```
reg <font>.<mag-no><type> '(.*)$0\.([0-9]*)$1(.*)$2' 'dvifonts/\2/\1/\0.\2'
```

There is, unfortunately, a significant change in syntax here. The library regular expression compiler `regcmp()` requires that substrings to be extracted be identified by enclosing them within ordinary brackets (...) and suffixing the brackets with \$*digit*, where *digit* identifies the substring buffer into which the result is to be placed. It's more awkward than the original mechanism, doesn't add any expressive power and, really, just means more complication. Examination of the source suggests that doing it this way made life easier for the implementor of `regex()` ...

Just a little more effort gives you `rel`, which does the '`grep | cut`' or '`cut | grep`' job somewhat more simply:

```
rel -d: -i5 -xpattern -o5 /etc/passwd
```

(The text of the program has been skipped because there's quite a lot of it!) The `-d` flag changes the default field delimiter, as in `cut`, `-i` specifies which input field to key on, `-x` specifies the regular expression to use and `-o` gives the output field specification. The program is considerably smaller and faster-starting than `awk` as well as taking considerably less time to learn to use.

The intention here is not to flash around 'look what a clever program I've just written'—neither of the programs presented here need to be particularly clever, because of the availability of strategically useful and general library routines. No-one in their right mind would voluntarily sit down to write regular expression compiler and execution functions.

Having been deeply immersed in regular expression technology at times over the past year or two, the author now has a mailer whose address transformation mechanism views the world's electronic mail networks as the result of the execution of an often-cascaded system of about 250 regular expression match and replace pairs. Whereas `lex` is designed to handle general forms of input with a specific set of regular expressions, the mailer's address transformation mechanism handles fairly specific input forms with a user-configurable set of regular expressions. The author is fairly convinced that there isn't much that can't be achieved with the correct set of tortuous regular expressions.

Odds and ends

Thanks Sam.

Alright the rest of you out there? Got that? (This is Doc Strange back again.)

There's just space to mention some odds and ends. First, on my UNIX system administration courses I usually recommend David Fiedler and Bruce Hunter's *UNIX System Administration* (Hayden, 1986) as being the best book there is on the subject. Just recently, though, Rebecca Thomas and Rik Farrow's *UNIX Administration Guide for System V*, Englewood Cliffs, NJ, 1989 (Prentice-Hall, ISBN: 0-13-942889-5) has appeared and I think it is even better.

Second, I recently bought a copy of the **JetRoff** shareware Documenter's Workbench postprocessor for HP LaserJet+ and Series II laser printers. If you too only have the binary

distribution of Documenter's Workbench and have despaired of getting it to work with your HP laser printer, then I strongly recommend **JetRoff**. It supports all the special characters and drawing commands of the `eqn`, `grap`, `pic` and `tbl` preprocessors, lets you include bitmaps within your document, can print in portrait or landscape orientation, plus much else besides. What's more, as well as the **JetRoff** postprocessor you also get a `troff` preprocessor for making floppy disk and mailing labels, a pretty printer for unformatted text and a wall calendar printer. At \$100 (approximately £60) for complete source it is incredibly good value. **JetRoff** is available from: Rick Richardson, PC Research Inc, 94 Apple Orchard Drive, Tinton Falls, NJ 07724, USA, or e-mail to `rick@pcrat.UUCP`.

I have no idea what next issue's column is going to be about. Any suggestions?

Standards Report on C and POSIX

*Cornelia Boldyreff
cs.brunel.ac.uk!corn*

UK C 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.

The Continuing Saga of the C Standard

The ISO C Standard has been undergoing development in parallel with the ANSI C Standard. When this work was proposed by ANSI as a New Work Item to ISO, ISO in approving the work delegated it to ANSI, and ANSI in turn delegated it to the committee within CBEMA, X3J11, which was developing the ANSI C standard. Thus, ISO members participating in the work on the C standard have been reviewing the work of X3J11 and providing X3J11 with critical feedback so that the C standard is acceptable to the international community at large.

A problem has arisen because X3J11 feels they have finished their work, that the draft standard in its present form is 'good enough' [PLA 88]; but some national member bodies within ISO feel that with just a little more effort, it could be better. Two outstanding issues are:

- Readable alternative to trigraphs—a standard solution is required¹

- Clarity of the standard, in particular, suitability for testing conformance

It seems likely that there will be more work on the standard at ISO level to resolve these issues; however, just a little more work is required and we should have an international standard for C that is usable and testable by the end of the year. At the recent meeting of the ISO Working Group on C in April 1989, it was proposed that the following compromise was proposed: to seek international approval of the latest X3J11 draft of the C standard currently undergoing an ISO Draft Proposal ballot and rectify known errors and omissions by means of an Addendum to the standard.

The X3J11 draft standard is still lacking full approval as a national standard by ANSI in the USA. The X3J11 committee neglected to respond to some comments raised during the Public Review and at their recent meeting which coincided with the ISO meeting, they were obliged to put right this oversight.

[PLA 88] P J Plauger, *STANDARD C Addresses Challenge Of The International Market*, The C Users Journal, Dec/Jan 1989.

1. See "A European Representation for ISO C" by Keld Simonsen in this newsletter.

Future Meetings:

BSI IST/5/14 C Panel	7th Feb 1989 London, England
ISO WG 14 - C	10-11 Apr 1989 Seattle, Washington, USA (This will be a joint meeting with X3J11)
BSI IST/5/14 C Panel	9th May 1989 London, England
BSI IST/5/14 C Panel	8th Aug 1989 London, England
ANSI/CBEMA X3J11	21-22 Sept 1989 Salt Lake City, Utah, USA (This meeting coincides with a DECUS meeting.)
BSI IST/5/14 C Panel	7th Nov 1989 London, England

What's happening on the POSIX Standard Front

The IEEE Standard P1003.1 Portable Operating System Interface for Computer Environments (popularly known as POSIX) has been undergoing a final formal six month ballot at ISO level; if this ballot closes successfully, then the current ISO DIS 9945 will become a full ISO standard. In the UK, the ISO DIS is currently available from the British Standards Institution as a Draft for Public Comment to allow UK experts to have an opportunity to comment on the final draft.

The ISO Working Group on POSIX is anxious to supplement the base standard P1003.1 with the additional POSIX interfaces being standardised by subgroups within the IEEE P1003 committee. To this end, ISO members participating in the POSIX work were balloted regarding subdividing the POSIX work item to accommodate this additional work on the Shell and Utilities, Real-time, Security, Distribution Services, and System Administration as well as the working on various language bindings and a language independent specification of the interfaces. This ballot closed recently and was not unanimously approved; some national members felt that the proposed adoption of the X/Open Portability Guide as a European standard obviated the need for any ISO work related to these supplements. There seems to be something of a misunderstanding here as a European standard cannot hope to address the needs of all the participating international members of the ISO working group on POSIX; the next meeting of the ISO Working Group is scheduled at the beginning of May and I hope to have a more positive report as an outcome of this meeting.

Future Meetings:

1-3 May 89	WG15 - Ottawa, Canada
19 May 89	POSIX Panel BSI, Room 201
18 August 89	POSIX Panel BSI, Room 201
Oct 89	WG15 - Brussels
17 November 89	POSIX Panel BSI, Room 201
Apr 90	WG15 - Paris
Oct 90	WG15 - USA

UNIX Standards: Repertory, Reps and Representation

*Dominic Dunlop
domo@sphinx.co.uk*



The Standard Answer Ltd.

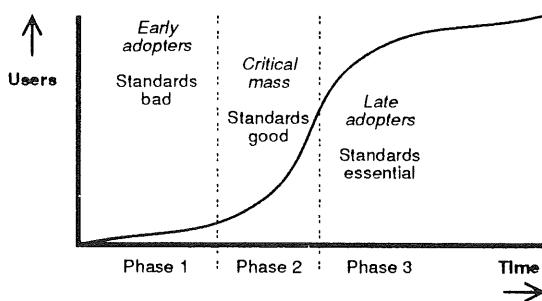
Equipped with an undergraduate degree in Electrical Engineering from the University of Bradford in England, Dominic sidled into the world of mini- and micro-computers. From there, he managed to effect an entry into the hallowed temples of UNIX, and has hung around there ever since, writing the odd paper,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

There's a small repertory company touring the UNIX conferences of the world this year. I've seen its performances several times, and the players always put on a good show when they are on the stage together—even if the audience sometimes gets a little restless during their monologues.

I'm talking, of course, of the public sparring between the Open Software Foundation and UNIX International over the future of the UNIX operating system. Both organisations welcome invitations to present their points of view on any public platform; those who erect such platforms usually invite both organisations in the interest of balance. Or perhaps in the interest of putting on a good show. When this happened at the recent European UNIX Systems Users' Group conference in Brussels, I was lucky enough to be master of ceremonies at a panel session which followed an afternoon of presentations about UNIX 'Standards'*.

* The quote marks appear in the published programme, as not all the speakers were talking of accepted standards, but rather of things that they wished to see accepted as standards.

While there was some interesting dialogue when everybody was on stage together, most of the conference audience thought most of the afternoon's presentations (the monologues) turgid. Presenters with commercial corners to fight (Bob Duncanson from AT&T UNIX Europe Ltd.; Henning Oldenburg of OSF; Morris Schwartz from UNIX International; and John Totman from X/Open) did not help their respective cases by using standard marketing materials which, although highly polished, had not been prepared with an audience of sceptical hackers† in mind. Walter De Backer of the Commission of European Communities put his finger on the reason why those who attend EUUG conferences dislike sales pitches when he presented this graph (which I have modified somewhat):



In a technology's early days, it is adopted by those who can recognise its merits without needing to be sold the concept. At this stage, standards are a Bad Thing, as experimentation is still required before the technology is ready for wider distribution. Wider distribution comes in the second phase, during which it is possible to begin to formulate standards covering the basic aspects of the technology. The more standards that are formulated, the safer new users feel—and the more susceptible they are to sales pitches. Eventually, in the third phase, even those who have been holding out capitulate and catch up with their rivals by adopting the technology, their old excuse that they are waiting for standardisation no longer being valid. By this time, sales pitches are less necessary: the choice of technology is obvious.

The technology that is UNIX has reached the half-way point of phase two, with phase three looming on the horizon. Consequently, various interest groups are wrangling to make sure that it is their particular view of the world which come to be seen as the obvious answer. (That these views differ only slightly in most respects is masked by the melodrama of the confrontation.) Meanwhile, early adopters who have shown the skill, commitment, and stamina which has been needed to bring UNIX to its current position—and who attend EUUG conferences—see standards as constraining their freedom to experiment. They resent people with suits telling them that previously fluid aspects of their favourite systems have been solidified by decree, or have been outlawed by some standard or another. As one unhappy questioner from the floor put it, ANSI had ruined the C language by defining its every aspect. Where now could he discover esoteric features and work on new techniques?[†]

The answer to the general question is that it's always possible to be ahead of the pack, but only if you keep running. While IEEE standard 1003.1-1988 (POSIX) may have nailed down

many aspects of the interface to the UNIX kernel, plenty of other areas are still open for research: administration, particularly of distributed systems, is a case in point, so is security, internationalisation... But you'll have to get in quickly, as already there are standards working groups crawling over these and other topics, inventing technology when they come up against an area where there is little experience with practical systems. They may invent something that you don't like, which you could have told them wouldn't work, or which goes against some precedent you could have told them about. You've got to watch these people!

The EUUG is actually paying me to watch the International Standards Organisation on your behalf as Sub-Committee 22, Working Group 15 moves Posix to world standard status. I will be reporting back to you in these articles in the future (while trying not to tread on Cornelia's toes). Please contact me by email if there is any aspect of that standard which concerns you particularly. I may also be able to point you in the right direction if you want to participate in other standards efforts—such as those for administration, distributed systems, and internationalisation. Again, contact me by email. I shall endeavour to give you your money's worth!

[†] I use the word in its traditional, non-defamatory, sense.

[‡] I suggested he should try C++; but perhaps I was wrong: ANSI has started examining that language with a view to its standardisation. Hewlett-Packard and American Airlines have even founded an organisation, the Object Management Group, to look at standardising descriptions of objects across applications.

Software | Review

Donal Daly

uucp daly@cs.tcd.ie



Distributed Systems Group

Department of Computer Science

Trinity College Dublin

Eire

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

Introduction

Welcome again to *Software / Review*. The purpose of this column is to review public domain software that appears on the net amongst other places. This offering includes a short review on *perl*. Also reviews on *ream* (a mail reader) and *ssh* (a supershell). These packages have not been posted in the source news groups, but I think they are worth a review, as they are very good examples of the quality of public domain software. Following on from this is a review of a very fast *grep*. Some more detailed information on the upcoming new news software is given. Finally an update to the information I presented about archive sites in the last issue. I hope you enjoy this months offering, and remember I actively encourage you to contribute to this column.

PERL

Perl is a interpreted language optimised for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. Its author is Larry Wall (of *rn* and *patch* 'fame'). Perl is a

good language for many system management tasks. The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal). It combines some of the best features of C, *sed*, *awk*, and *sh*, so people familiar with those languages should have little difficulty with it. Expression syntax corresponds quite closely to C expression syntax. If you have a problem that would ordinarily use *sed* or *awk* or *sh*, but it exceeds their capabilities or must run a little faster, and you don't want to write the silly thing in C, then *perl* may be for you. There are also translators to turn your *sed* and *awk* scripts into *perl* scripts.

Perl comes complete with an extensive manual (nearly forty pages). It has a configure script which will work out the system dependencies of your system and edit the *Makefile* and *config.h* as necessary. Compiling and installing *perl* couldn't be easier. There is also a series of test programs provided to ensure *perl* is working correctly. *Perl* also comes with a number of examples (I have a number of other ones which I can send to people also). The way *perl* is packaged should serve as a good example in how all major public domain

packages should be packaged before being released. Larry has also written a configure generator (*dist2* in *comp.sources.unix* volume 16) to help software developers create these magical configure scripts.

From recent email I had with Larry, here are some of the features to expect in the next release of *perl*: (apart from little bug fixes) the support for arrays and array handling has been greatly improved. You can now do arbitrary *iocvtl* functions. Other new functions include: *mkdir*, *rmdir*, *getppid*, *getpgrp*, *setpgrp*, *flock*, *rindex*. Error reporting and syntax checking is improved. There should also be socket and select support for those systems that can support them. *Perl* can now perform greater optimisation on *perl* scripts.

The current release is 2.0 patchlevel 10. *Perl* was published in *comp.sources.unix* volume 15. I found *perl* to be a very useful tool and look forward to its next release. I would recommend it as the language of choice for fast prototype or for simple system management tasks.

SSH

(Paul Dorish jpd@etive.ed.ac.uk)

Like most modern shells, *ssh* features filename completion and command-line editing. The editing, however, is completely configurable—you can either set it up on the fly, or use one of the predefined keysets (the distribution version includes modes for *em*, *vi*, *emacs* and *csh* emulation).

History is also here, and job control too. However, *ssh* goes further than most job control shells, by offering job control on System V! Sxt devices are used to block I/O, giving pseudo job control in much the same way as *shl*, but much more conveniently, since it uses the familiar BSD *csh* system (suspend characters, *fg*, *bg*, *jobs*, etc).

In fact, this System V job control facility is a clue to another important *ssh* feature, which is wide portability. It originally ran under Version 7 (and still does)—but since then it's been ported to more machines than most of us will encounter, under variants of System V, BSD 4.x (and 2.x), Version 7 and a few rather strange ‘homebrews’.

Now, what about those really strange features I mentioned? Well, how about system calls available from shell script? Yup, I've seen network daemons listening on IP sockets and answering requests written as *ssh* script. You can

also redirect to sockets, if you have the right facilities (or FIFOs, if you prefer). And for something really strange, how about a DWIM (“Do What I Mean”) facility? Imagine:

```
$ mroe file
not found: "mroe" -> more
```

I didn't make that up—I included it from an *ssh* window! Before you worry about it translating ‘mr’ into ‘rm’ accidentally, I should add that the cursor sits after the correction, waiting for you to accept, reject or edit it.

If GNU Emacs were a shell, it would be *ssh*. And look how popular Emacs is! [Actually, Simon will hate me saying this, being a confirmed *wi* hacker; but if he can't take a compliment...]

Ssh is available in source form from Simon Brown. Try mailing him at simon@meiko.co.uk.

REAM

(Simon Brown - simon@meiko.co.uk)

Think of how it is when new mail is arriving all the time, but you don't read it because it's just too much effort. Or reading your mail takes up all your free time, so you're threatening to just give up on it. Well, it's not the mail itself that's the problem—it's the mail reader you're using. “But who wants yet another mail reader?”

