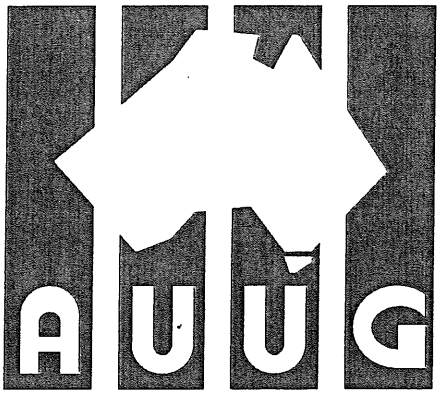
ISSN 1035-7521



Australian UNIX systems User Group Newsletter

AUUGN

Volume 13, Number 6

December 1992

Registered by Australia Post, Publication Number NBG6524

The AUUG Incorporated Newsletter

Volume 13 Number 6

December 1992

CONTENTS

AUUGN is the journal of AUUG Incorporated, an organisation with the aim of promoting knowledge and understanding of Open Systems including but not restricted to the UNIX* system, networking, graphics, user interfaces and programming and development environments, and related standards.

---

* UNIX is a registered trademark of UNIX System Laboratories, Incorporated

AUUG General Information

Memberships and Subscriptions

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

Membership and General Correspondence

All correspondence for the AUUG should be addressed to:-

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

Phone: (02) 361 5994
Fax: (02) 332 4066
Email: auug@munnari.oz.au

AUUG Business Manager

Liz Fraumann,
P.O. Box 366,
Kensington, N.S.W. 2033.
AUSTRALIA

Phone: +61 2 953 3542
Fax: +61 2 953 3542
Email: eaf@softway.sw.oz.au

AUUG Executive

President

Phil McCrea
phil@softway.oz.au
Softway Pty. Ltd.
79 Myrtle Street
Chippendale NSW 2008

Vice-President

Glenn Huxtable
glenn@cs.uwa.oz.au
University of Western Australia
Computer Science Department
Nedlands WA 6009

Secretary

Peter Wishart
pjw@lobo.canberra.edu.au
EASAMS Australia
Level 6
60 Marcus Clark St.
Canberra ACT 2600

Treasurer

Frank Crawford
frank@atom.ansto.gov.au
Australian Supercomputing Technology
Private Mail Bag 1
Menai NSW 2234

Committee Members

Rolf Jester
rolf.jester@sno.mts.dec.com
Digital Equipment Corporation
P O Box 384
Concord West NSW 2138

Chris Maltby
chris@softway.sw.oz.au
Softway Pty. Ltd.
79 Myrtle Street
Chippendale NSW 2008

John O'Brien
john@wsa.oz.au
Whitesmiths Australia P/L
#5 Woods Centre
ANSTO Business & Tech. Park
Lucas Heights NSW 2234

Michael Paddon
mwp@iconix.oz.au
Iconix Pty Ltd
851 Dandenong Rd
East Malvern VIC 3145

Greg Rose
ggr@acci.com.au
ACCI
723 Swanston St
Carlton VIC 3053

AUUG General Information

Next AUUG Meeting

The AUUG 1993 Summer Conference Series are to be held between February and April 1993 (see later in this issue for more details).

The AUUG'93 Conference and Exhibition will be held from the 28th to 30th September, 1993, at the Sydney Convention and Exhibition Centre, Darling Harbour, Sydney (see page 15 in this issue).

Advertising

Advertisements to be included in AUUGN are welcome. They must be submitted on an A4 page. No partial page advertisements will be accepted. Advertising rates are $300 for the first A4 page, $250 for a second page, and $750 for the back cover. There is a 20% discount for bulk ordering (ie, when you pay for three issues or more in advance). Contact the editor for details.

Mailing Lists

For the purchase of the AUUGN mailing list, please contact the AUUG secretariat, phone (02) 361 5994, fax (02) 332 4066.

Back Issues

Various back issues of the AUUGN are available. For availability and prices please contact the AUUG secretariat or write to:

AUUG Inc.
Back Issues Department
PO Box 366
Kensington, NSW, 2033
AUSTRALIA

Conference Proceedings

A limited number of the Conference Proceedings for AUUG'92 are still available, at $50 each. Contact the AUUG secretariat.

Acknowledgement

This Newsletter was produced with the kind assistance of and on equipment provided by the Australian Nuclear Science and Technology Organisation.

Disclaimer

Opinions expressed by authors and reviewers are not necessarily those of AUUG Incorporated, its Newsletter or its editorial committee.

AUUG Newsletter

Editorial

Welcome to AUUGN Volume 13 Number 6, the last issue for 1992. This issue has been delayed due to the printer being on holiday over the Christmas break. Hope everyone had a good Christmas and wish you all a prosperous New Year.

In this issue we have the results of the FaceSaver at AUUG'92 (it is nice to put faces to various names). In terms of AUUG announcements we have the summer conferences, AUUG'93 and various discounts for AUUG members. Two letters to the editor are included. It would be nice to get more of these.

In the !AUUGN section we have a copy of Piers Lauder's paper from AUUGN Volume 2, Number 6, showing where ACSnet came from. (Is this the first mention Piers?)

The formation of SAGE is rolling along and some notes have been included to keep people in touch.

Finally, we do have a couple of papers which should convey something to those after technical information.

Jagoda Crawford

AUUGN Correspondence

All correspondence regarding the AUUGN should be addressed to:-

AUUGN Editor,
P.O. Box 366,
Kensington, N.S.W. 2033.
AUSTRALIA

Phone: +61 2 717 3885
Fax: +61 2 717 9273
Email: auugn@munnari.oz.au

AUUGN Book Reviews

The AUUGN Book Review Editor is Dave Newton. David has no network access at present, so please contact the AUUGN editor for more details. A number of books are available for review, please keep an eye on aus.auug for books available.