This is where Ream (Paul Dourish's e-mail reader) comes in. Ream is to other mailers what ‘vn’ is to ‘readnews’ for reading Usenet news! It provides an environment specially designed for dealing with mail in the most flexible way possible.

Ream presents the user with a screen of mail header information, allowing quick and easy access to all of your mail simultaneously. It provides a very sophisticated message naming scheme to allow any mail message, or group of messages, to be dealt with in any of a vast number of ways. This is particularly useful if you receive a large amount of mail—you can see at a glance what mail you've got, and deal with it accordingly.

The most common problem with mail reading tools is that they hide too much detail of the mail mechanisms, causing unbearable frustration for those dedicated mail enthusiasts who love nothing better than looking through pages of routing information. But the other extreme is equally unpleasant—nobody can use /bin/mail as a mail

reader for very long and still keep a remnant of sanity. What is needed is a highly flexible tool that can be tailored exactly to an individual's needs, but which can still be used in a raw default form by novices.

So Ream is more than a simple mail reader—it provides an entire environment dedicated to the efficient and easy manipulation of mail. Mail folders can be loaded in or written out, just as an editor deals with files. Pattern matching is available, whereby messages may be selected by any of a number of attributes. You want to save all messages coming from a particular person into a particular file? Easy—a few keystrokes will do the trick. You want to format the header fields of mail messages you send out, in a particular style of your own? Easy—a few lines in your personal configuration file will do it for you. You want to live in safety and be sure of never losing anything by typing the wrong thing by mistake? Easy—Ream can look after you and query commands that seem dangerous, if you want this. If you don't, then of course you don't get it.

This is the whole philosophy of Ream—it's all environments to all people. Nothing is hard-wired, everything is programmable. But the defaults (or should I say the 'default defaults'?) are arranged so as to make it usable from the very start.

Ream has been widely ported—it runs under both Berkeley and System V UNIX, on a vast variety of hardware, and can be used with either Sendmail or MMDF (the two most common UNIX mail delivery agents).

Ream is available in source form from Paul Dourish at the University of Edinburgh Computing Service, where Ream is now the standard supported mail tool. Try mailing him for details—his email mail address is *jpd@etive.ed.ac.uk*. You can be sure of a prompt reply—after all, although he receives more mail per day than most people see in a month, he has the most sophisticated mail-reading environment you could hope for at his disposal...

Woods/Spencer Grep

(Mike Selway *mas@ssl.uucp*)

Everybody who is really in the know, knows that egrep is the fastest of the three greps. The real scientists among us know that egrep is faster because they've actually timed it. And the figures

are impressive: for a simple dictionary search on my machine, taking grep as 100%, fgrep takes 82% and egrep takes 42%. That saves a lot of time if you're searching through megabytes of disc space.

Compare this with wc at 41% and it looks like egrep is close to optimum.

I had been aware of various greps which had emerged in recent times which were supposed to be quite fast: Andrew Hume's paper at the London EUUG conference described work he had been doing to speed things up. However, when someone sent me a copy of the Woods/Spencer grep from *comp.sources.unix* I wasn't really prepared for the results I got.

The simple search which egrep cuts to 42% is cut to just 17%. On one of our development machines this grep runs at the same speed as cat if the pattern matches every line. The inner loop of this program compiles to just 4 instructions on a 68020, and it runs faster on a Sun 3 than standard egrep on a Cray 2.

Woods/Spencer grep has some very fast code for spotting exact matches which it uses to determine whether the line is worth presenting to the regular expression code. If the expression is very complex or you invoke some obscure options, or, indeed, if it notices that it's taking a long time to get through the first bit of the file, grep will just exec the most suitable standard version from /bin or /usr/bin. In this way it limits its worst case behaviour. Unfortunately, if your standard grep family takes a different set of flags from those that the new grep expects it to take, you can get some odd error messages:

```
$ newgrep -i 'a*b*c*d' /usr/dict/words
newgrep: illegal option -- i
usage: egrep [ -bcInv ] ...
```

A GNU grep is now available which I'm told is a later version of the Woods/Spencer grep, but without the exec feature.

Woods/Spencer grep is available for UKNET sites in the UK from the *comp.sources.unix* archive on the Kent info-server as *volume9/fastgrep/part01* and *volume9/fastgrep/part02*.

NewsFlash

Here is some more detailed information on the upcoming new news software (really I just wanted

to use that heading again :-)). At the time of writing neither of the news software packages has been released. The overviews of the new news software are by the authors themselves.

C News

(Geoff Collyer geoff@utstat.toronto.edu)

'C News' is a rewrite of the transport and expiry subsystems of the Usenet news software. It is approximately compatible with B News 2.11, which is the current production version in most places. In particular, it preserves full compatibility in the way articles are stored and in the format of control files, so that existing news-reading programs work with it. C News doesn't include any new news readers, in fact: it has the Australian reimplementation of 'readnews' for naive users, and 'rn' for sophisticated ones. Various other news readers have been tried with it; they all work.

The main benefits of C News are performance and robustness. It was written to try to solve some of B News's performance problems. It's fairly successful at that; the current C News processes incoming news at roughly 25 times the speed of B 2.11. On a fast machine like a Sun 3, C News's performance approaches the theoretical ultimate, since a large fraction of its time is spent in system calls, and considerable pains have been taken to minimise the number of calls needed.

C News offers some other minor benefits. It uses shell files wherever possible, which makes maintenance simpler. Its 'expire' runs from a control file which allows specifying things like expiry times on a per-newsgroup basis. The code is generally fairly cleanly written, unlike the unholy mess inside 2.11.

At the moment, the main disadvantage of C News is that it is in a private beta test. An 'alpha' release was distributed via *comp.sources.unix* a while ago, with some minor updates sent out in *news.software.b* later. The alpha release runs fairly well, but it has a number of annoying quirks and minor problems and is *very* poorly documented. A definitive release has been 'in the works' for some time, and we hope to actually get it out the door this spring.

C News was written primarily by Geoff Collyer and Henry Spencer at the University of Toronto. It is unrelated to B News 3.0, a separate project which is also underway right now.

News 3.0

(Eric S. Raymond eric@snark.uu.net)

The software that will be 3.0 netnews is in final beta at sixty sites now, including one in England. It is a complete rewrite of the news suite, including both an improved transport layer, a complete suite of readers, tools and service libraries for building custom interfaces, and extensive documentation. The code (which has been reorganised as a stack of abstract data types in a way somewhat analogous to the ISO 7-layer networking model) is 8-bit clean.

Notable new features include: full support for conversation-following in the readers (using the followup-to relation and its inverse); a customisable summary-page mode that assists in filtering a lot of news rapidly; rn-style kill-language processing in every reader; a '-' command that really works (one can back up clear to the beginning of a session along the conversation track, then forward again).

Also, setup and administration have been greatly simplified. Much closer control of article expiration is supported. Support for multicasting and exotic point-to-point layers is improved. Extensions to the subscription syntax make specification of complex partial feeds less cumbersome.

Though completely functional in itself and upward-compatible with older news versions, the rewrite is also the first step in a plan for implementing planetary-scale distributed hypertext using the netnews protocols as a base. For further information and access to beta sources, contact the author, Eric S. Raymond, at eric@snark.uu.net.

Archive Servers

Quite a few of the responses I have had to my first article went something like this "Can you send me this public domain software" or "How can I get hold of these public domain sources?"

Firstly a lot of this public domain software is available in the form of EUUG tape distributions. Contact Frank Kuiper (euug-tapes@cwi.nl) for more information. The latest conference tape from the Brussels conference contains the most recent versions of the GNU sources.

The EUnet backbone managers are currently investigating how a EUnet wide archive service can be set up. This would be available to you via a simple email message to your local backbone site. I will publish more information when details have been finalised. For now, here are details of two backbone sites offering an archive site. Thanks to Keld Simonsen (keld@dkuug.dk) and Peter Houlder (uknet@ukc.ac.uk) for the information.

Denmark

In Denmark there is a mail-based news and source archive service reachable at archive@dkuug.dk. Mail it with Subject: help for further information. The archive contains news and selected portions of the latest EUUG tapes (GNU and X excluded). It is restricted to use within Denmark for accounting reasons.

England

In England there is a mail-based source archive service reachable at info-server@ukc.ac.uk. This server contains several public domain system including the news code and *comp.sources.unix*. There is also a server at netdir@ukc.ac.uk which will return you the uucp map information for a site specified in the subject line of the message. For more information about UKnet archive services see the last EUnet column.

Next Issue

As ever, I am interested in comments if you found anything useful or interesting. Remember I actively encourage you to contribute to keeping this column alive, so email me your reviews of the latest and greatest piece of software you use. I would really like to get some reviews on any of the GNU software. I will gather them together and publish them in a special section on GNU, hopefully in the next issue.

Collective noun phrases from the computer field:

- a batch of mainframes
- a deck of VAXes™
- a clique of workstations
- a double clique of Macintoshes™
- a session of terminals
- a stack of programmes
- a heap of programmers
- a class of object-oriented programmers
- a relationship of database designers

The list can certainly be augmented and improved!

Best regards,

Markku

Markku Sakkinen
I.markku@jytko.jyu.fi
 Department of Computer Science, University of Jyväskylä
 Seminaarinmaa 15, SF-40100 Jyväskylä, Finland

USENIX Association News for EUUG Members

Donnalyn Frey
donnalyn@frey.com

Frey Communications



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

1989 Summer USENIX Association Conference and Exhibition

The USENIX Association's 1989 Summer Conference and Exhibition, held June 12-16 in Baltimore, Maryland, was a success. Over 3,500 people attended the conference. The first two days were devoted to tutorials, with the next three days for technical sessions. The technical exhibition was held on June 13, 14 and 15. Bill Wulf, of the National Science Foundation, presented the Keynote Address entitled "What Will We Do With All Those Cycles In The Year 2000?"

Tutorials were presented on many subjects, including MACH, 4BSD TCP/IP performance improvements, internals of the GNU C compiler, security issues, Postscript, AIX technology, System V internals, network programming, C++, software contracts, and programming in the X Window System.

Technical papers were presented on streams and the kernel, networks, administration, windowing systems, program development, file systems, security and performance issue. A Work in Progress session was held on Thursday afternoon to preview new work not yet ready for publication.

Security

Two full sessions were held on computer security, including a paper entitled "Who Can You Trust?" by Thomas Malarkey of the National Computer Security Center. Other papers discussed controlling gateways, security testing, and B1 security portability.

Terminal Room

The USENIX Association again sponsored a terminal room at the Baltimore Conference. Attendees could read their mail or contact their offices from terminals at the conference.

FaceSaver

The FaceSaver returned to the USENIX Association Conference. The FaceSaver was operating in the vendor exhibition area, with on-line access to faces on uunet. Attendees were able to have their faces saved and uploaded to uunet.

To request copies of the Summer 1989 USENIX Association Conference Proceedings, send email to {uunet,ucbvax}!usenix!office or office@usenix.org.

1990 Winter USENIX Conference

The 1990 Winter USENIX Conference will be held in sunny Washington, DC on January 20-26, 1990. The first two days will be devoted to tutorials, with the next three days for technical sessions. For further information on the 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 November 16-17, 1989 at the Doubletree Hotel in Monterey, California. For information on these workshops, contact the USENIX Association Conference Office at 22672 Lambert Street, Suite 613, El Toro, CA 92630, USA.

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 its new location at 22672 Lambert Street, Suite 613, El Toro, CA 92630, 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 1990 C++ conference and the semi-annual technical conferences. A schedule of upcoming events includes:

Postscript

The USENIX Association regrets to announce the departure of Dr. Peter Salus, outgoing Executive Director. Dr. Salus has taken a position as Director of University Relations at the Open Software Foundation. The Association is pleased to announce the appointment of Ellie Young as the new Executive Director. Ms. Young has been the Assistant Executive director for much of 1988 and into 1989. The USENIX Association is looking forward to a rewarding working relationship with Ms. Young.

The USENIX Association also announces that it is changing its mailing address. The actual office is not moving. However, the new mailing address is:

USENIX Association
2560 Ninth Street
Suite 215
Berkeley
CA 94710

The telephone number: +1 415 528 8649 will not change.

The electronic mail address remains as:

office@usenix.org

USENIX Association Future Meetings

Date	Location	Topic
1-2 May 1989	Pittsburgh, PA	Transaction Processing Workshop
12-16 June 1989	Baltimore, MD	Semi-Annual Conference & Exhibition
7-9/8 September 1989	Austin, TX	System Administration III Workshop
5-6 October 1989	Ft.Lauderdale, FL	Distributed Systems Workshop
16-17 November 1989	Monterey, CA	Graphics Workshop V
23-26 January 1990	Washington, DC	Semi-Annual Conference
11-15 June 1990	Anaheim, CA	Semi-Annual Conference and Exhibition
22-25 January 1991	Dallas, TX	Semi-Annual Conference
10-14 June 1991	Nashville, TN	Semi-Annual Conference & Exhibition
20-24 January 1992	San Francisco, CA	Semi-Annual Conference
8-12 June 1992	San Antonio, TX	Semi-Annual Conference & Exhibition

UNIX System V Data Networking Architecture

Laurence M Brown

AT&T



Laurence Brown is a supervisor in the Network Systems Engineering and Architecture Department at AT&T in Summit, New Jersey. He has worked on the design of operating systems, computer networks and database management systems and was responsible for the architecture of the networking features of the UNIX System V Release 3. He is currently responsible for the architecture of full OSI protocol stacks using the STREAMS feature of UNIX System V Release 3. He holds an M.S in Computer Science from Stanford University.

please contact Janet Davis <janet@uel.uucp> for more information on the contents of this column.

The computer industry is converging on two communications standards—Open System Interconnection (OSI) and Systems Networking Architecture (SNA). OSI will be used in multivendor environments such as factory automation, office automation and telecommunications. OSI is at the heart of the AT&T Data Networking Architecture. SNA will be used to communicate with corporate mainframes. UNIX systems must be able to communicate fluently using both of these standards and must provide a smooth migration path to these standards.

The STREAMS facility of UNIX System V Release 3 provides a powerful tool for handling these problems. This paper describes the properties of the STREAMS technology that makes it ideal for the implementation of communication protocols. The STREAMS facility of UNIX System V Release 3 provides a flexible and efficient mechanism for solving a variety of communications problems.¹²³ This article describes how the STREAMS facility can be used to solve challenging networking problems. The first section of this article examines trends in networking that present both challenges and opportunities for UNIX System vendors. The second section identifies the attributes that the preeminent networking operating system. The third describes how the STREAMS technology can be exploited to provide

these attributes. The final section shows how popular networking architecture can be implemented in a consistent and compatible fashion using the STREAMS facilities.

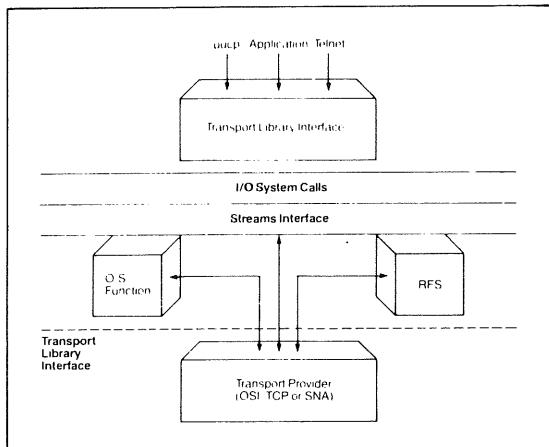


Figure 1. Transport Layer Interface

The Transport Library Interface is a shared library, used by applications and standard networking utilities. These applications and services are made protocol-independent through TLI. Any transport provider (OSI, TCP or SNA) can be used to provide the transport service.

Trends in Networking

The successful computer vendor today must be able to cope with a wide variety of communications standards. In broad terms, the communications standard required is determined by the target market. In the scientific, academic, defence and computer-aided design (CAD) markets, the Transmission Control Protocol/Internet Protocol (TCP/IP) defined by the U.S Department of Defence and popularised by the ARPANET (Advanced Research Project Administration Research Network) and 4.2 BSD (Berkeley System Distribution—a derivative of the UNIX operating system) are predominant. Office Automation is currently an unstable market in terms of communications standards, but products based on XNS for Xerox have the largest current market share. Factory automation requires the OSI protocols as specified in the Manufacturing Automation Protocol (MAP) standards. IBM's Systems Network Architecture (SNA) has the major position in the Back Office Market. The successful vendors will be the ones who can handle diverse standards with consistency and provide smooth migration to a more orderly future.

In the future, OSI and SNA standards will survive at the most prominent. Each will have its own sphere of operation and the successful vendors will be the ones who can provide interworking between them.

OSI

OSI protocols will be the standards of choice in any open market. AT&T's Data Networking Architecture is based on OSI. Included in this category are markets that are inherently multi-vendor such as Office and Factory Automation and markets where government policy preclude single vendor dominance.

The OSI protocols are a good choice for these markets since the OSI protocols are open standards that have been designed and evolved through a consensus the last five years, the OSI protocols now represent a complete solution to current networking problems and provide a solid architecture on which to evolve future solutions. Thanks to the efforts of the Corporation for Open Systems (COS) in the United States and the Standards Promulgation and Acceptance Group (SPAG) in Europe, there are implementable subsets of OSI that customers are demanding from

the vendors.

SNA

SNA is popular in the Back Office because of the large investment in IBM architecture machines that speak SNA. Any vendor who wishes access to the valuable information stored on these corporates mainframes must be prepared to speak SNA.

Challenges for UNIX System V

To deal with the evolution of these networking standards, UNIX System V must be very nimble in its support of networking. The basic challenge is to support all of these different protocol suites in a consistent manner that will allow vendors to support multiple protocol standards and allow customers to migrate from one to another without rewriting their applications. This presents specific challenges for UNIX System V.

Application Portability

The first challenge for System V is to ensure the same degree of portability for networking applications that the System V Interface Definition guarantees for traditional applications. For networking applications it is important to provide portability across protocol suites as well as across machines. The need is for a standard interface to networking services that is independent of the protocol that provides the service. This means that an application written to this interface must work whether the underlying network is SNA or OSI or TCP/IP or XNS. In order to have a precise and unbiased model of networking, this interface should be based on the service definitions of the OSI reference model.⁴

Protocol-Independent Networking Services

The second challenge involves providing the useful networking services that customers need in a protocol-independent manner. Included in this set are traditional networking services such as electronic mail, file transfer, and remote execution, as well as industry specific services such as videotex and electronic funds transfer, and also traditional operating system services such as the file system or process management distributed across the network. Traditionally these services have been constrained to work over a single set of protocols. However, if UNIX System V is to be an open communication environment, it must provide a uniform set of networking services that

work in any protocol environment.

Protocol Implementation Environment

The third challenge is to meet the needs of the protocol implementors. The operating system must provide the muscle and sinew necessary to implement modern protocol architectures in a modular and reconfigurable fashion.

Protocol Substitution. The first requirement is to support protocol substitution—that is the ability to dynamically substitute one protocol, say TCP, for another functionally equivalent protocol, say ISO Class 4 Transport. This is the facility that is needed to support a protocol independent application network services. In addition, this facility is needed if a vendor needs to use a single protocol, such as X.25, in several different protocol architectures, such as SNA, TCP/IP and OSI—all of which require X.25 wide area networking.

Protocol Portability. The second requirement is that the operating system should provide an environment that allows protocol modules to be portable in the same way that applications are portable today. This hasn't been the case previously. For reasons of efficiency and timing, protocols have been implemented in the UNIX System kernel, but kernel code is not typically portable. Any manufacturer who has a wide range of machines would benefit from being able to rapidly port a single communications package across the entire line. In addition, portable protocols could be useful to the computer industry as a whole. An inordinate amount of time is spent in verifying protocol correctness and multivendor interoperability. If there were standard reference implementations of the protocols that any vendor could port, the need for extensive multivendor testing would be reduced.

Internetworking. The nature of modern communication architectures present the third requirement. To support internetworking, the kernel must facilitate the routing of data to multiple device drivers. The traditional UNIX System I/O architecture has a linear connection between the file descriptor and the device driver. Protocol developers have had to use tricks to program around this constraint. The operating system should make this task easier.

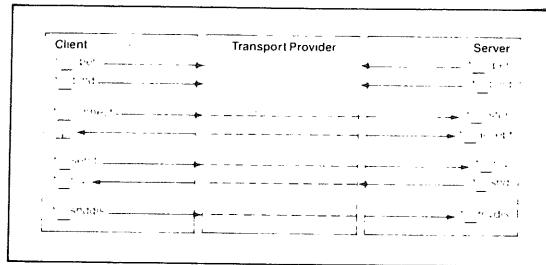


Figure 2. Client /Server Interaction Using TLI
The client who makes a call and the server who responds use a set of primitives to talk to each other. Both client and server must open the Transport Provider and bind to a local address. The client issues a connect request, and the server listens for incoming connect requests on that address. Once the virtual circuit is established, either side may send or receive data until a disconnect occurs.

Meeting the Challenges with STREAMS

How STREAMS technology can be used to meet these challenges will be discussed in the following paragraphs. STREAMS is designed to support a multitude of protocols of diverse functionality and to provide broad support for networking and internetworking systems.

The STREAMS framework does not impose any specific network architecture but enables various protocols to be easily implemented. Further, it facilitates the development of applications independent of the underlying protocols.

Common Application Interface

The level at which a common application interface can be defined depends on the characteristics of the protocol suites that must be supported. Given the currently popular protocol suites—SNA, TCP/IP, OSI and XNS—this interface must be at the Transport Layer in the OSI reference model. Transport protocols provide reliable machine to machine communication independent of the underlying network topology. In a sense, this is the fundamental networking service. It is the foundation for all higher level services. It would be undesirable for an application or a file transfer protocol to have to worry about the details of error detection and recovery or routing. Since Transport is such a

valuable service, every major protocol family has a Transport protocol. In XNS it's SPP. In TCP/IP it's TCP. In SNA it is provided by the basic conversational verbs of LU6.2. In OSI there are five different transport service, but they differ in the assumptions that they make about the underlying network layer protocols.

The strategy in System V Release 3 is to provide an application interface to Transport Services that is based on the formal OSI definition of Transport Services.⁵

There are three types of OSI documents. The seven layer OSI Reference Model gives the general framework for modern networking architectures. It defines terms and partitions networking functionality into distinct layers. It provides the tools necessary to meaningfully compare protocol architectures. However, it does not precisely define the services provided by each layer. The second OSI document type, the OSI Service Definitions, defines in precise terms the obligations of a protocol at each layer of the Reference Model. There is a potentially infinite set of protocols that can provide these services, but the consumer of the services has an unambiguous definition of the services that any of these protocols would provide. The final level of OSI documentation is the Protocol Specifications themselves. A Protocol Specification defines the operation of a particular protocol that meets an OSI service definition. The strategy in UNIX System V is to base the application interfaces for networking services on the OSI Service Definitions. These interfaces are defined as 'C' language library interfaces through which an application can request networking services at the appropriate layers in the reference model. The customer can then position under this interface any protocol that provides the desired service.

The UNIX System V Release 3 library for Transport Services is called the Transport Library Interface (TLI). TLI is a UNIX System user level shared library through which an application can request reliable data communication. This interface has been engineered so that it will work with all the popular protocol suites. The result is that an application developer or the developer of a networking service can write a program and be guaranteed that it will work without modification over SNA or OSI or TCP or XNS or any other provider of reliable data communications. TLI provides the foundation for protocol-independent

applications in UNIX System V. It is specified as a standard interface to networking services in the System V Interface Definition (SVID)⁶ and, as such, is included in the System V Application Operating Environment (AOE).

TLI provides an essential service that will be used by application developers for many years to develop applications that will work in a wide variety of networking environments. However, the time will come when it will be commercially important only to write applications that will run in the preeminent protocol environments—OSI and SNA. At this time, it will be appropriate to move up in the Reference Model to the application layer—specifically to the ISO TPE /SNA LU 6.2 level. OSI and SNA have a common service interface to Transaction Processing services at this level. In AT&T, we are currently defining a UNIX System Library interface that will give an application access to Transaction Processing services in both SNA and OSI. Using STREAMS, the appropriate protocol suite can be substituted under the interface to communicate with the customer's network.

A detailed look at TLI will show how these Service Interfaces work. TLI provides three groups of services. The first is 'Common Management Primitives' that are used to bring the application into communication with the local Transport Provider. Through this interface the application can open the Transport Provider, bind itself to a local address, and negotiate any protocol options or discover any boundary conditions.

The second group of services is 'Connection-Mode Transport Services'. These services are only provided by protocols that support reliable virtual circuit service. Included here are TCP, SPP (from XNS), the five classes of OSI Transport Protocols and LU 6.2 (from SNA). Through these primitives, an application can open a virtual circuit, send and receive data from the circuit, and release the virtual circuit.

The third group of services is 'Connectionless-Mode Transport Service Primitives'. These are unreliable datagram services that are provided by UDP from the DARPA protocols and the OSI Connectionless Transport protocol.

Figure 1 shows how the Transport Library interface (TLI) looks in the UNIX system. TLI is a shared library. An application, or a network

service such as uucp or telnet, can receive Transport Services by making calls to the library interface routines. The TLI Library communicates these services requests to the Transport Provider by making STREAMS system calls. The Library uses getmsg and putmsg to send requests to the provider and to receive results from the provider. The provider sees these requests as STREAMS messages. Using the dynamic nature of STREAMS, any provider can be plugged under the service interface to provide the Transport Service.

The messages of the Transport Provider Interface (TPI) can also be generated directly by an object in the kernel sitting upstream from the Transport Provider. The UNIX System V Release 3 Remote File Sharing facility uses this capability to achieve its protocol independence.⁷ This same technique could be used by a higher level protocol in the kernel to become transport independent.

Figure 2 shows the actual primitives that two applications would use to talk to each other over an arbitrary network using reliable virtual circuits. There are two parties: the client who makes the call and the server who responds.

The client first opens the Transport Provider and binds itself to a local address on that provider. This is the common management phase. Once bound, the client can issue a connect request. When the connect request returns, a virtual circuit has been established. The client can send and receive data over the virtual circuit. The conversation is terminated when either side sends a disconnect request. All of these primitives are part of Connection-mode Transport Service.

The server must also first open the Transport Provider and bind itself to a local address on the Transport Provider. It is this address that the client gives in the connect request. Once the server has bound itself to an address it can then listen for incoming connection requests on that address. When the server accepts a request, the connect request on the client side returns and the virtual circuit is established. Either side may now send or receive data over the connection until the disconnect occurs.

Standard Services

A standard set of protocol-independent end-user networking services can be provided by implementing the services to TLI. Figure 1 also shows three services that have been made protocol

independent through TLI. UUCP is the standard UNIX system utility for file transfer, terminal emulation, remote execution and mail. Previously, it was limited to operation over dial-up lines. Through the auspices of TLI it works over any valid Transport Provider in System V Release 3. Telnet is the standard DARPA service for terminal emulation. Traditionally it has worked only over TCP/IP, but through TLI it can work over any transport protocol. The same has been done for the other DARPA services -FTP for file transfer and SMTP for mail. Since RFS has been implemented to the kernel TPI, customers can interactively share files over any of the commercially important protocol suites. All the networking services in the AOE are protocol-independent since they have been implemented to TLI.

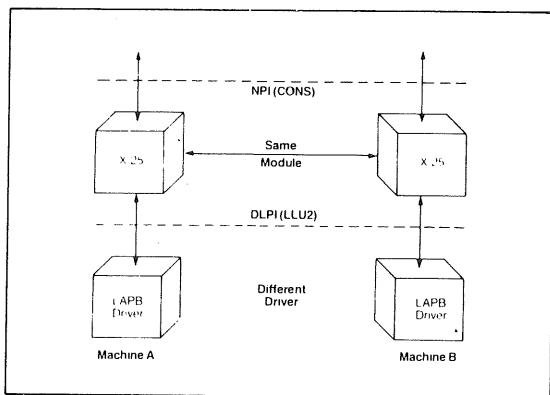


Figure 3. Protocol Portability

A protocol implemented in STREAMS is portable to any machine that supports the STREAMS environment and has a STREAMS provider that provides the services that a protocol requires. In this figure, an X.25 module has been ported to Machine B.

Protocol Substitution

Protocol Substitution is a straightforward application of the service interface concept. Figure 1 shows an operating system function, for example, process management, that is being implemented to work over a network. It is not acceptable to constrain the function to work over only one network. So instead of making calls directly to the network as would be done in System V Release 2 or 4.2 BSD, the function is implemented using STREAMS technology.

When the function requires a network service, it sends a service request downstream. These service requests can be fielded by any module that understands them. So, the distributed process manager can work in an OSI network or a DARPA network or an SNA network. This same concept can be used at any layer in the reference model. For example, a customer can migrate from MAP protocols to Technical Office Protocols (TOP) simply by substituting an 802.3 CSMA/CD protocol module for the functionally equivalent 802.4 Token Bus module.

Protocol Portability

One of the nice things about STREAMS is that it isolates the protocol module from its surrounding environment. This leads to protocol portability. A protocol module can be ported to another machine by a STREAMS environment on that machine and providing modules downstream that provide the service that the module requires. In Figure 3, an X.25 module has been ported to Machine B by providing a STREAMS environment on B and writing a STREAMS device driver on Machine B that controls the LAPB hardware that X.25 needs. Since the device driver is hardware dependent it is not portable.

This protocol portability has major ramifications for computer manufacturers. It means that all protocols, except those implemented in hardware, are directly portable across all machines that use the UNIX system.

This also has major significances for the software industry it means that ISVs can produce protocol modules that are portable to any vendor's machine in the same way that System V applications are currently portable.

Internetworking

Internetworking illustrates the power and flexibility of STREAMS. Internetworking is the problem of connecting several dissimilar physical subnetworks into one logical network. The solution is to use a common internetworking protocol to route data from one subnetwork to another until it reaches its final destination.

There are two popular styles of internetworking. In the connectionless school, the internetworking protocol routes individual pieces of data from one subnetwork to another. All internetworking protocol asks is that the subnetwork be able to

send the datagram from one edge of the subnetwork to the other (that is, for example, to send a datagram from one node on an Ethernet to another).

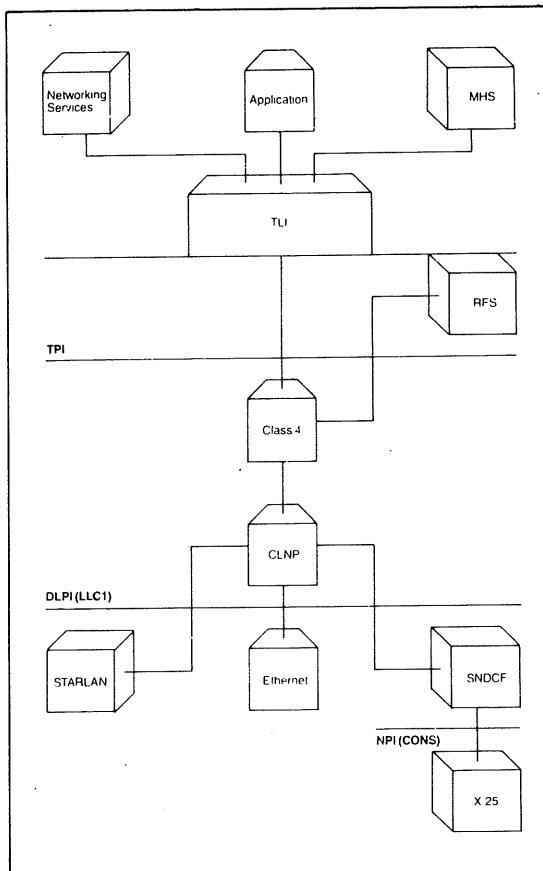


Figure 4. OSI Today

An implementation of OSI Transport X Network Layer Protocols in STREAMS provides the ability to internetwork between STARLAN and ETHERNET style local area networks and X.25 wide area networks. CLNP is the connectionless internetworking protocol, and ISO Class 4 transport protocol is the transport provider. Standard UNIX system applications as well as RFS and other network services access the transport provider directly through the TLI.

In the connection-oriented school of internetworking, the internetworking protocol expects the subnetworks to provide reliable virtual circuits. The job of the internetworking protocol is simply to weld the individual virtual circuits to an end-to-end path.