Contributions

The Newsletter is published approximately every two months. The deadlines for contributions for the next issues of AUUGN are:

Volume 14 No 1	Friday 29th January
Volume 14 No 2	Friday 26th March
Volume 14 No 3	Friday 28th May
Volume 14 No 4	Friday 30th July
Volume 14 No 5	Friday 24th September
Volume 14 No 6	Friday 26th November

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

I prefer documents to be e-mailed to me, and formatted with troff. I can process mm, me, ms and even man macros, and have tbl, eqn, pic and grap preprocessors, but please note on your submission which macros and preprocessors you are using. If you can't use troff, then just plain text or postscript please.

Hardcopy submissions should be on A4 with 30 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.

THE UNIVERSITY OF NEW SOUTH WALES

P.O. BOX 1 • KENSINGTON • NEW SOUTH WALES • AUSTRALIA • 2033
TELEX AA26054 • TELEGRAPH: UNITECH, SYDNEY • TELEPHONE 697 2222

DEPARTMENT OF COMPUTER SCIENCE FAX (02) 313 7987
ASSOCIATE PROFESSOR J. LIONS TELEPHONE (02) 697 4071

Dr J. Crawford,
Australian Nuclear Science &
Technology Organisation,
Private Mail Bag No. 1,
Menai, NSW 2234.

November 18, 1992

Dear Jagoda,

Residual Uses for Troff

I still use *troff* and I like to write my own macro packages, since this is what I have always done. My first exposure to text formatting was via *roff* which did not come with any macro packages such as *ms* or *mm* or *me*.

I used *roff* to produce a newsletter for the Computer Users Society at the University of New South Wales in the mid-1970's. This society was formed by Warren Hastings of the School of Mechanical Engineering with the (implicit) goal of pressuring the university administration to improve the campus computing facilities. In 1974 the ageing IBM 360/50 had been replaced by a CDC Cyber computer supported by several PDP11/40 based batch stations. Our school had a 'large' batch station (208K bytes of core memory and three RK05 (2.5 Mbytes each) of disk memory. Mechanical Engineering and other schools were jealous ...

The newsletter was called *CUSWords*, and was printed on the PDP 11/40 line printer (upper case only). Copies were individually addressed, and both sides of the paper were used if the issue was particularly large. *CUSWords* succeeded in its purpose (for us the PDP11/40 was eventually supplemented by a PDP11/70), and eventually the Computer Users Society quietly folded its tents.

Troff

The author of *troff* was Joe Ossana who died prematurely in a car accident about 20 years ago. Development of *troff* thus ended suddenly. It is a great tribute to its author that the version we still use is still essentially what he had planned.

Early versions of *troff* were left with several subtle bugs that have been fixed over the years, principally by Brian Kernighan at BTL. I now find it reliable and I am happy that the specification is completely static. It provides most of the facilities that I needed, and few of the hassles. However it does have its limitations, especially with two-character macro and string names, and no way to hide internal names. These are less of a problem if you write your own macro packages.

I have my own system for generating letters (such as this one). Naturally it was based first on *nroff*, and later on *troff*. This system was implemented originally when I was editor of the Australian Computer Journal (1982-1987). As editor I generated about 1000 items of correspondence per year. Most letters were standard (e.g. to acknowledge receipt of a paper or

report) and only about 10% needed customising to any degree. Nearly all letters fitted on a single sheet of paper, and were printed on a Qume printer that, while it provided the full ASCII character set, was not sophisticated by today's standards.

Now that I am no longer ACJ editor I still use my letter generating software but in a different way. 100% of letters are customised, and many run to several pages. The Qume impact printer was replaced several years ago by an Apple Laserwriter — the first of a series of Postscript laser printers.

As editor I was always concerned about *orphans* and *widows* — respectively the first lines or the last lines of paragraphs that become separated from their fellows by a page break. There seem to be two approaches to handling these (apart from just refusing to recognise or acknowledge their existence): (1) judicious rewording of the text; and (2) adjusting the vertical spacing of the formatted text so that the page break appears *between* paragraphs.

It is amazing how often adjusting just one or two words or phrases in the right place can resolve a problem. This approach requires some skill and experience and is not guaranteed to work every time. Sometimes more draconian changes are needed. An alternative approach involves adjusting the interline spacing by whatever is needed (sometimes by as little as 1%). The formula for adjusting the vertical spacing is simple:

*new_spacing = ((desired vertical size) * old_spacing) / (actual vertical size)*

The desired vertical size is the size of the space available. In applying this formula one needs to remember that *troff* arithmetic is all integer: fractions are simply truncated, so the multiplication should be performed before the division. The result could be rounded but this is usually not worth worrying about.

I have devised two macros for vertical adjustment for inclusion in other macro files:

```
.de A_
.br
.nr L_ \\$2
.\"***************** execute first argument for size
.di Y_
.\\$1
.br
.di
.nr D_ \\n(dn \" size of last completed diversion
.ie \\n(D_>0 \{\
.\"***************** recalculate vertical spacing
.nr V_ \\n(L_*\\n(.v/\\n(D_u
.if \\n(V_>\\n(.v .tm BIGGER NOW \\n(V_ \\n(.v
.vs \\n(V_u \}
.\"*****************
.el tm ERROR: Diversion Size is Zero
.\"***************** reserve space and go for it
.ne \\n(L_u
.\\$1
.br
..
```

```
.\"***************** end A_ ; begin PP
.de PP
.br
.ne 27c  \" new page ?
.sp |2c
.A_ \\$1 25c
..
.\"***************** end PP
```

The basic macro is called A_ and requires two arguments: another troff macro name, and a desired size (suitably expressed according to usual *troff* conventions). The first argument is executed and its output is diverted into the macro L_. The size of the diversion will have been left in the number register dn. The new vertical spacing is calculated and stored as V_. The .vs command resets the vertical spacing before the first argument is executed for the second time, without diversion.

Note that there is an implicit assumption that no footer macros will be fired off. Also this will not work directly if the first argument was obtained via e.g. preprocessing with eqn or tbl.

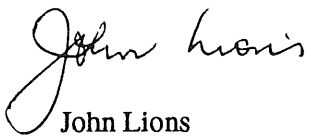
The second macro, PP, represents both an example of the use of A_ and the principal reason for developing it. It addresses the problem of producing output that will fit exactly on a single A4 page. A4 pages are 29.7 cm long, so a 2 cm margin at the top leaves room for a 2.7 cm margin at the bottom if the text size is 25 cm.

The text will be expanded or contracted as needed. For some purposes it is advisable to limit the degree of expansion to a maximum of say 20%. In other situations it may be desirable to consider varying the point size of the text characters to increase the range of possibilities.

I have done this for the most heavily used part of letter generation package, and I am well pleased with the result.

This brings me finally to my real reason for writing this letter. May I suggest that similar procedures could be applied to parts of AUUGN. I am thinking particularly of the *pro forma* membership that often overflow (unnecessarily I believe) beyond a single page. Every such page saved either saves expense for AUUG or provides you with the opportunity to print something more useful and more generally interesting.

Yours sincerely,



John Lions

280/mlc

From jeff@dialix.oz.au Thu Oct 22 10:20 EST 1992
Received: by uniwa.uwa.edu.au (5.65c)
id AA22249; Thu, 22 Oct 1992 08:19:38 +0800
From: Jeff Johnson <jeff@dialix.oz.au>
To: jc@atom
Subject: DIALix Services in Sydney
Cc: jeff@dialix.oz.au
X-Mailer: SCO Portfolio 2.0
Date: Thu, 22 Oct 1992 8:09:51 +0800 (WST)
Message-Id: <9210220809.aa01582@DIALix.oz.au>
Content-Type: text
Content-Length: 4162
Status: RO

Hi,

Liz Fraumann gave me your name some time back and I asked my "agent" in Sydney to contact you, but I am not sure that he has so I thought I'd drop a line and introduce myself.

I recently setup a dialin Unix system in Sydney that is on line specifically for individual and small business Internet accounts. Email and newsfeeds to and from the Internet are the main activities but, as I also intend to extend my services beyond Perth and Sydney, commercial file transfer will be available. Over 1,000 newsgroups are available for newsfeeds via uucp. News can also be read on-line.

Source code for C-news and the nn newsreader are on-line as well as public domain uucp emulator for PC-clones, Amiga, Atari and Apple.

I will include my standard "flyer" for your information.

Thanks,

JJ

Hi,

Thanks for your interest.

I operate DIALix Services which is a Mail Service Affiliate Member of AARNet.

DIALix offers E-mail and newsfeed to individuals and small companies who are unable to obtain or afford a direct AARNet connection.

Users connect to DIALix through multiple dial up lines on 300/1200/2400 and 9600 modems to send and receive E-mail or browse newsgroups and post news items. Currently DIALix is available at a local call fee in Perth and Sydney. Trunk callers are encouraged to ask about the cost of a trunk connection for Email. UUCP feeds are especially catered for.

The general public are actively sought to participate in reading newsgroups and receiving E-mail and posting to local newsgroups and E-mailing other DIALix users. Clubs and organisations are catered for in creation of special interest newsgroups and free login during club meeting hours (fortnightly or monthly:-).

Schools and special interest groups can even have their own newsgroup (i.e. dialix.schools adawa dialix.farming) so members and staff could have a "customised" bulletin board without the need for a local person to maintain the equipment and service.

Net write access is provided to users who meet the requirements of the AVCC ("some benefit to the higher education and research sector") with each applicant being considered on their merits. Users who have write access, may have that access terminated at any time, if they abuse the resources of the AARNet (or Internet).

A commercial traffic service is now available between Perth and Sydney.

Charges for DIALix are:-

All connect time 1c/minute ($10 minimum transaction [Corporate $25]).

Temporary hard disk storage space up to 1Mbyte. Additional storage space available at $10 per Megabyte per annum.

Optional extra, Net write access, for messages sent and received including the first Megabyte:-

Individuals (user@DIALix.oz.au) $80/annum ($10/month)

Corporate connect (users@site.DIALix.oz.au) $225 per annum per host

Interstate or overseas traffic in excess of 1Mbyte per annum costs 1c/Kbyte.

All fees are "in advance".

Visa, Bankcard and Mastercard are accepted via E-mail or phone. Cheques, etc. to the postal address below.

You can choose your own login name which is then unchangeable. It must be unique on DIALix and lower case letters and numbers beginning with a letter. Write to the postal address or phone me to arrange your login.

Please feel free to copy and distribute this to anybody who may be interested in DIALix.