The bottom portion of Figure 4 shows a connectionless internetworking scheme as used in TCP/IP, XNS and MAP/TOP. The way that

STREAMS technology is used would be the same in a connection-oriented example. In this connectionless scheme, CLNP (Connectionless Network Protocol) is the internetworking protocol. It provides a Connectionless Network Services (CLNS)—that is an unreliable end to end datagram service. It relies on each of its subnetworks to provide a single-hop datagram service as specified in the IEEE 802.2 local area network standards.⁸ CLNP makes routing decisions based on the internet address of the

target end system. It decides, based on this address, where to send the datagram to take it to the next stage towards its destination. For example, if the datagram originated on the 802.3 (CSMA/CD) LAN and needed to go across the X.25 WAN before reaching the target on the 802.4 (Token Bus) LAN, CLNP on the router on the 802.3 LAN would send the datagram across the X.25 subnetwork to the router on the 802.4 subnetwork. The implementation of this routing function is made easy using STREAMS.

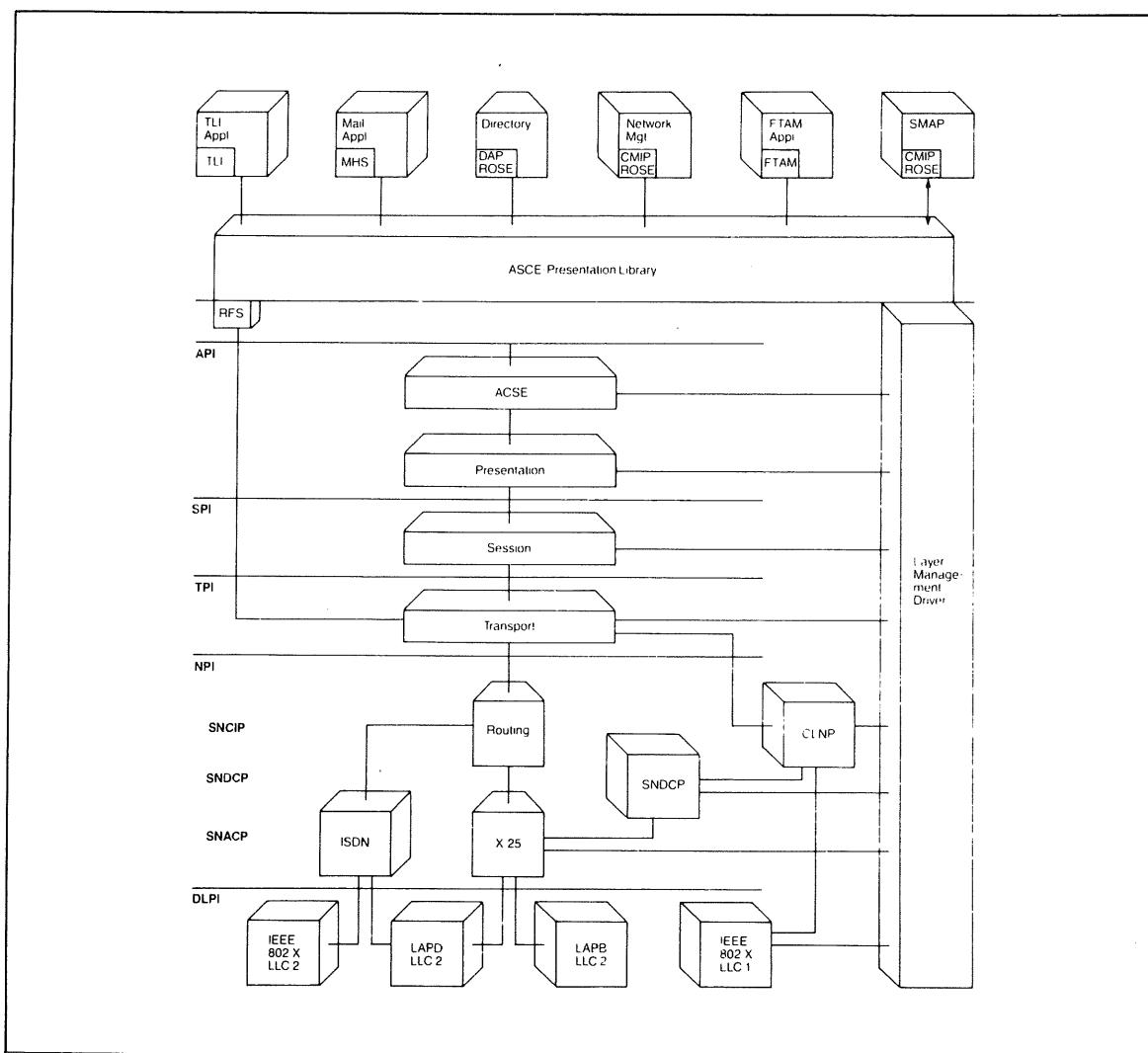


Figure 5. OSI Long Term

STREAMS will enable ISDN to be either integrated as a subnetwork under CLNP or be used as a provider of network layer service. The upper layer Protocols will be implemented directly over the Transport Provider Interface.

CLNP maintains a table that maps an internet address into the stream leading to the appropriate subnetwork. When it receives a datagram it looks at the internet address and sends the datagram down the appropriate stream with an 802.2 single hop datagram service request. It doesn't care that these subnetworks are physically different. All it knows is that each one supports the 802.2 service interface. The drivers for the LANs provide this service directly; while for X.25 a function must be provided that maps the single hop datagram service requests into the virtual circuit requests that X.25 understands. This convergence function (SNDCF in Figure 4) is a virtual circuit manager. When it receives a datagram service request, it looks to see if it has a circuit open to the target system. If it does, it sends the datagram directly over the circuit. If it doesn't, it must first open a virtual circuit. For the sake of economy, it may wish to shut down some less frequently used circuit. This convergence function is specified by the OSI reference model. The good thing about STREAMS is that neither CLNP nor X.25 is affected by this convergence. CLNP treats the combination of X.25 and the SNSDF as a connectionless subnetwork. X.25 keeps a pure connection-oriented view of the world. The same X.25 module could be used to participate in a connection-oriented ISDN or SNA internetworking scheme or be used directly to provide reliable virtual circuits over a public Packet Switching Network (PPSN).

Protocol Architectures

The System V AOE specifies support of the three major commercial protocol suites: OSI, TCP/IP and SNA. The following sections described how these protocol suites are implemented in STREAMS.

OSI

Figure 4 shows a STREAMS implementation of the currently stable OSI protocols. It provides the ability to internetwork between STARLAN and ETHERNET style LANs and X.25 PPSN WANs.

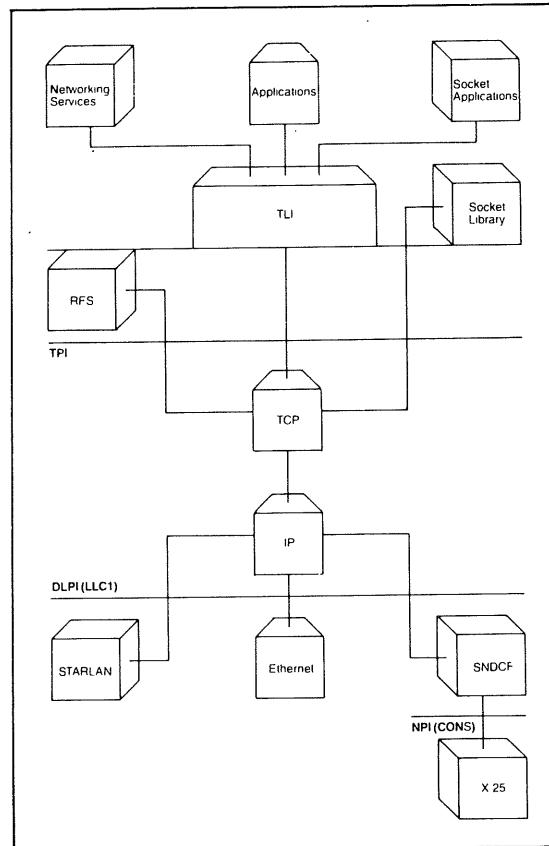


Figure 6. TOP/IP Protocols

The Transmission Control Protocol/Internet Protocol (TCP/IP) also provide internetworking between local area networks like STARLAN and ETHERNET and wide area networks like Telnet and ftp are implemented directly to the Transport Library Interface. This configuration also supports the Sockets applications in 4.2 BSD through a Sockets library interface.

Each subnetwork is treated as an 802.2 link layer provider (with an SNDCDF over the X.25 provider). CLNP is the internetworking section. The ISO Class 4 Transport protocol is the transport provider. Standard UNIX system applications, as well as RFS, uucp and other networking services access Class 4 directly through TLI. Upper layer applications such as the MHS electronic mail application are implemented at the UNIX system user level on top of TLI. This is because these protocols are still being refined in the standards bodies and are subject to frequent changes.

In the next few years there will be major enhancements to commercial OSI implementations. First, ISDN will be incorporated as an important class of network service. Second, the upper layer protocols will stabilise and migrate to the System V kernel. Figure 5 shows how these changes can be accommodated gracefully through the flexibility of STREAMS. ISDN can either be integrated as a subnetwork under CLNP or be used directly as a reliable and efficient provider of Network Layer service. The upper layer protocols will be implemented directly over TPI without breaking compatibility with existing applications or networking services.

TCP/IP

The implementation of the DARPA protocols in Figure 6 looks very much like the near term OSI implementation in Figure 4. TCP is the transport provider and IP the internetworking protocol. Note that the subnetworks remain precisely the same and it is possible to run OSI and TCP/IP over them simultaneously. TCP/IP has no upper layer architecture, so the networking services such as telnet or ftp are implemented directly to TLI. UNIX system applications and services such as uucp and RFS talk to TCP through TLI. In this configuration the sockets library for compatibility with BSD 4.2 is also supported.

SNA

UNIX System vendors must be able to communicate effectively with SNA. There are three things that a UNIX system machine should be able to do with an SNA network. First, it must be able to communicate with another UNIX machine across an SNA network and support standard UNIX system applications and networking services. In addition, the UNIX system must be able to access information that resides on the IBM architecture mainframes. To do this, it must be able to speak the higher-level SNA protocols such as LU 6.2. Finally, to enable mainframe applications to migrate to UNIX system, 3270 style synchronous terminal anywhere in the SNA network must be able to log on the UNIX system and access applications in the accustomed manner.

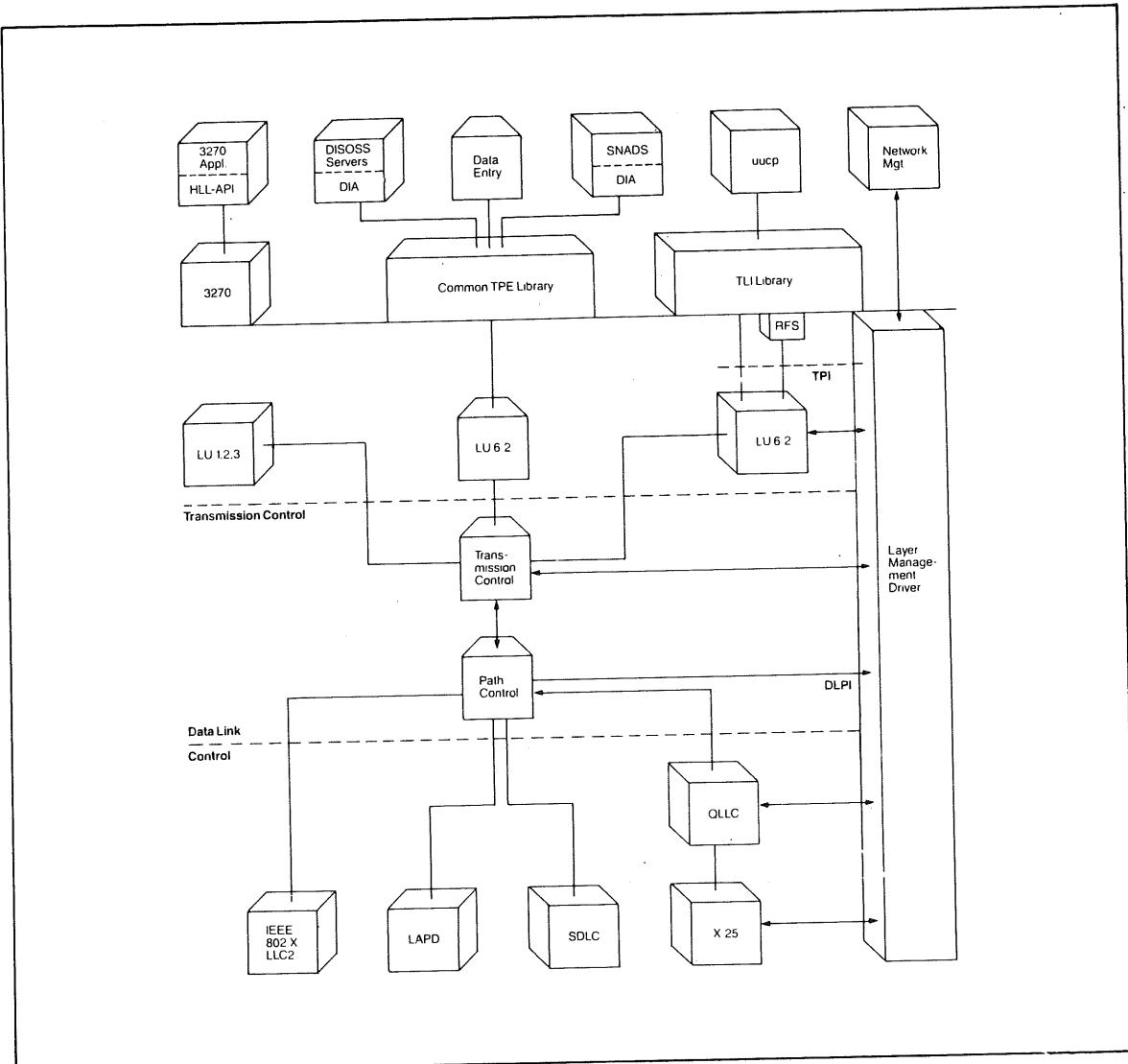
The long term solution to the problem is to implement SNA Host (PU Type 5) host services on the UNIX system. Figure 7 shows a potential implementation of PU Type 5 SNA node under

STREAMS. The heart of the architecture is a STREAMS-based implementation of the Path Control protocols. These protocols work either over SDLC lines or over X.25 wide area networks.

Each of the three modes of communication are implemented by accessing this STREAMS based Path Control transport provider. The System V applications and services access the LU6.2 basic conversation verb Transport Provider directly through TLI just as they did with TCP/IP. LU 6.2 is implemented as a STREAMS protocol module directly over Path Control. Applications and higher level services such as DIA can be implemented over the Common Transaction Processing interface library. The 3270 terminal traffic is handled by a protocol module that knows how to crack and format 3270 screen images. Applications can paint 3270 screens through a library interface similar to the current curses interface to asynchronous terminals.

References

- 1.D.M. Ritchie, A Stream Input-Output System, AT&T Bell Laboratories Technical Journal, Vol. 63, Issue 8, October 1984.
- 2.Olander, David J. McGrath, Gilbert J. & Israel, Robert K., I A Framework for Networking in System V, Proceedings from 1986 Summer Usenix Conference, Usenix Association, June 1986.
- 3.UNIX System V Release 3 STREAMS Programmer's Guide, AT&T , June 1986.
- 4.CCITT Recommendation X.200,I Reference Model of Open System Interconnection for CCIT Applications, 1984.
- 5.ISO IS 8072, Information Processing Systems - Open Systems Interconnection - Transport Service Definition, 1984.
- 6.System V Interface Definition, AT&T, 1986
- 7.Rifkin, A.P., RFS Architectural Overview, Proceedings of the 1986 Summer Usenix Conference, Usenix Association, June 1986.
- 8.ANSI/IEEE Standard 802.2, Logical Link Control, IEEE/Wiley-Interscience, 1984.

**Figure 7. SNA Long Term**

In the long term, the UNIX system must be able to communicate with IBM's System Networking Architecture (SNA). This will be accomplished by a STREAMS-based implementation of the Pat Control protocols. The UNIX System V applications and services access and LU.62 transport provider directly through TLL.