JJ

Jeff Johnson (jeff@DIALix.oz.au)
Modem (09) 244-3233 Perth (02) 948-6918 Sydney
Phone (09) 244-2433 voice (All Hours)

DIALix Services
Box 371
South Perth 6151

AUUG Institutional Members as at 05/01/1993

Adept Software
Alcatel Australia
Allaw Technologies
Amdahl Pacific Services
ANI Manufacturing Group
ANSTO
Anti-Cancer Council of Victoria
ANZ Banking Group/I.T. Development
Attorney Generals' Dept
Attorney-General's Dept
Ausonics Pty Ltd
Auspex Systems Australia
Australian Airlines Limited
Australian Archives
Australian Bureau of Agricultural
and Resource Economics
Australian Bureau of Statistics
Australian Computing & Communications Institute
Australian Electoral Commission
Australian Museum
Australian Taxation Office
Australian Wool Corporation
Automold Plastics Pty Ltd
AWA Defence Industries
B & D Australia
Bain & Company
BHP CPD Research & Technology Centre
BHP Information Technology
BHP Petroleum
BHP Research - Melbourne Laboratories
BHP Research - Newcastle Laboratories
BICC Communications
Bond University
Burdett, Buckeridge & Young Ltd.
Bureau of Meteorology
C.I.S.R.A.
Cape Grim B.A.P.S
Capricorn Coal Management Pty Ltd
CITEC
Classified Computers Pty Ltd
Co-Cam Computer Group
Codex Software Development Pty. Ltd.
Colonial Mutual
Com Net Solutions
Com Tech Communications
Commercial Dynamics
Communica Software Consultants
Composite Buyers Ltd
Computer Sciences of Australia Pty Ltd
Computer Software Packages
Corinthian Engineering Pty Ltd
CSIRO
Curtin University of Technology
Customised Software Solutions Centre
Cyberdyne Systems Corporation Pty Ltd
Cyberscience Corporation Pty Ltd
Data General Australia
Deakin University
Defence Housing Authority
Dept of Agricultural & Rural Affairs
Dept of Defence
Dept of Education, Qld
Dept of Industrial Relations, Employment,
Training & Further Education
Dept of Planning & Housing
Dept of the Premier and Cabinet
Dept. of Conservation & Environment
Dept. of Defence
Dept. of the Premier and Cabinet
Dept. of the Treasury
Dept. of Transport
Easams (Australia) Ltd
EDS (Australia) Pty Ltd
Emulex Australia Pty Ltd
Equinet Pty Ltd
Ericsson Australia Pty Ltd
ESRI Australia Pty Ltd
FGH Decision Support Systems Pty Ltd
Fire Fighting Enterprises
Flinders University
Fujitsu Australia Ltd
G. James Australia Pty Ltd
GCS Pty Ltd
Geelong and District Water Board
Genasys II Pty Ltd
GeoVision Australia
GIO Australia
Golden Circle Australia
Great Barrier Reef Marine Park Authority
Gunnedah Abattoir
Haltek Pty Ltd
Hamersley Iron
Harris & Sutherland Pty Ltd
Hermes Precisa Australia Pty. Ltd.
IBM Australia Ltd
Iconix Pty Ltd
Insession Pty Ltd
Insurance & Superannuation Commission
Ipec Management Services
IPS Radio & Space Services
James Cook University of North Queensland
JTEC Pty Ltd
Knowledge Engineering Pty Ltd
KPMG Solutions
Land Information Centre
Land Titles Office
Leeds & Northrup Australia Pty. Limited
Liquor Administration Board (NSW Govt.)

AUUG Institutional Members as at 05/01/1993

Logica Pty Ltd
Logical Solutions
Mayne Nickless Courier Systems
McDonnell Douglas Information Systems Pty Ltd
Mentor Technologies Pty Ltd
Meridian Information Services Pty Ltd
Metal Trades Industry Association
Mitsui Computer Limited
Motorola Computer Systems
Multibase Pty Ltd
NCR Australia
NEC Australia Pty Ltd
Office of Fair Trading
Office of the Director of Public Prosecutions
Open Software Associates Ltd
Oracle Systems Australia Pty Ltd
Ozware Developments Pty Ltd
Philips PTS
Port of Melbourne Authority
Powerhouse Museum
Prentice Hall Australia
Prospect Electricity
pTizan Computer Services Pty Ltd
Public Works Department
Pulse Club Computers Pty Ltd
Qantek
Quality By Design Pty Ltd
Redland Shire Council
Release4
Rinbina Pty Ltd
Royal Melbourne Institute of Technology
SBC Dominguez Barry
Scitec Communication Systems
Sculptor 4GL+SQL
SEQEB Business Systems
SEQEB Control Centre
Siemens Nixdorf Information Systems Pty Ltd
Snowy Mountains Authority
Software Developments
Softway Pty Ltd
St Vincent's Private Hospital
Standards Australia
Steedman Science and Engineering
Steelmark Eagle & Globe
Sydney Electricity
Sydney Ports Authority
System Builder Development Pty Ltd
TAB of Queensland
Tattersall Sweep Consultation
Technical Software Services
Telecom Network Engineering Computer Support Services
Telecom Payphone Services
The Far North Qld Electricity Board
The Fulcrum Consulting Group
The Opus Group Australia Pty Ltd
The Roads and Traffic Authority
The University of Western Australia
TNT Australia Information Technology
Toshiba International Corporation Pty Ltd
Tower Technology Pty Ltd
Tradelink Plumbing Supplies Centres
Triad Software Pty Ltd
Turbosoft Pty Ltd
UCCQ
Unisys
University of Melbourne
University of New South Wales
University of Tasmania
University of Technology, Sydney
UNIX System Laboratories
Unixpac Pty Ltd
Victoria University of Technology
VME Systems Pty Ltd
Wacher Pty Ltd
Walter & Eliza Hall Institute
Wang Australia Pty. Ltd.
Water Board
Workstations Plus
Zircon Systems Pty Ltd

AUUG President's Report

Standards encourage competition

The world of open systems opens up competition at all levels in a computing system: hardware, operating system, communications, database, transaction monitor, and user interface. The boundaries between these layers, commonly referred to as Application Programming Interfaces (APIs), are publicized as standards, which are now generally adhered to by most technology producers.

The problem with standards is that they take such a long time to be compiled, because the committees who produce the standards often comprise special interest groups, each pushing their own particular barrow... This gives rise to defacto standards, which are so named because their popularity effectively makes them a standard. Examples are TCP/IP and X.

In an ideal open systems world, purchasers of computing systems would be able to select technologies at all levels from a number of technology providers. This is not yet the case because the relevant standards are not sufficiently defined. This leads to a trap: products may be advertised as being standards compliant, but may in fact also use a set of proprietary extensions to the standard, which generally ensures that portability is made more difficult, if not impossible.

Vendor lockin is alive and well

The attraction of open systems is that it removes vendor lockin: if one vendor's solution does not exactly fit the bill, or there are problems with quality, then the purchaser should be able to select an offering from another vendor.

But vendor lockin is still alive and well. The main culprits at present are the database vendors, who often use proprietary extensions to standards to provide attractive features, which of course provide product differentiation. But the lure of these attractive features frequently leads to vendor lockin.

Microsoft and standards?

The biggest culprit of all of course when it comes to vendor lockin is Microsoft, who exhibit a flagrant disregard for standards. You have to admire Microsoft as a corporation, but their main goal is one of total domination, which leaves no room for standards adherence, because adherence to standards leaves an organization more vulnerable to competition.

It seems like Microsoft has learned nothing from history: IBM tried to dominate the mainframe and communications world with its operating system and communications protocols, with the intention of locking vendors in to a software architecture under SAA. This strategy worked for quite some time, but has fallen apart with the fall in demand for mainframes. The request by users for open systems had all but decimated SAA.

Microsoft is the 'new age' IBM, in many respects. Like a delinquent child it has split from it mentor, IBM, in relation to operating system technology, in the belief that it can dominate that sector without IBM's assistance.

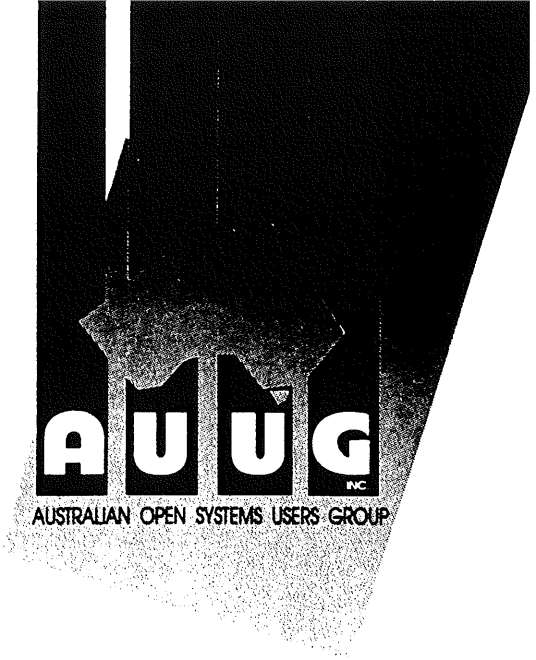
It is likely that people power will win out in the end, and Microsoft will be forced to adhere to standards, as a result of requirements from influential users such as the US Government. In the interest of open systems let's hope so.

AUUG and standards

AUUG represents both the technical interests of the UNIX programming fraternity, as well as the more commercial interests of the open systems world, where emerging standards play a more important role.

We have begun to liaise with another open systems user group, the *Australian User Alliance for Open Systems*. Many of the aims of this group, whose membership is largely drawn from Government departments, are similar to the aims of AUUG. It is hoped that the alliance between these two user organizations will enhance the cause of open systems in Australia.

Announcement



WHAT: AUUG Summer Conference Series
WHEN: February - April 1993
WHERE: In all states

For further information:
Organisers listed or
AUUG Secretariat - (02) 332-4622 tel
Liz Fraumann - Business Manager - (02) 953-3542 tel/fax

AUUG is please to announce its summer conference series for 1993. Local organisers will host a 1 to 2 day conference in each state across Australia. The conference will cater to the local market interests and may include a workshop. Parties interested should contact the conference organiser listed below.

HOBART - 11 FEB 1993
VENUE - Centenary Lecture Theatre, Univ. of Tasmania
Steven Bittinger
Computing Centre
University of Tasmania
GPO Box 252C
Hobart, TAS 7001
002 202811 tel
002 231-772 fax
Steven.Bittinger@tasman.cc.utas.edu.au

ACT -
16 Feb 1993 - workshops
VENUE - Australian National Univ.
17 Feb 1993 - conference
VENUE - National Convention Ctr.
Peter Wishart
EASAMS Australia
Level 6
60 Marcus Clark St.
Canberra, ACT 2600
06 261-2894 tel
06 261-3806 fax
pjw@pdact.pd.necisa.oz.au

DARWIN - 18 FEB 1993
VENUE - Northern Territory Univ.
Phil Maker
Dept. of Computer Science
Northern Territory University
P.O. Box 40146
Casuarina, NT 0811
089 466382
089 270612
pjm@pandanus.ntu.edu.au

SYDNEY - 19 FEB 1993
VENUE - Sydney University
Lucy Chubb
Softway Pty. Ltd.
P.O. Box 305
Strawberry Hills, NSW 2021
02 698-2322
02 699-9174
lucyc@softway.sw.oz.au