UNIX Security Workshop

Organised by the UK UNIX systems User Group

This one-day workshop was held in February 1989 at the Institute of Education, University of London. The proceedings are now available on a three hour VHS video. Neil Todd (GID Ltd) was in the chair and the speakers are:

—○—

Russell Brand (Lawrence Livermore National Labs)

HACKMAN: A Systematic Study of Real Computer Security Holes

—○—

Piete Brooks (Cambridge University)

Experiences at Cambridge

—○—

Lindsay Marshall (Newcastle University)

666: The Mark of the Beast

—○—

Chris Milsom (UNIX Europe Ltd)

System V/MLS

—○—

Jim Reid (Strathclyde University)

An Analysis of the Internet Worm

—○—

William Roberts (Queen Mary College)

The Sun Yellow Pages System

—○—

Andy Rutter (The Instruction Set)

Secure RPC

—○—

Mario Wolczko (Manchester University)

Some Myths and Facts about UNIX Security

—○—

The cost is £50.00, including post & packing. The price does not include VAT, which is payable if you order from the UK. Copies of the tape can be ordered from (you will be sent an invoice):

Birkbeck College Video Services

Malet Street

London WC1E 7HX

England

(+44) 1 631 6351



Window Systems

*William Roberts
liam@qmc-cs.uucp*

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

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

This article has been written by *Michael Gehret (mg@xsoft.uucp)* of *X/software, Grönenbach, West Germany*

X/ST/window

*Michael Gehret
(mg@xsoft.uucp)*

After studying computer science and economics and graduating Michael Gehret founded X/software. The 'X' and the slash stand for UNIX, of course: the company has specialised in the development and distribution of system software for UNIX computers and for computers connected to UNIX machines. Meanwhile, 'X' has a second meaning: "a window system called X, not a system called X-Windows". A major interest lies with the X Window System—from the X Window server through libraries and toolkits upto adaptation of software to X Window.

Introduction

UNIX and windows? From today's viewpoint, the name 'X' seems to be more programmatic than expected by its founders (Robert W. Scheifler and Jim Gettys state the descent from the 'V' and 'W' systems [1]). The X Window System is the graphics system, which comes closest to the original UNIX philosophy (modularisation, tools, distribution). However, one has to notice the questionable tendency to over-featuring and complexity in the X world, too.

Nach dem Abschluß des Studiums der Wirtschaftsmathematik gründete Michael Gehret die Firma X/software. Das 'X' und der Schrägstrich stehen—natürlich—for UNIX. Das Unternehmen ist spezialisiert auf Entwicklung und Vertrieb von Systemsoftware für UNIX-Rechner und für mit UNIX-Systemen vernetzte Computer. Inzwischen hat 'X' eine zweite Bedeutung: "a window system called X, not a system called X-Windows". Ein Tätigkeitszweck der X/software ist das X Window System—vom X Window Server über Büchereien und Toolkits bis zur Anpassung von Software an den neuen Standard.

Einführung

UNIX und Fenster? Der Name 'X' scheint mir aus heutiger Sicht programmatischer als von den Erfindern gedacht (Robert W. Scheifler und Jim Gettys beschreiben die Abstammung von den Systemen 'V' und 'W' [1]). Das X Window System ist das Grafiksystem, das der ursprünglichen UNIX-Philosophie (Modularisierung, Tools, Verteilung) am nächsten kommt. Allerdings ist auch im X-Umfeld die sehr bedenkliche Tendenz zum Feature-ismus und zur Unübersichtlichkeit zu bemerken.

William Roberts' first article of this column [2] compared the X Window System to other graphics systems and mentioned the advantages: manufacturer independence, quality, network transparency, flexibility and the developer's great freedom—but also responsibility—in making design decisions.

First, I'll summarise the design principles of the X Window System published in [1]. Then I'll introduce X/ST/window, our port of the X Window server to the ATARI ST, a powerful microcomputer. Finally, I'll outline the design of the system and the development environment.

The X Window System...

The X Window System is a quasi-standard for graphics on UNIX, DEC and other computer systems. It was developed at the Massachusetts Institute of Technology in cooperation with renowned computer companies. The nine conceptual requirements of the X Window system distinguish it from other graphics systems:

The system must be able to be implemented on various graphics hardware.

Presently, the machines that are suitable for the X Window server range from simple monochrome raster monitors to colour subsystems with graphic processors and multiple monitors.

The system must allow device independent applications programming.

Applications should not have to be modified to suit the graphics hardware. Another aspect is that every graphics function defined by the system should work on virtually every supported display without questioning the capabilities of the X Window server.

The system must be network transparent.

The X Window server, responsible for user input and graphics output, and the application itself need not run together on one machine.

William Roberts hat im ersten Artikel dieser Kolumne [2] das X Window System mit anderen aktuellen Grafiksystemen verglichen und dabei die Vorteile genannt: Herstellerunabhängigkeit, hohe Qualität, Netzwerktransparenz, Flexibilität und große Entscheidungsfreiheit—aber auch Verantwortung—for den Software-Entwickler.

Ich fasse hier zunächst die Design-Konzepte des X Window System aus [1] kurz zusammen. Anschließend stelle ich X/ST/window, vor, unsere Portierung des X Window Servers für den ATARI ST, einen leistungsfähigen Microcomputer. Schließlich beschreibe ich kurz den Entwurf des Systems sowie die Entwicklungsumgebung.

Das X Window System...

Das X Window System ist ein Quasi-Standard für Grafik auf Rechnern unter dem UNIX-Betriebssystem, DEC-Computer und andere Maschinen. Es wurde am Massachusetts Institute of Technology unter Beteiligung namhafter Rechnerhersteller entwickelt. Die konsequente und erfolgreiche Realisierung von neun Anforderungen an das Konzept unterscheidet das X Window System von anderen Grafiksystemen:

Das System soll auf unterschiedlicher Grafik-Hardware implementiert werden können.

Heute reichen die Geräte, für die X Window Server angeboten werden, von einfachen monochromen Rasterbildschirmen bis zu farbigen Subsystemen mit Grafik-Prozessoren und mehreren Bildschirmen.

Das System soll die geräteunabhängige Programmierung von Anwendungen ermöglichen. Einerseits sollen Anwendungen unabhängig von der speziellen Grafik-Hardware sein. Andererseits sollen alle Grafikfunktionen des Systems von jeder Grafik-Hardware unterstützt werden, und zwar ohne besondere Eigenschaften des X Window Servers erfragen zu müssen.

Das System soll netzwerktransparent sein.

Der X Window Server, der für Benutzereingaben und Grafikausgaben zuständig ist, und die Anwendung müssen nicht notwendigerweise auf ein und derselben Maschine ausgeführt werden.

This means you can control a number crunching simulation program on a distant super computer from your X Window terminal and be able to watch the graphics results simultaneously. The applications and the X Window server communicate mainly over standardised network protocols—regardless of machine architecture or operating system.

The system must support multiple applications running simultaneously.

Due to network transparency, applications running on several computers within the network can display output on a single X Window terminal all at the same time.

The system must be capable of supporting every kind of user interface.

The X Window System doesn't specify one kind of user interface, it offers an implementation basis for various interfaces, depending on the type of application or the kind of user group.

The system must support overlapping windows.

Windows separate the different kinds of output from one or several applications, but also steer—in coordination with the mouse pointer—input to the chosen application. Overlapping windows use the monitor surface to its full advantage. The X Window System allows output to partially or totally obscured windows, which in turn allows applications to be absolutely independent of the present position and visibility of its windows.

Every application must be able to use a hierarchy of resizable windows.

If an application needs to subdivide windows, it can make use of the X Window System's window abstraction for its own sub-windows. Every graphic object can own a window associating arbitrary many, including application specific, attributes.

So können Sie von Ihrem X Window Terminal aus ein rechenintensives Simulationsprogramm auf einem entfernten Supercomputer steuern und dessen Grafikausgabe betrachten. Die Anwendungen kommunizieren mit dem X Window Server meist über standardisierte Netzwerkprotokolle — unabhängig von Maschinenarchitekturen und Betriebssystemen.

Mehrere Anwendungen sollen gleichzeitig bedient werden können.

Aus der Netzwerktransparenz folgt, daß mehrere, auf verschiedene Rechner eines Netzwerkes verteilte Anwendungen simultan auf ein X Window Terminal ausgeben können.

Das System soll offen sein für jede Form von Benutzerschnittstelle.

Das X Window System legt nicht eine bestimmte Art von Benutzerschnittstelle fest, sondern bietet die Grundlage zur Implementierung der unterschiedlichsten Schnittstellen, je nach Anwendungstyp oder Benutzergruppe.

Das System muß sich gegenseitig überlappende Fenster unterstützen.

Fenster trennen einerseits die verschiedenen Arten von Ausgabe einer oder mehrerer Anwendungen und lenken andererseits—in Kombination mit dem Mauszeiger—die Eingabe zur Anwendung. Sich überlappende Fenster nutzen die Bildschirmfläche optimal aus. Das X Window System ermöglicht die Ausgabe in teilweise oder total verdeckte Fenster, so daß ein Anwendungsprogramm vollkommen unabhängig von der momentanen Lage und Sichtbarkeit seiner Fenster ist.

Jede Anwendung soll eine Hierarchie von in ihrer Form veränderbaren Fenstern verwalten können. Benötigt eine Anwendung eine weitere Unterteilung ihrer Fenster, kann sie die Fensterabstraktion des X Window Systems für eigene (Sub-)Fenster nutzen. Jedem grafischen Objekt kann ein Fenster mit beliebigen, auch anwendungsspezifischen, Attributen zugeordnet werden.

The system must provide graphic functionality on a low abstraction level.

The X Window System doesn't determine the kind of graphic interface used by the application. The simple functionality of the X Window server's graphics core system (2-D raster graphic, text) allows a higher abstraction level on the application side, such as object orientated interfaces (e.g., Athena Widgets, Andrew Toolkit, Xray) or graphic terminal emulation (e.g., Tektronix).

The system should be extensible.

An interface for the extension of the X Window server itself (e.g., PEX PHIGS 3-D graphics) is defined.

...for the

Presently, X/ST/window is the only port of the X Window server (version 11, release 3) on the ATARI ST.

Many ATARI STs are used as terminals on UNIX and DEC computers. The non-flickering screen, the DEC compatible keyboard, the possibility of running local applications and naturally the "Power Without the Price" are the advantages for this solution. However, the use of the ATARI ST is generally restricted to simple ASCII-terminal emulation and file transfer.

The ATARI ST's hardware is well suited for the implementation of the X Window server:

- The high performance MC68000 processor addresses a linear, unsegmented memory area of up to 4 Megabytes.
- The screen memory is accessed via a second memory bus, and its address is selectable.
- The graphic coprocessor Blitter relieves the main processor.
- Another processor deals with mouse and keyboard input.
- Extra hardware (e.g., a bigger monitor) can be attached via the processor bus or the DMA port.

Das System soll Grafikfunktionalität auf einem niedrigen Abstraktionsniveau bieten.

Das X Window System legt die Art der Grafikschnittstelle einer Anwendung nicht fest. Auf der einfachen Funktionalität des Grafikkerns des X Window Servers (2-D-Rastergrafik, Text) können auf der Anwendungsseite höhere Abstraktionsebenen wie objektorientierte Schnittstellen (z.B. Athena Widgets, Andrew Toolkit, Xray) oder Grafikterminal-Emulationen (z.B. Tektronix) implementiert werden.

Das System soll erweiterbar sein.

Für die Erweiterung des X Window Servers selbst (z.B. PEX PHIGS 3-D-Grafik) ist eine Schnittstelle definiert.

...für den

X/ST/window ist z.Z. die einzige Portierung des X Window Servers (Version 11, Release 3) auf den ATARI ST.

Sehr viele ATARI ST werden als Terminals an UNIX- oder DEC-Rechnern eingesetzt. Der vollkommen flimmerfreie Bildschirm, die DEC-kompatible Tastatur, die Möglichkeit, lokale Anwendungen zu fahren und natürlich der günstige Preis sprechen für diese Lösung. Allerdings bleibt die Verwendung des ATARI ST in der Regel auf einfache, zeichenorientierte Terminalemulation und Dateitransfer beschränkt.

Die Hardware des ATARI ST eignet sich sehr gut zur Implementierung eines X Window Servers:

- Der leistungsstarke Prozessor MC68000 greift auf einen linearen, unsegmentierten Adressraum von bis zu 4 MByte zu.
- Der beliebig verschiebbare Bildschirmspeicher wird über einen zweiten Speicherbus adressiert.
- Der Grafik-Coprozessor Blitter entlastet den Hauptprozessor.
- Ein weiterer Prozessor verarbeitet Maus- und Tastatureingabe.
- Am Prozessorbus oder an der DMA-Schnittstelle kann zusätzliche Hardware, z.B. ein größerer Bildschirm, angeschlossen werden.

Operating system, ...

The X Window server has very high demands on the computer's system software, which is generally UNIX. The ATARI ST's operating system, TOS, does not have the necessary features for a direct implementation of the server. This is the reason why X/ST/window is based on X/ST/multi, a multi-tasking operating system that is totally compatible to TOS. The following concepts are used especially by X/ST/window:

- The process manager administers a number of processes with different priorities to be run quasi-parallel. Besides the X Window server process there are at least one protocol and two daemon processes running on the ATARI ST (see below).
- Processes are synchronised with semaphores.
- The inter process message system uses semaphores without polling and is functionally compatible to UNIX System V. Message queues are used by X/ST/window to enable the server to run independent of the time critical protocol process.
- The very efficient dynamic memory management supports the message system: Processes exchange the addresses of data areas, instead of the data areas themselves (*shared memory* and *memory transfer*). Data flow appears as a circulation of data areas from the network to the server and vice versa.

...graphics...

The partly newly developed graphics core system of X/ST/window supports monochrome raster graphics. Only a section of a virtual screen is to be seen, the dimensions can be chosen freely, only limited by the available main memory. The section is moved automatically the instant the mouse crosses the screen's borders.

Some of the graphic functions make use of the graphics coprocessor (Blitter), but don't necessarily need it.

Betriebssystem, ...

Der X Window Server stellt sehr hohe Anforderungen an die Systemsoftware des zugrundeliegenden Rechners—in der Regel handelt es sich um UNIX. Das Betriebssystem des ATARI ST, TOS, eignet sich nicht zur direkten Implementierung des Servers. X/ST/window basiert daher auf X/ST/multi, einem Multitasking-Betriebssystem, das zu TOS vollkommen kompatibel ist. Die folgenden Konzepte werden insbesondere von X/ST/window genutzt:

- Der Prozeßmanager verwaltet beliebig viele Prozesse, die quasi parallel mit verschiedenen Prioritäten ausgeführt werden. Auf dem ATARI ST laufen neben dem eigentlichen X Window Server mindestens ein Protokoll- und zwei Dämon-Prozesse (s.u.).
- Prozesse können über Semaphore synchronisiert werden.
- Das der Interprozeßkommunikation dienende Message-System ist funktional zu UNIX System V kompatibel und mit Semaphoren ohne Polling realisiert. X/ST/window verwendet *message queues* zur Entkoppelung des Servers vom zeitkritischen Protokollprozeß.
- Eine sehr effiziente dynamische Speicherverwaltung unterstützt das Message-System: Prozesse tauschen nicht Datenflächen, sondern Adressen von Datenflächen aus (*shared memory* und *memory transfer*). So stellt sich der Datenfluß in X/ST/window als Kreislauf von Datenflächen vom Netzwerk zum Server und umgekehrt dar.

...Grafik...

Der zum Teil neu entwickelte Grafikkern von X/ST/window realisiert monochrome Rastergrafik. Auf dem Monitor ist nur ein Ausschnitt eines virtuellen Bildschirms zu sehen, dessen Abmessungen frei wählbar und lediglich durch den Arbeitsspeicher begrenzt sind. Dieses Fenster wird automatisch verschoben, sobald der Mauszeiger eine Begrenzung des Ausschnitts überschreitet.

Einige der Grafikfunktionen nutzen den Grafik-Coprozessor Blitter. Er wird aber nicht vorausgesetzt.

X/ST/window can load a keyboard definition from the local mass storage or from a UNIX-computer within the network to meet national or special requirements. You can activate two keyboard layouts at the same time.

...and network

X/ST/window can communicate with one or more X applications over a network connection, this being mainly TCP/IP on the UNIX side. With the choice of daemon and protocol processes, the following options are possible:

- One or more ATARI STs are connected in a star-shaped manner to a UNIX computer over serial terminal lines (19200 Baud). A very safe and efficient protocol was implemented for this solution. The connection to TCP/IP is through a daemon process running on the UNIX computer, which simulates one or more X Window servers for applications throughout the network.
- The ATARI STs are linked to the UNIX machine by Ethernet interfaces. The Ethernet hardware (without cpip) takes the part of the serial lines mentioned above.
- The ATARI ST is used as a node within an Ethernet network under TCP/IP. This solution does not require additional software on the UNIX side.

The choice is made at the start of the X Window server, which is totally independent of the kind of network connection available. This way **X/ST/window** need not be modified when upgrading to new, more expensive and higher-performance network technology.

Font files can be managed either locally on the ATARI ST or centralised or distributed over the computers in the network, in ASCII, binary or compress format.

X/ST/window kann eine Tastaturdefinition vom lokalen Massenspeicher oder von einem Rechner des Netzwerks laden, um einem nationalen oder einem speziellen Tastatur-Layout zu entsprechen. Es ist auch möglich, gleichzeitig zwei Tastaturbelegungen zu aktivieren.

...und Netzwerk

X/ST/window kommuniziert über eine Netzwerkverbindung mit einer oder mehreren X Window Anwendungen. Auf der UNIX-Seite handelt es sich dabei meist um TCP/IP. Durch die Wahl verschiedener Dämon- und Protokollprozesse sind folgende Optionen möglich:

- Ein oder mehrere ATARI ST sind über serielle Terminalleitungen (19200 Baud) sternförmig mit einem UNIX-Rechner verbunden. Hierzu wurde ein sehr sicheres und effizientes Protokoll implementiert. Der Anschluß an TCP/IP erfolgt über einen Dämon-Prozeß, der auf dem UNIX-Rechner für im Netzwerk verteilte X Window Anwendungen einen oder mehrere X Window Server simuliert.
- Die ATARI ST werden über Ethernet-Schnittstellen mit einem UNIX-Rechner verbunden. Die Aufgabe der o. a. seriellen Leitung übernimmt hier die Ethernet-Hardware (ohne TCP/IP).
- Der ATARI ST wird als Knoten in einem Ethernet-Netzwerk mit TCP/IP betrieben. Diese Lösung benötigt auf der UNIX-Seite keine zusätzliche Software.

Die Auswahl erfolgt beim Start des X Window Servers, der vollkommen unabhängig von der Art der Netzwerkverbindung ist. So ist die einfache Migration von kostengünstigen zu leistungsfähigen sowie die Integration neuer Netzwerktechnologien ohne Änderung von **X/ST/window** möglich.

Zeichensatzdateien können auf dem ATARI ST lokal sowie zentral oder verteilt auf Rechnern des Netzwerks verwaltet werden, in ASCII-, binärem oder compress-Format.

A fileserver daemon runs in background to the X Window server. It provides UNIX-lookalike commands like `cp`, `mv`, `ls`, `pwd`, `cd`, `mkdir`, and `rmdir`. These commands affect the local filesystem of the ATARI ST and the remote virtual filesystems of connected hosts. For example, the command `cp unix_file c:\tos_file &` copies a file from the UNIX host to the ATARI ST—parallel to the graphics output of the X Window server. The command `print` prints an arbitrary file on the local printer.

The situation presently...

After being presented at the Hanover CeBIT fair in March, **X/ST/window** is now available. The software needed for connections to TCP/IP has been ported to several UNIX computers. By the time you read this text, TCP/IP for the ATARI ST should be available. The software requires 2 MByte of main memory (i.e., ATARI ST Mega 2 or Mega 4), but no hard disk.

...and the future

X/ST/window will always be up to date with the latest X Window standard. In keeping with continual improvement, we will attempt to integrate suggestions coming from the user side. Thus, the server will run with big screens providing higher resolutions.

Design...

As mentioned above, the ATARI ST's operating system, TOS, is not comparable to UNIX. Therefore, there are two ways to bring the X Window server to the ATARI ST: reducing the server's functionality, or implementing UNIX-lookalike capabilities on the ATARI ST. Of course, our server should be fully compatible to the X Window Protocol definition [3]. So we decided to go the second, harder, way. The multitasking system **X/ST/multi** allows to port UNIX-based software to the ATARI ST with only few modifications, and even to combine it with TOS software.

Im Hintergrund zum X Window Server läuft ein Fileserver-Dämon, der es erlaubt, UNIX-ähnliche Kommandos wie `cp`, `mv`, `ls`, `pwd`, `cd`, `mkdir`, `rmdir` abzusetzen. Diese Befehle wirken auf das lokale Dateisystem des ATARI STs wie—virtuell—auf die Dateisysteme der über das Netzwerk angeschlossenen Hostrechner. So kann z. B. mit `cp unix_file c:\tos_file &` eine Datei von einem UNIX-Rechner zum ATARI ST kopiert werden—in Hintergrund zur Grafikausgabe des X Window Servers. Mit `print` können beliebige Dateien lokal ausgedruckt werden.

Der aktuelle Stand...

Nach der Vorstellung auf der CeBIT in Hannover ist **X/ST/window** jetzt erhältlich. Die Software zum Anschluß an TCP/IP wurde auf verschiedene UNIX-Rechner portiert. Zum Zeitpunkt der Veröffentlichung dieses Textes ist TCP/IP für den ATARI ST wahrscheinlich verfügbar. Die Software setzt z.Z. noch einen Arbeitsspeicher von 2 MByte voraus (ATARI ST Mega 2 oder Mega 4), eine Festplatte ist nicht erforderlich.

...und die Zukunft

X/ST/window wird immer dem neuesten Standard von X Window entsprechen. Parallel zu dieser laufenden Aktualisierung werden wir Anregungen aus dem Benutzerkreis verwirklichen, so z.B. die Unterstützung großer Monitore mit höherer Auflösung.

Entwurf...

Wie bereits bemerkt, ist das Betriebssystem des ATARI STs, TOS, nicht mit UNIX zu vergleichen. So gibt es grundsätzlich zwei Wege, den X Window Server auf dem ATARI ST zu implementieren: entweder die Einschränkung der Funktionalität des Servers oder die Schaffung UNIX-ähnlicher Voraussetzungen auf dem ATARI ST. Selbstverständlich soll unser Server der X-Protokoll-Definition [3] entsprechen, wir haben uns also für die zweite, schwierigere Möglichkeit entschieden. Das Multitasking-System **X/ST/multi** erlaubt es uns, UNIX-basierte Software mit relativ geringen Änderungen auf den ATARI ST zu portieren und mit TOS-Software zu kombinieren.

Of course, the server's graphics parts needed extensive enhancements and modifications. I mention the important ones only:

- keyboard driver with autorepeat and loadable keymaps;
- mouse driver simulating the third, middle button;
- simulation of a hardware cursor in software, for greater efficiency on a very low level;
- implementation of the virtual screen manager;
- integration of the graphics coprocessor;
- optimisations and adaptations to the architecture of the MC68000.

The networking part —protocols for serial line and raw Ethernet, daemon programs, and the fileserver—is totally new. This software is supported by X/ST/multi's message system.

...and development

Soon it became clear, that even the 4-MByte ATARI ST Mega 4 is not the appropriate development platform for the port of the X Window server: the MIT distribution of the software is based on BSD UNIX, and we wanted to use the extensive development environment of UNIX.

Now the software is maintained on a Hewlett-Packard 9000/330 workstation. This computer uses the MC68020 processor, contrary to the ATARI ST which is built around the first processor of this family, the MC68000.

Fortunately, HP-UX's C development environment contains compiler and assembler for the MC68000 and MC68010 processors. The generated code emulates the floating point coprocessor MC68881. We created driver programs for the special compiler and linker, and a program, which translates executable program files from the UNIX workstation's a.out format into a format loadable by the ATARI ST. The symbol table is translated, too, preserving the significance of long symbol names.

Die Grafik-Software des Servers erforderte natürlich umfangreichere Ergänzungen und Modifikationen. Hier seien nur die wichtigsten genannt:

- Treiber für die Tastatur mit Wiederholfunktion und ladbarer Tastaturbelegung;
- Treiber für die Maus mit Simulation des dritten, mittleren Mausknopfes;
- Simulation eines Hardware-Cursors in Software, zur Effizienzsteigerung auf einer sehr maschinennahen Ebene;
- Implementierung des virtuellen Bildschirm-Managers;
- Ansteuerung des Grafik-Coprozessors;
- Optimierungen und Anpassungen an die Architektur des MC68000.

Die Netzwerk-Seite —Protokolle für serielle Terminalleitung und rohes Ethernet, Dämonprogramme und der Fileserver—wurden auf der Basis des Message-System von X/ST/multi neu entwickelt.

...und Entwicklung

Es war recht bald klar, daß selbst ein ATARI ST Mega 4 mit 4 MByte Arbeitsspeicher nicht die geeignete Maschine für die Portierung des X Window Servers darstellt. Zum einen basiert die Software des MIT auf BSD UNIX und zum anderen sollten die umfangreichen UNIX-Tools bei der Entwicklung Verwendung finden.

Die Software wird z.Z. auf einer Hewlett-Packard 9000/330 Workstation weiterentwickelt. Dieser Rechner arbeitet mit einem MC68020 Prozessor, während der ATARI ST mit dem ersten Prozessor dieser Familie, dem MC68000, ausgerüstet ist.

Die C-Entwicklungsumgebung unter HP-UX beinhaltet jedoch einen Compiler und einen Assembler für die MC68000 und MC68010 Prozessoren. Der Floating-Point-Coprozessor MC68881 wird dabei emuliert. Wir entwickelten Treiberprogramme für den speziellen Compiler und den Linker, sowie ein Programm, das ausführbare Programmdateien vom a.out-Format der UNIX-Workstation in ein Format umwandelt, das vom ATARI ST geladen werden kann. Dabei wird auch die Symbolebene gewandelt, und zwar unter Beibehaltung der Signifikanz langer Symbolnamen.

Thus we can let a symbolic debugger drive X/ST/window on the ATARI ST.

The system calls of the C library are transformed to correspondent TOS-system calls by the multitasking system X/ST/multi.

HP-UX, Hewlett-Packard's variant of UNIX is a "System V with BSD extensions". Nevertheless, the compilation of the core software and of the less portable contributions was rather straightforward. For some time we ran Hewlett-Packard's X10-server and X10-clients beside ported X11-clients controlled by a X11-X/ST/window simultaneously.

Conclusions

Once more, the ATARI ST turned out to be an underestimated powerful computer. Using extensive development tools one can implement sophisticated systems on top of suitable system software.

X/ST/window offers inexpensive graphics functionality and may substitute a workstation in a network. Of course, its performance is less than that of a high-end workstation. But often the demands on a graphics terminal are lower.

X/ST/window can be used as a UNIX terminal for text, too: the possibility to control multiple shells—even on different machines—at the same time from one terminal is very valuable.

So können wir X/ST/window auf dem ATARI ST unter einem symbolischen Debugger betreiben.

Die Systemaufrufe der C-Bücherei werden unter dem Multitasking-System X/ST/multi auf die entsprechenden TOS-Systemaufrufe abgebildet.

HP-UX, die UNIX-Variante von Hewlett-Packard, ist eher an System V als an BSD orientiert. Trotzdem gab es bei der [bersetzung der *core*-Software wie auch der weniger portablen *contributions* wenig Probleme. Zeitweilig betrieben wir gleichzeitig einerseits X10-Server und -Clients von Hewlett-Packard und andererseits von uns portierte X11-Clients in Verbindung mit einem X11-X/ST/window.

Zusammenfassung

Der ATARI ST erwies sich wieder als allgemein unterschätzter, leistungsfähiger Computer. Auf entsprechender Systemsoftware und mit leistungsfähigen Entwicklungswerkzeugen lassen sich anspruchsvolle Systeme realisieren.

X/ST/window bietet einen preisgünstigen Grafikarbeitsplatz und kann so eine Workstation in einem Netzwerk ersetzen. Selbstverständlich ist die Performance geringer als die einer Hochleistungs-Workstation. Oft werden jedoch nur geringere Anforderungen an einen Grafikarbeitsplatz gestellt.

X/ST/window kann natürlich auch als UNIX-Terminal für Textanwendungen verwendet werden: die Möglichkeit, gleichzeitig mehrere Shells—sogar auf verschiedenen Rechnern—from einem Terminal aus zu kontrollieren, ist sehr wertvoll.

Bibliography

— Literatur

- [1] Robert W. Scheifler, Jim Gettys: The X Window System, in: ACM Transactions on Graphics, Vol. 5, No. 2, April 1986
- [2] William Roberts: Introduction to Window Systems, in: EUUG Newsletter, Vol. 9, No. 1, Spring 1989
- [3] Robert W. Scheifler: X Window System Protocol, MIT

C—The Effective Solution, An Intensive Video Course

Mick Farmer and Richard Murphey, Chartwell-Bratt, 1988, 2 Volumes plus 2 tapes (VHS—other formats also available), ISBN 0-86238-190-8 and 0-907904-13-0 (UK) Price £850, Soft Back, 186 pp, size: (text book) A5, (work book) A4.

Reviewed by Jane Spreull of the Agricultural and Food Research Council Computing Centre.

This is a video based self-paced course in 'C'. It assumes only basic programming knowledge in other languages, and covers all the standard 'C' commands and most of the more common library functions.

The package contains a video tape, a workbook (*C: The Effective Solution Workbook*), and a textbook (*The Intensive 'C' Course*). The video tape consists of 10 units, each lasting about 30 minutes.

The Textbook is well structured with many example code fragments illustrating the use of the language in each section, giving the reader the 'flavour' of the 'C' language. The language presented is a standard version with none of the usual extensions which abound particularly in the PC 'C', and this I would consider to be a great advantage (portability, availability of compilers, and the like).

The video presentation was good with changes in the viewing angles, and separate full screen displays for 'C' text adding interest (assuming there is such a thing as interest in a programming

language). The use of the screen layout for the demonstrations was clear but at times not very obvious. The workbook clarified the points but a videotape player with a remote controller with a 'Stop' button is essential to give time for the script to be consulted. It is clear that many viewings of the tape mixed with some practical work would be needed to get the most out of the tape—you are, after all, trying to learn a new language.

The tone of voice used by Mick Farmer was modulated in a pleasing manner giving the impression of a well practiced and clear presentation. This leads to a feeling that the information given is complete and without errors. The presentation was very good compared with other video courses where the information was presented in such a monotone that it became irritating after a very short period of time.

The Workbook is *not* a simple transcript of the video but combines with it to give a package that provides a very good basis on which to build real 'C' programming expertise.

The course is available now from:

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

it will soon be available from your favourite bookshop.

UNIX-Communications

Bart Anderson, Bryan Costales, and Harry Henderson, The Waite Group (A Division of Macmillan), 1987, Price \$29.50, 542 pp.

Reviewed by Anke Goos of University of Dortmund, IRB <anke@unido.uucp>.

Introduction

"This book is bad", said an experienced News-Postmaster, weighing about one kilogram of paper in his hand, "our postmaster-team could have written this as well". As I had the experience of waiting a long time for one article about the News systems from this very person, my estimation is:

"Some knowledge in my hand is better than inside others' heads". Especially as the head of this person has already left us for "serious professional work". There is a lot of information in those 540 pages of 'UNIX Communications' which has been compiled by three authors of the Waite Group.

In the context of this book UNIX Communications means three major chapters about 'Mail', the news conferencing systems of 'Usenet' and 'UUCP'. There are more than 200 pages dedicated only to the last topic. The book is not written for people who like digging in UNIX

manuals. Instead it's a book for newcomers and interested people, beginning at point zero. BTW: zero is an introduction to UNIX as an operating system in 20 pages.

Mailing First...

At this stage the innocent UNIX-fan is initiated in the ABC of mailing step by step through well-structured units and a lot of illustrations. This how-to-mail is presented by the author Bert Anderson, following the example of mailx as the mailing software for System V machines and mail for Berkeley 4.2.

But to get to those 'elitist' :-) functions like including a file or a message you have to survive thirty pages to reach a chapter on 'Advanced Mailer'. I would recommend that you do not follow each chapter and paragraph, but pick out all the functions that you may need for a happy mailing life. This might already be the neat explanation for manipulation of your *.mailrc* file for a CC of an automatic copy of your mail to people other than the recipient.

What's missing in this first part about mailing? Well, for example, a comparison to other mail user agents like the screen-oriented PD program Elm. Or other advanced systems like MH, recommandable if you really have to handle a lot of e-mail. Furthermore, there's no preparation on the 'other' maybe shocking form of communication in daily e-mail life like:

>The indentation of those sentences you are referring to..
Yes, that's what I'm thinking of...
>The neat smileys :-), turn your head at the left side by 90 degrees, thank you. You got it? No ?-) Too bad :-(.

The author shows a reluctance to deal with the complex topic of addressing in UNIX networks, or even through to other networks—just as if all that exists is the old-style bang-addressing of UUCP (*host1!host2!!user*) throughout networld—the book was written in 1987. The concentration on the more chaotic American UUCP network is also shown by a whole chapter on how to construct a path of hosts to reach your end-user. Sometimes the Europeans are a bit advanced :-).