MELBOURNE - 26 FEB 1993
VENUE - Clunies Ross House Parkville
Michael Paddon
Iconix Pty. Ltd.
851 Dandenong Road
East Malvern, VIC 3145
03 571-4244 tel
03 571-5346 fax
mwp@iconix.oz.au

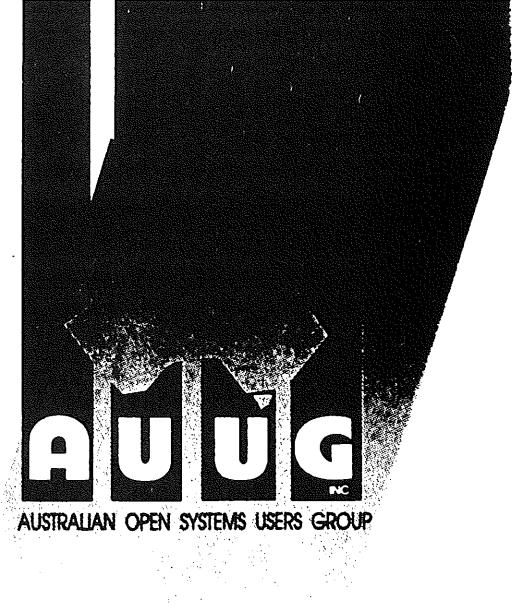
PERTH - 16 APRIL 1993
VENUE - Orchard Hotel
Adrian Booth
Adrian Booth Computer Conslt.
7 Glenrowan Pl.
Willeton, WA 6155
09 354-4936 tel
abcc@Dialix.oz.au

ADELAIDE - 25 FEB 1993
VENUE - ETSA Theatrette

Michael Wagner
Systems Services Pty. Ltd.
32 Grenfell St.
Adelaide, SA
08 212-2800 tel
08 231-0321 fax

BRISBANE- 27 APRIL 1993
VENUE - TBD
Tim Butterfield
C&T Computers
51 Looranah Rd.
Jindalee, QLD 4074
07 279-0149 tel
07 279-0249 fax

AUUG '93
Darling Harbour, Sydney, Australia, September 27-30



1993 Preliminary Announcement and Call for Papers

AUUG, Inc., forum for UNIX® Open Systems Users Presents:

"Results through Open Systems."

Over the past several years we have heard about 'What are Open Systems', and 'Maintaining Control with Open Systems'. Now it's time to hear about the results which have been achieved. Rapid expansion, the challenge of integration, global networking, and security are all issues of importance and concern to users around the world. AUUG '93 solicits papers on all aspects of UNIX and open systems, and particularly on successful applications and implementations of open systems technology to age-old and newly emerging problems.

Events:

AUUG '93 will be a four day conference, commencing September 27, 1993. The first day will be devoted to tutorial presentations, followed by three days papers, work-in-progress sessions and BOFs.

Tutorials:

Provisions for two full-day tutorials and up to eight half-day tutorials have been made. These sessions, typically in a lecture format, are targeted to educate the audience and arm them with innovative "how to" lessons. Please submit tutorial abstracts, along with preference for a half- or full-day slot to address below.

Papers:

AUUG '93 provides dual Technical and Management tracks for the presentations.

To share your innovative implementations, applications, and similar areas submit your abstract for the technical track. We are also interested in your experiences, case studies, strategic issues, and the like. If your topic better fits these areas submit your abstract for the Management track.

The above should not, of course, discourage papers which are appropriate for both audiences at once.

Vendor product announcements will be automatically rejected unless specifically submitted for the special advertising stream.

Prize for the Best Student Paper:

A cash prize of $500 will be awarded for the best paper submitted by a full-time student at an accredited tertiary education institution. In addition, the ten 'runners-up' will be rewarded with free registration.

Work-in-Progress and Advertising Sessions:

These brief 15 minute sessions are designed to report on current work with fundamental aspects highlighted. New to the AUUG conference are the Advertising sessions. These are devoted to new products only. Product specification sheets should be submitted with your abstract.

Birds-of-a-Feather Sessions (BOFs):

Are you interested in discussing particular problem areas, sharing arcane on favourite programs, using the internet, or other controversial topics? During the lunch hour and at the end of each presentation day, one hour time slots for BOFs will be available. We distinguish two types of BOF; general interest and vendor sponsored. Please contact the Programme Committee if you would like to organise a Birds-of-a-Feather Session. There may be some facilities charge to vendor sponsored events.

Speaker Incentives:

Presenters of papers are afforded free conference registration. Tutorial presenters will receive 25% of the profit for their session and a free conference registration.

Form of Submissions:

Please indicate whether your submission is relevant to the technical or management audiences, or both. In either case, submissions are required to be in the form of an abstract and an outline. Please provide sufficient detail to allow the committee to make a reasoned decision about the final paper; of course a full paper is also perfectly acceptable. A submission should be from 2-5 pages and include:

1. Author name(s), postal addresses, telephone numbers, FAX and e-mail addresses.

2. A biographical sketch not to exceed 100 words.

3. Abstract: 100 words

4. Outline: 1-4 pages giving details of the approach or algorithms pursued. Shorter outlines will not give the programme committee enough informationto judge your work fairly, and, in most cases, this means your paper will be rejected. Longer outlines and full papers simply cannot be read by the committee in the time available. However, you may append a full paper to your outline; this is sometimes useful during evaluation.

5. References to any relevant literature

6. Audio-visual requirements
   35 mm slides are preferred, however, overheads will be accepted.
   Hand written or typewriter generated overheads will not be accepted.

Acceptance:

Authors whose submissions are accepted will receive instructions on the preparation of final papers for inclusion in the conference proceedings, and the format requirements for slides.