Usenet for the Advanced Mailer

You might have imagined that network communication is explained in the 'Usenet'

chapter by Harry Henderson. It is, but it is focussed on the computer conferencing system of the 'readnews' and the topological network of the Usenet backbone, secondary feeders and end-user hosts. A lot of standard information out of the newsgroups is summarised. Extended paragraphs inform you about modern user interfaces for reading the 'news', like *rn* or the screen-oriented *vnews*. You might even find interesting details, like changing the the newsgroup list of a posting or the deleting of articles which have already been posted. I estimate some details are useful even for an advanced news reader.

Last but not least: UUCP

This comprehensive part covering UUCP migrates away from the needs of an end-user in the direction of the programmer, system administrator or curious UNIX-wizard. Although this information is already well prepared and explained, there are more hard facts about the UUCP-Program in general and in detail.

Examples are the copying of files by UUCP and USEND, UUID and UUPICK or remote execution on other hosts by ux, Status Reports and Job Control or CU for login on other machines. Or the copying and packing of files, directories and binaries with shar, tar or cpio (Copying Archives and Binary Files).

In the very end this book becomes a typical American 'How-to-do-Book'. In the way of 'We are sending a mail' the cooperation of all UUCP tools is explained step by step. This is not too bad as an overview. But we are not yet finished.

Some critics remain: Of course it would have been better to have two or three different handbooks about the three different topics. At least to decrease the personal barrier. Although you might fall asleep, the book is not really suitable for bed time reading. Sometimes it goes into too much detail. Other criticisms? Yes. There is no word about EUnet in the whole book. Neither in the index nor in the description of the backbone topology. Europe is simply considered to be a part of Usenet and referenced by some obsolete backbone names. But you can't have everything. Unless you want to write a book for yourself. Any volunteers? EUnet is still searching for good documentation to be included in a starter-kit for new members...

Calendar of UNIX Events

This is a combined calendar of planned conferences, workshops, or standards meetings related to the UNIX operating system.

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

Abbreviations:

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

year mon days	conference	(sponsor,) (hotel,) location
1989 May 1-2	USENIX Transactions	W Pittsburgh, USA
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 Symposium	Tokyo, Japan
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 7-8	Sun UK UG	C Manchester, UK
1989 Sep 7-9	USENIX Sys Admin	W Austin, TX, USA
1989 Sep 18-22	EUUG	WU Vien, Vienna, Austria
1989 Sep 19-22	ACM SIGCOMM	Austin, TX
1989 Sep 25-29	GUUG	CT Wiesbaden, Germany
1989 Oct 16-20	IEEE 1003	Brussels, Belgium
1989 Oct	UNIX Expo	New York, NY
1989 Oct 5-6	USENIX -Distrib Proc	W Florida, USA
1989 Oct 31-Nov	2 IETF	IAB, U. Hawaii, Honolulu, HI
1989 Nov 1-3	UNIX EXPO	Javits Conv. C, New York, NY
1989 Nov 6-10	DECUS S	Anaheim, California
1989 Nov 9	NLUUG C	The Netherlands
1989 Nov 9-10	14th JUS UNIX Symposium	Osaka, Japan
1989 Nov 16-17	USENIX Graphics	W California, USA
1989 Nov 24	AFUU C	Paris, France
1989 Dec 5-6	JUS UNIX Fair 89	Tokyo, Japan
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

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

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

Everyone who comes here wants three things:

1. They want it quick.
2. They want it good.
3. They want it cheap.

I tell 'em to pick two and call me back.

Sign on the back wall of
 a small printing company
 in Delaware, USA

UKUUG Summer Abstracts

Here are the abstracts of the papers that will be delivered at the UKUUG meeting at Strathclyde on 27th-28th June 1989.

Interconnection in a Graphical Environment
Robin Faichney - University of Kent, Canterbury

One of the most significant features of UNIX is the ability to build applications from small, modular, reusable units. This means ease of inter-connection, achieved largely through command line redirection. The redirection of user-interaction is complicated in a graphical environment due to the nature and variety of graphical actions, and also the highly interactive nature of many graphics-interfaced applications. Two systems have been designed and implemented to facilitate interconnection. (While one is complete, the other exists as a partial prototype.)

The first is, in effect, a high level interface to existing inter-process communication facilities, incorporating additional features. The other consists of a library of routines designed to allow the redirection of all traffic between the user-interface and the underlying functionality of an application, to and/or from another program. Apart from the switching on/off of redirection, it is fully transparent to the application. Applications which demonstrate the use of both systems have been implemented.

An Introduction to GoodNeWS
Arthur van Hoff - Turing Institute, Glasgow

GoodNeWS is a NeWS-based window interface that was developed at the Turing Institute. It provides a complete windowing environment which includes terminal emulators, graphics tools and a LaTeX previewer. Most of the Sun-supplied PostScript software has been replaced to provide a more consistent interface. GoodNeWS also incorporates an object-oriented programming environment in PostScript called HyperNeWS, which presents a HyperCard-style user interface.

An Interactive UNIX spelling corrector
Philip M Sleat, Sunil K Das - City University, London

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

An Overview of the Rekursiv
Bruno Beloff - Linn Smart Computing Ltd., Glasgow

The Rekursiv is a novel computer architecture developed by Linn to provide a processor that efficiently executes object-oriented programming languages. This paper describes that architecture of the Rekursiv and discusses some of the problems of using UNIX as a "slave" operating system to the Rekursiv processor.

NFS refreshes the filesystem(s) A/UX cannot reach
William Roberts,
Matthew Kempthorne-Ley-Edwards,
Robert Bradshaw
Queen Mary College, London

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

The Independent—It Is. Are UNIX?
Dwight A. Ernest, Maria Tuck
Newspaper Publishing PLC, London

The Independent is a major national quality newspaper in the UK. Since shortly after its launch in October 1986, its I.T. team has been using Sun workstations running a variant of the UNIX operating system to perform a wide and varied set of tasks. The importance and prominence of this operating system is increasing at the newspaper because of its 'developer friendliness', its standard networking capabilities, and the increasing price/performance ratio of various computers and workstations which are based on it (including Suns).

The presentation and accompanying paper will explain the environments in the development and operations area, and will describe how the various elements are networked. The use of the various other non-UNIX computers at the newspaper will also be very briefly explained so as to demonstrate how it all fits together and how UNIX networking provides indispensable links between the various disparate elements.

UKnet Update
Steve Binns - University of Kent, Canterbury

The head of the UKC Software Division, will give a short explanation of what UKnet is, how to join, the costs and other general information. He will also explain some of the future connectivity services both in the UK, Europe and the wider world for UKnet sites.

Installing Software on UNIX
Tony Banford - Parliament Hill Computers, Caversham

How is a 'shrink-wrapped UNIX product' installed on any random UNIX engine? Traditional methods are discussed, current standards work is looked at and possible 'great leap forwards' talked about. The presentation will follow up the issues raised from the recent London UNIX User Group meeting on this topic.

Generis -

*An Intelligent Knowledge-Based Management System
Jim Williamson - Deductive Systems Ltd., Glasgow*

Generis provides an intelligent knowledge engineering environment based on entity relationship database technology. This has been extended with object-oriented representation within a semantic network structure utilising the unique generic relational data model. Generis incorporates a comprehensive rule base together with integral spreadsheet, document management and intelligent query language.

*A Model-based Diagnostic System for UNIX
Gail Anderson - AIAI, University of Edinburgh*

A model of the UNIX Operating System based on file system structure is developed. Application of the model is described and current work on the project is outlined.

*Experience with a Graphics Processor as an X-Server
Stephen McInnes - Silicon Products Ltd., Glasgow*

This paper describes the problems encountered by the author in using a novel graphics processor to provide a high performance X-Windows server. This server will allow up to eight users to run graphics application programs under the X Window system. One application area will briefly be described—the system's use as a VLSI design tool.

So You Think Your Network Is Slow?

William Armitage - University of Nottingham

Phil Karn (KA9Q) has written a C based implementation of TCP/IP for packet radio applications that runs on a number of cheap machines, e.g., PC clones, Amigas. This code is public domain. As well as bringing familiar services to these small machines it is a useful network gateway for SLIP and ethernets. A number of the 4.3Tahoe network congestion handling enhancements arose from experience gained with this package in the much lower bandwidth environment of packet radio. This paper examines the package, its novel solutions to certain problems and some applications.

AUUG Management Committee Meeting

3rd February, 1989

MINUTES

The meeting opened at 10:20 with the following committee members present: Secretary Tim Roper (TR), Treasurer Michael Tuke (MT), Chris Maltby (CM), Frank Crawford (FC) and Rich Burridge (RB). Also present was the AUUGN Editor John Carey (JC). In the absence of the President Greg Rose (GR), CM was elected to the Chair.

1. Apologies

Apologies were received from GR and Tim Segall.

2. Minutes of last meeting (28th October, 1988)

Moved FC/MT That the minutes of the last meeting be accepted.
Carried.

3. Business arising from Minutes

Re 3(a), TR reported that the donation to Wollongong Hospital had been sent and acknowledged.

Re 3(e) TR reported that the stationery had been printed and delivered at a total cost of \$1437.10 and tabled samples.

Re (8) TR reported that most of the money from AUUG88 had been received from ACMS.

Re (9) TR reported that reminder notices had not yet been sent.

Re (10) TR reported that the letter to Dave Horsfall terminating any arrangement for him to sub-edit a USENET section for AUUGN had been sent.

Re (13) TR had not prepared the timetable, GR had approached Pat Duffy to take on the marketing of AUUG89, TR had discussed alternative financial arrangements with Wael Foda of ACMS, Peter Barnes had agreed to be Programme Committee Chair.

Re (15) TR reported that the letter to PC89 advising them of the decision to not accept their invitation to endorse the UNIX feature at OFFICE TECHNOLOGY 89 section had been sent.

Re (16) TR reported that he was awaiting the invoice from University of California Press for fifty copies of Computing Systems for USD1310.40. Also that no action had yet been taken on purchasing /usr/group Product Directories for Institutional members.

4. President's Report

In the absence of GR there was no President's report.

5. Secretary's Report

TR reported on the following.

(a) The Inaugural Software Distribution

Approximately fifteen orders had been received and filled

so far; more were expected once AUUGN Vol 9 No 6 was posted. Material published about this offer was tabled.

(b) Nutshell Handbook Offer

Orders totalling over \$5000 had been received; more were expected once AUUGN Vol 9 No 6 was posted. The difference between the total orders so far and the approved expenditure of \$5000 was covered by prepayments. The order with O'Reilly and Associates was expected to be placed very soon. Material published about this offer was tabled. The issue of whether this activity would attract sales tax had arisen. A warning that we may have to charge extra to cover sales tax had been posted on ACSnet but had not made it into AUUGN Vol 9 No 6.

(c) OpenLook Specification

Ten copies of the OpenLook specification left at AUUG88 by Larry Crume of AT&T had been made and sold at cost to members.

(d) Membership

Numbers of financial members were 38 Institutional, 239 Ordinary and 8 Student. Unfinancial members totalled 6 Institutional and 21 Ordinary. There were 20 newsletter Subscriptions.

TR tabled correspondence sent and received since the last meeting.

Moved RB/FC That the Secretary's report be accepted.
Carried.

6. Treasurer's Report

MT presented a statement of the group's financial position (see attached). MT reported that approximately \$2000 per month had been deposited over the last two months from membership fees. There was discussion of whether to withdraw some money from a maturing term deposit to cover the extraordinary expenditure expected soon. It was resolved to not do so.

Moved TR/RB That the Treasurer's report be accepted.
Carried.

7. AUUGN Editor's Report

JC reported that AUUGN Vol 9 No 6 was at the printers but had been delayed by the need to reprint at their cost on account of them leaving a page out. JC asked for opinions about the quality of the printing as he had received a complaint about one copy being bound back-to-front and another having a page blank. David Purdue was making a new cover design and was storing and handling back issues. Despite the letter to David Horsfall a submission from him had been received for Vol 9 No 6. This would be treated as an ordinary submission.

Moved TR/MT That the Editor's report be accepted. Carried.

8. 1989 Summer Meetings

TR reported that the speaker from overseas for the summer meetings had been forced to withdraw by a change in

circumstances. There was discussion about what to do about the schedule meetings of which there were seven. TR suggested that local organisers be given the option of postponing or going ahead with a local programme possibly supplemented by an interstate speaker funded by AUUG. TR indicated that he thought that Perth would probably go ahead, Darwin and Brisbane would postpone and the remainder were unknown. There was discussion of possible speakers.

9. 1989 Winter Conference and Exhibition

The theme *No one ever got fired for buying UNIX* was confirmed. Pat Duffy (PD) arrived at 12 noon.

TR reported that Dennis Ritchie and Sunil Das had accepted invitations to attend and speak as guests of AUUG. RB reported that James Gosling had accepted the invitation to attend and speak, funded half-half by AUUG and Sun Microsystems.

PD suggested that a speaker from a major commercial user of UNIX be invited and agreed to look for one.

There was length discussion about various aspects of AUUG89.

The meeting adjourned for lunch at 12:45 and resumed at 13:30. Discussion of AUUG89 continued, particularly about publicity.

Wael Foda (WF) arrived at 14:00 and reported on preparations and bookings for the exhibition and sponsorships. WF presented two alternative financial arrangements between AUUG and ACMS for AUUG89. He indicated that he preferred a fixed fee arrangement rather than a percentage.

After WF left there was discussion about the financial arrangement options. It was resolved that we should stay with the percentage arrangement. Fees were set at \$240 for members, \$320 for non-members, \$50 for late registrations, \$150 for day registrations (with no late fee), \$50 for extra dinner tickets and \$20 for extra cocktail reception tickets.

It was resolved that the Secretary should write to ACMS re the financial arrangements and fees.

10. Appointment of next AUUGN Editor

TR reported that David Purdue had volunteered to become the new Editor.

Moved TR/FC That David Purdue be appointed AUUGN Editor on the retirement of John Carey. Carried.

11. Budget FY 1988/89

MT tabled the following.

AUUG Inc Budget 1988-89

As presented/discussed at October 1988 Meeting.

INCOME

Membership	17000
Interest	2700
AUUG88 Profit	14000

	\$33700

EXPENDITURE

4 Committee Meetings	7000
AUUGN (6 issues)	20000
Stationery	1500
Couriers & Freight	500
Bank Charges	150
Accountant	1000
Trademark Protection	500

	\$40650

PROFIT (LOSS) (\$6950)

Note: It was agreed to underwrite State meetings up to a total cost of \$10000.

Revised as at 1 February 1989

INCOME

Membership	22000
Interest	2700

	\$24700

EXPENDITURE

4 Committee Meetings	7000
AUUGN (6 issues)	20000
Stationery	1500
Couriers & Freight	500
Accountant	1000
Trademark Protection	500
AUUG88 Loss (\$11600-\$12400)	800
Computing Systems Journal	1650

	\$33100

PROFIT (LOSS) (\$8400)

Notes: 1) It was agreed to underwrite State meetings up to a total cost of \$10000.

2) A number of activities are expected to be revenue neutral. These include OpenLook Spec, Nutshell Handbooks and Usenix Journals.

3) The sum of \$12705.61 being return from AUUG87 has been returned to the general account in this period.

12. Secretarial Assistance
GR was absent but as far as anyone knew no action had been taken.

13. Constitutional Changes
No action. TR to contact Robert Elz.

14. Other Business
(a) There was discussion of the use of telephone conference calls between meetings.

(b) TR reported that CiTR, a commercial organisation attached to the University of Queensland, and offered to host AUUG90, both conference and exhibition. There was general satisfaction with the way that ACMS had handled previous exhibitions and agreement that the relationship between AUUG and ACMS should not be broken for the sake of one conference and exhibition.

Moved RB/MT That the exhibition part of AUUG90 be done by ACMS. Carried.

(c) TR reported that an application for a refund of AUUG88 registration fees had been received from someone unable to attend due to illness. It was resolved that in general no refunds be allowed after the nominated date.

Moved RB/MT In the case in point a discount of \$100.00 at AUUG90 would be offered as an act of good faith.
Carried.