Programme Committee:

Piers Lauder - Sydney University (Chair)
Liz Fraumann - AUUG
Ian Hoyle - BHP Research Labs
Hugh Irvine - connect.com
Rolf Jester - Digital Equipment Corporation
Bob Kummerfeld - Sydney University
Phil McCrea - Softway P/L
Andrew McRae - Megadata P/L
Greg Rose - Australian Computing and Communications Institute

Relevant Dates:

Abstract and outlines due: April 6, 1993
Notifications to authors: April 26, 1993
Final Papers due: July 26, 1993

Addresses:

Please submit one hard copy and one electronic copy (if possible) to the addresses below:

e-mail: auug93@cs.su.oz.au

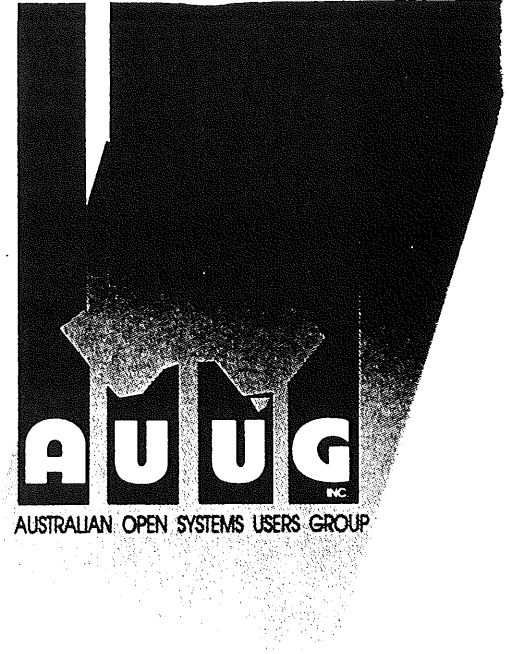
Phone: +61 2 361-5994
Fax: +61 2 332-4066

AUUG '93 Programme
P.O. Box 366
Kensington, NSW 2033

Tutorial abstracts to: ggr@acci.com.au

Please be sure to include your complete contact information (phone, fax, postal code and electronic mail addresses) in all correspondence.

AUUG realigns activities in response to membership survey



Sydney, Australia, November 17, 1992 . . . In response to a recent survey of members, AUUG Inc, the Australian UNIX and Open Systems Users Group, has initiated a number of new activities designed to meet members' stated needs, and to attract additional members to the association.

Cornerstones of AUUG's new programme include realignment of the annual conference programme to parallel the primary area of interest of the AUUG membership and closer cooperation with the Australian Computer Society (ACS).

Demographics of the association show that the primary job function of members is split almost evenly between executive/senior/MIS management and the more technical domains of systems administration and programming/manager technical services.

AUUG president, Dr Philip McCrea, underlined the value of the membership survey in providing clear direction for the association. "Our charter as a user group is to provide UNIX and open systems users throughout Australia with relevant and practical information, as well as services and education. The nature of the membership is changing in parallel with market changes, and we must respond accordingly."

AUUG '93 to reflect areas of special interest

The AUUG '93 conference programme will depart from tradition by offering three distinct themes — one for each day of the event. The main areas of special interest cited by members in the survey were communications, networking, TCP/IP, system integration/administration, and security. The AUUG '93 programme will mirror these areas of interest, with the three themes, communications (covering networking and TCP/IP), system administration and integration, and security, under the umbrella theme "Results through Open Systems."

.../more

With keen interest being expressed in the establishment of special interest groups within local (state) chapters, and a significant number of people expressing interest in leading such groups, AUUG sees these of being of great value and is working with local chapters to arrange them.

Cooperation with ACS

As 50 per cent of survey respondents have a similar affiliation with ACS, AUUG has agreed to extend discounted registration fees for the annual conference that AUUG members receive to all ACS members.

ACS national conference manager, Ms Elizabeth Bloxam, described the association with AUUG as a strategic move for both organisations. "We see great value in working with user groups, especially in a growth area such as open systems."

AUUG business manager, Ms Liz Fraumann, said that in reviewing the survey results she found some strong recurring themes: the need for peer communication and contact, the need to address business issues and the need for more technical information. "With these demands so clearly defined we can develop activities and programmes that give our members even greater value."

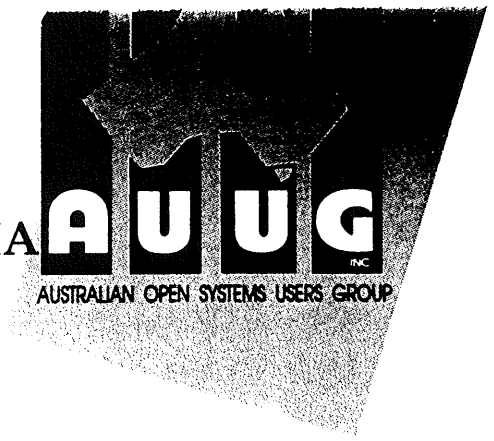
ends

AUUG, the Australian UNIX and Open Systems User Group, exists to provide UNIX and open systems users throughout Australia with relevant and practical information, services and education through co-operation among users.

Contacts:

Liz Fraumann	(AUUG)	(02) 953 3542
Lachie Hill	(Lachie Hill Consulting)	(02) 953 5629

Announcement →



AUUG & PRENTICE HALL AUSTRALIA REACH AGREEMENT

Sydney, NSW -- 23 November 1993
In a continuing effort to improve benefits for members, we are pleased to announce that Prentice Hall and AUUG have reached an agreement to provide a discount on all computer titles to members. "We are particularly please that the relationship with Prentice Hall could be extended," said AUUG Business Manager, Liz Fraumann. In the spirit of the season Elizabeth Guthrie of Prentice Hall provides the following message to all AUUG members:

Merry Christmas from
PRENTICE HALL AUSTRALIA!