15. Next Meeting
It was resolved that the next meeting should be held in Melbourne on 12th May, 1989.

Adopted 31/5/87
Rules for
the Australian Unix systems User Group Incorporated

NAME

1. The incorporated association shall be known as the **AUUG Incorporated**, abbreviated hereinafter to AUUG.

DEFINITIONS

2. (1) In these rules, unless otherwise stated:
“he”, “him” and “his” shall also be construed to mean “she”,
“her” and “her” respectively;
“The Act” means the *Associations Incorporation Act 1981* (Vic).
“Financial year” means the period from 1 June to 31 May;
“General Committee Member” shall mean a general member of the Management Committee;
“mail” shall imply the transmission of information in written or printed form, first-class pre-paid, via the general post or public or private courier service;
“unfinancial member” shall mean any member whose most recent term of membership has expired and who has not yet paid the subscription for the next twelve month period;
“voting member” shall mean any member entitled to cast a vote.
- (2) In these Rules, a reference to the secretary of the AUUG is a reference:
 - (a) where a person holds office under these Rules as Secretary of the AUUG, to that person; and
 - (b) in any other case to the Public Officer of the AUUG.
- (3) Words or expressions in these rules shall be interpreted in accordance with, and subject to, the Act as in force from time to time.
- (4) If any doubt arises as to the proper construction or meaning of any clauses in these Rules, the decision of the Management Committee thereon shall be final and conclusive provided such decision be reduced to writing and recorded in the minutes of a meeting of the Management Committee.

AIMS

3. The aims for which the AUUG is established are to promote knowledge and understanding of the UNIX system, and of similar or related computer systems.

For the furtherance of these aims and to achieve its purposes, the AUUG may carry out any or all of the following activities: conduct technical meetings, conferences, discussion groups, panels, lectures and other types of meeting; prepare and distribute a newsletter and other publications; collect software and distribute said software to its members for their use; verify licences of members for the purposes of administering the services of the AUUG; subscribe to or cooperate with or affiliate with or amalgamate with other associations formed elsewhere with similar aims; accumulate assets; and establish and promote other activities not included in the above list consistent with its aims for the benefit of its members.

ELIGIBILITY FOR MEMBERSHIP

4. Any individual or organisation who subscribes to the aims of the association, and who agrees to be bound by its rules and regulations and who has not been previously expelled from the association shall be eligible to join the AUUG.
5. An application for membership shall be in writing on the form approved by the Management Committee and shall provide such information as shall from time to time be prescribed by the Management Committee.
6. (1) Membership shall become current on the first day of the month following the date on which a valid membership application accompanied by payment of the appropriate entrance fee plus annual membership subscription is received by the Secretary, and shall continue for twelve months from that date.
(2) Upon completion of the initial membership period and any subsequent periods, membership may be renewed for a further period of twelve months by payment of an additional annual membership subscription.

7. (1) There shall be four classes of members: Ordinary members, Institutional members, Student members and Honorary Life Members.
 - (2) Any natural person who is eligible to be a member may become an Ordinary Member.
 - (3) Any person or organisation who is eligible to be a member may become an Institutional Member.
 - (4) Any full-time student who is eligible to be a member may become a Student Member.
 - (5) Any person who is an Ordinary Member of at least five years standing and who has rendered special services to the AUUG may be elected via a ballot of the members as an Honorary Life member.
 - (6) If before the first day of May the Secretary receives a petition from at least twenty voting members requesting the election of a member of the AUUG to the position of Honorary Life Member, then he shall arrange a ballot of the membership on this question to be conducted in conjunction with the annual election of Officers and General Committee Members.
8. All Ordinary, Institutional and Honorary Life Members whose membership is current shall be entitled to cast a vote.

MEMBERSHIP SUBSCRIPTIONS AND FEES

9. The Management Committee shall determine before the commencement of each financial year a scale of fees for entrance to the AUUG, for annual subscriptions and for the attendance at meetings, for each class of members to be applied during that financial year.

REGISTER OF MEMBERS

10. (1) The Secretary shall keep and maintain a register of members in which shall be entered the full name and address of each member and the register shall be available for inspection by members at the address of the Public Officer.
- (2) Nothing in the previous subsection shall entitle any member to make a copy of the register of members, except with the permission of the Management Committee, and on such terms and conditions as the Management Committee shall from time to time determine.

TERMINATION OF MEMBERSHIP

11. (1) A member may resign his membership at any time by giving notice in writing to the Secretary. No member who resigns shall have any claim for a refund of subscriptions paid.
- (2) A member who has been unfinancial for more than two calendar months shall be deemed to have resigned his membership, and shall no longer be entitled to any privileges enjoyed by members.
- (3) Former members who have resigned will be entitled to rejoin the AUUG on the same basis as new members joining the AUUG.

EXPULSION OF MEMBERS

12. Upon receipt of a petition so requesting from twenty or more members, or half the membership, whichever is less, the Management Committee shall call upon any member to explain any alleged misconduct, and the Management Committee shall have power to suspend or expel any member who in its opinion has either been guilty of misconduct or has acted prejudicially to the interests of the AUUG or who has wilfully infringed any of the Rules of the AUUG.

GENERAL MEETINGS

13. The Annual General Meeting shall be held within the second half of each calendar year. The date and general location of each Annual General Meeting shall be determined at the preceding Annual General Meeting but either the date or location or both may be changed by the Management Committee if it proves impossible or highly inconvenient to meet at the location previously selected or on the date previously selected.
14. An ordinary general meeting of the AUUG shall be called by the Management Committee in conjunction with any technical meeting or conference or other function where attendance by a quarter or more of the voting members is expected by the Management Committee.
15. (1) Written notice of the time and place for each meeting and its agenda shall be mailed to each voting member of the AUUG at least four weeks before the date of the meeting.
- (2) Business conducted at such meetings shall be confined to matters included in the written agenda, reports from Officers, and resolutions instructing the Management Committee to conduct a formal ballot of the membership on matters of substance. Such resolutions shall not be binding on the Management Committee unless the meeting was attended by at least twenty voting members, or half the membership,

- whichever is less, and the resolution was supported by at least three-quarters of the members voting.
- (3) All voting members shall be entitled to cast one vote.
 - (4) Any voting member may award his proxy to another voting member for the period of a single General meeting providing he so notifies the Secretary in writing at least 24 hours before the appointed time of commencement of the meeting.
16. (1) Upon receipt of a petition so requesting from twenty or more members, or half the membership, whichever is less, the Secretary shall call an Extraordinary General meeting of the AUUG for a date no later than two calendar months after receipt of the petition.
- (2) The business of the meeting shall be confined to matters described in the petition and to other matters specifically provided for in these rules and recorded in the written agenda sent to all members by mail at least four weeks before the date set for the meeting.
17. (1) For each general meeting, the quorum shall be twelve members personally present and entitled to vote.
- (2) If within half an hour after the appointed time for the commencement of a general meeting, a quorum is not present, the meeting if convened upon the requisition of members shall be dissolved and in any other case the members present (being not less than 5) shall be a quorum.
18. At all general meetings of the AUUG the Chair shall be taken by the President, or in his absence, by a member elected by the meeting.

OFFICERS

19. The Officers of the AUUG shall be: the **President**; the **Secretary**; the **Treasurer**; the **Returning Officer**; and the **Assistant Returning Officer**;

MANAGEMENT COMMITTEE

20. The management and control of the business and general affairs of the AUUG shall be vested in a Management Committee of seven members, namely: the President; the Secretary; the Treasurer; and four General Committee Members.

ELECTIONS

21. (1) The election of Officers and General Committee Members shall be by a postal ballot held annually.
 - (2) Nominations for each position shall be received by the Secretary up until the first day of May each year. Each nomination must be in writing, must name the position or positions sought, must be signed by at least three voting members, and must be countersigned by the nominated member who must be a financial voting member of the AUUG.
 - (3) The Management Committee shall ensure that at least one valid nomination is obtained for each position such that if no further nominations are received all offices and positions may be filled. Where only one valid nomination is received for a particular position by the close of nominations, the nominee shall be declared elected forthwith, and no ballot for that position shall be held.
 - (4) Within first seven days of May, the Secretary shall advise the Returning Officer of all valid nominations received, and if a ballot is required shall advise him of a date no later than the fifteenth day of May for the ballot for all contested positions, and shall provide him with a list of voting members.
 - (5) While any Ordinary Member may be nominated to more than one office or position, no person shall be elected to more than one position. Ballots shall be determined in the following order: for President, for Secretary, for Treasurer, for General Committee Members, and for Returning Officers.
 - (6) All voting members shall be entitled to cast one vote.
22. The term of office for all Officers and General Committee Members shall be for one year, from July 1 to June 30.

VACANCIES ON THE MANAGEMENT COMMITTEE

23. (1) The position of any General Committee Member shall be vacated if the member fails to attend any Management Committee meeting without furnishing a satisfactory explanation as to the cause of his absence, and if the Management Committee resolves that his office be vacated.
- (2) If at any time any of the principal Officers (President or Secretary or Treasurer) be unable to continue in office for any reason, then the Management Committee shall appoint one of their number to the vacant office.

- (3) Should a vacancy occur among the other Officers, or among the General members of the Management Committee, then the Management Committee shall appoint an Ordinary Member of the AUUG to fill the vacancy.
- (4) The Management Committee shall make the approval of such appointments an order of business for the next General Meeting of the AUUG if any such meeting will be held before the next election of Officers and General Committee Members.

MANAGEMENT COMMITTEE MEETINGS

- 24. (1) The Management Committee shall meet formally at least twice per year.
 - (2) Notification of time, place and agenda for each meeting shall be made in writing to each member of the Committee by the Secretary at least four weeks in advance.
 - (3) All members of the AUUG are entitled to be present at such meetings, and may speak when invited by the Chairman, but only members of the Management Committee may vote.
- 25. At meetings of the Management Committee the President shall take the chair, or in his absence, a member of the Management Committee elected by the meeting.
 - 26. The quorum for such meeting shall be four. Resolutions of the committee shall require a simple majority of the members present and voting. The chairman shall have a casting vote in the event of a tie.

DISTRIBUTION OF INCOME

- 27. The income and property of the AUUG however derived shall be applied solely towards the aims and purposes of the AUUG as set out in these Rules, and no portion thereof shall be paid or transferred directly or indirectly by way of dividend to any member of the AUUG at any time.
- 28. The AUUG shall not appoint a person who is a member of the Management Committee to any office in the gift of the association to the holder of which there is payable any remuneration by way of salary, fees or allowances.

29. Notwithstanding the previous section the AUUG may compensate the reasonable expenses actually incurred by any member in the conduct of the business of the AUUG under the direction of the Management Committee.

CHAPTERS

30. (1) Ten or more members of the AUUG may petition the Management Committee to form a **chapter** of the AUUG.
- (2) General rules for the organisation, operation, obligations and privileges of chapters shall be as resolved by the Management Committee or the membership as a whole from time to time.
- (3) Each chapter shall appoint a chapter committee consisting of at least a Chapter Chairman and a Secretary/Treasurer.
- (4) The chapter committee may convene meetings consistent with the aims of the AUUG, but may not enter into any financial commitments on behalf of or in the name of the AUUG except with the written approval of the Management Committee.

AFFILIATION OR AMALGAMATION WITH OTHER ORGANISATIONS

31. The Management Committee may at any time seek or discuss the possibility of affiliation or amalgamation with any other organisation whose aims are similar to or compatible with those of the AUUG. No agreement for affiliation or amalgamation may be finalised until the matter has received the assent of three-quarters of the members voting in a postal ballot.

DISSOLUTION OF THE AUUG

32. (1) Upon receipt of a petition requesting the dissolution of the AUUG from twenty or more members, or half the membership, whichever is less, the Secretary shall arrange for the question to be put to the membership by ballot no later than one month after the date that he receives the petition.
- (2) If three-quarters of the members voting agree, the AUUG shall be dissolved.
- (3) If upon the dissolution of the AUUG there remains after satisfaction of all its debts and liabilities any property whatsoever, the same shall not be paid to or distributed among the members or Chapters if any, but shall be given or transferred to some public educational institution, or other institution to be determined at or before the time of dissolution by resolution of the membership.

CHANGES TO THE RULES

33. Changes to these Rules may be initiated at the request of a General meeting, or by the Management Committee. All proposed changes must be approved by a three-quarters majority of the votes received in a postal ballot of the members before having effect.

RIGHTS OF MEMBERS

34. (1) Each member shall be entitled to attend all meetings of the AUUG, including meetings of the Management Committee, provided any prescribed attendance fee is paid.
(2) Each member shall be sent a copy of the association's newsletter.
(3) Each member entitled to vote in a ballot shall be sent notice in writing of all ballots and copies in writing of the annual reports of the Secretary and Treasurer.

THE SECRETARY

35. (1) The Secretary shall furnish to the Returning Officer a complete list of all voting members whenever this is required for the conduct of a ballot.
(2) The Secretary shall keep or cause to be kept full and correct minutes of all resolutions and proceedings at General meetings and Management Committee meetings of the AUUG.
(3) The Secretary shall conduct correspondence on behalf of the AUUG.
(4) The Secretary shall, during his last month of office, prepare a written report on the state of the affairs of the AUUG for distribution to the membership.

THE TREASURER

36. (1) The Treasurer shall keep or cause to be kept correct accounts and books and records showing the financial affairs of the AUUG.
(2) The Treasurer shall notify the President and Secretary in writing of the usual location of said accounts, books and records whenever this location is changed.
(3) The Treasurer shall receive all fees and subscriptions and all other monies on account of the AUUG and provide receipts for the same. The Treasurer shall deposit all monies received into a bank account maintained by the AUUG.

- (4) The Treasurer shall receive accounts for payment for services rendered to the AUUG, and as directed by the Management Committee arrange for payment from the AUUG's account.
- (5) The Treasurer shall, during his last month of office, prepare or cause to be prepared a written report on the financial affairs of the AUUG for distribution to the membership.
- (6) The accounts and books referred to in sub-clause (1) shall be available for inspection by members.

FUNDS

- 37. The funds of the AUUG shall be derived from entrance fees, annual subscriptions, donations and such other sources as the Management Committee determines.
- 38. (1) Signing Officers for the AUUG's accounts shall be the President, the Secretary, the Treasurer and one other General Committee Member chosen by the Management Committee.
 - (2) All cheques, drafts, and other orders for payment of money out of the funds of the AUUG, if for less than a limit established by the Management Committee, may be signed by only one Signing Officer.
 - (3) For other amounts, each such instrument must be signed by at least two Signing Officers.

SEAL

- 39. (1) The Common Seal of the AUUG shall be kept in the custody of the Secretary.
 - (2) The Common Seal shall not be affixed to any instrument except by authority of the Management Committee and the affixing of the Common Seal shall be attested by the signatures either of two members of the Management Committee or of one member of the Management Committee and the Public Officer of the AUUG.

EXECUTION OF CONTRACTS

- 40. The Management Committee, except as otherwise provided in these Rules, may prospectively or retroactively authorise any Officer or member of the AUUG to enter into any contract or execute and satisfy any instrument, and any such authority may be general or confined to specific instances, except that any contract whose dollar value exceeds an amount predetermined by the Management Committee must be specifically authorised in advance by the Management Committee.

VOTING

41. (1) All voting by the members with respect to the election of Officers and General Committee Members, with respect to the election of Honorary Life Members, with respect to changes to these Rules, and all other substantive matters shall be conducted by postal ballot.
- (2) Every voting member of record as of the date of entry of a ballot into the mails shall be entitled to vote in the ballot.
- (3) On all questions to be put to a ballot, the Secretary shall designate a date for the ballot to be placed in the mails, and the due date shall be four weeks after that date.
- (4) The Returning Officer shall nominate the address to which voters shall return completed ballot papers by mail.
- (5) A ballot will not be counted if it is received after the due date or if the ballot paper does not comply with the instructions printed on it.
- (6) The ballots will be received by the Returning Officer, and counted by him and the Assistant Returning Officer.
- (7) The Returning Officer shall report the result of the ballot in writing to the Secretary no later than two weeks after the due date.

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

THIS PAGE INTENTIONALLY LEFT BLANK

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

- Renewal/New* Membership of the AUUG \$78.00
 Renewal/New* Student Membership \$45.00 (note certification on other side)
 International Surface Mail \$20.00
 International Air Mail \$60.00 (note local zone rate available)

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
PO 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

<input type="checkbox"/> New/Renewal* Institutional Membership of AUUG	\$325.00
<input type="checkbox"/> International Surface Mail	\$ 40.00
<input type="checkbox"/> International Air Mail	\$120.00
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	\$ 90.00
<input type="checkbox"/> International Surface Mail	\$ 20.00
<input type="checkbox"/> International Air Mail	\$ 60.00

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# _____