Prentice Hall Australia would like to wish all AUUG members a very happy Christmas and New Year by extending our Bookclub discount offer to include all computer books published by Prentice Hall. That's right! You need never pay full retail price again!

20% discount on featured AUUG Bookclub books and 10% discount on any other Prentice Hall computer book.

To accommodate this, you will notice the Bookclub order form has changed slightly to allow members to write in additional book requests.

If ordering by phone, call Sandra Bendall on (02) 939-1333, and simply quote your AUUG membership number.

We look forward to being able to further assist you with your book requirements in 1993.

Order forms are published bi-monthly in *AUUGN*, AUUG's newsletter.

For further information:
Liz Fraumann - AUUG
(02) 953-3542 tel/fax
email: eaf@swift.sw.oz.au

Announcement

AUUG & COM TECH AUSTRALIA REACH AGREEMENT

Sydney, NSW -- 23 November 1993
One of the primary aims of AUUG is the education of its members in the areas of UNIX® and open systems. We are pleased to announce an agreement between Com Tech and AUUG to provide a training discount to AUUG individual members.

A wholly owned Australian company, established in 1987, Com Tech's training division offers authorised training courses for Novell, UNIX and SynOptics. Their highly trained staff, excellent facilities and course materials will ensure students receive maximum benefit from their investment. Com Tech provides fully equipped training centres in Sydney, Melbourne, Canberra, and Perth.

Robin Millner, Training Manager said, "We are pleased to offer a 10% discount on SCO/UNIX training to AUUG individual members." "In the new year, we will also be offering training on UNIXWare (Univel) and USL UNIX which will also be available at the discounted prices."

Course offerings include:

- Basic SCO UNIX System V/386 Administration - 2 days
- SCO UNIX System V/386 Administration - 4 days
- SCO System V Network Administration - 2 days
- Basic SCO System V Communications - 2 days
- Shell Programming for System Administrators - 2 days
- SCO Advanced Certified Engineer (ACE) - 10 days

AUUG President, Phil McCrea said, "We are excited about the opportunities unveiling themselves to our members and pleased to be in a closer association with a fine company like Com Tech."

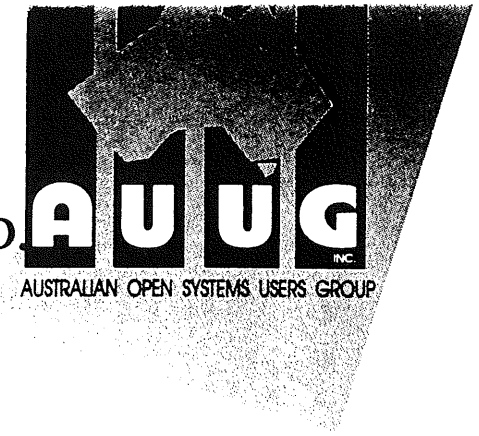
AUUG members should remember to state their affiliation and member identification number on all correspondence with Com Tech.

For further information:

Liz Fraumann - AUUG
(02) 953-3542 tel/fax
email: eaf@swift.sw.oz.au

Robin Millner or Carolyn Macken - Com Tech
(02) 317-3088 tel
(02 667-3092 fax

Announcement



AUUG & NETCOMM AUSTRALIA PTY. LTD. HELP MEMBERS GET IN TOUCH

Sydney, NSW -- 23 November 1993

Under a special arrangement NetComm Australia Pty. Ltd. will help AUUG members get in touch with the world. Distributors of leading modems, NetComm and AUUG have come to an agreement whereby AUUG members will receive a significant discount on the purchase of the famed Trailblazer with PEP and the new WorldBlazer with Turbo PEP.

Peter Morton, National Sales Manager for NetComm said, "We are pleased to provide the products to AUUG for sale to its members."

The following equipment is available:

Item #		Retail	AUUG
TR100	19.2Kbps Trailblazer with PEP	1999.00	1320.00
TR350	23Kbps, V.32bis, V42/V4bis WorldBlazer with Turbo PEP	2199.00	1610.74

It should be noted there is a limited supply of the TR100's left in stock. Members interested in this equipment should place orders soon. To order, contact the AUUG Secretariat on (02) 332-4622. Equipment will be shipped directly to members.

"We are excited about the continuing expansion of the benefits for our members," said Liz Fraumann, AUUG Business Manager, " and look forward to helping them secure the hardware which will help them stay in touch."

For further information:

Liz Fraumann - AUUG
(02) 953-3542 tel/fax
email: eaf@swift.sw.oz.au

Canberra Chapter of AUUG Inc.

4th Annual Canberra Conference and Workshops
Announcement plus Call For
Presentations and Workshops
16th/17th February 1993

AUUG in Canberra is holding its 4th annual conference and workshops on Tuesday and Wednesday the 16/17th February 1993. The current programme for the Conference follows, but we are still interested in presentations of either workshops or papers (if you are interested, contact numbers are given at the end of this article).

The conference proper is preceeded by a day of workshops. These involve a 3 hour tutorial per workshop, and are (unless otherwise specified) designed to introduce the selected topic.

Full details will be posted out in mid January including locations, times, costs, content, etc., but the preliminary programme looks like:

WORKSHOPS

To be given on Tuesday the 16th of February on the ANU campus. There may be two session (morning and afternoon) for each topic, depending upon demand. Morning and afternoon tea provided.

Network Management: *Mark Turner, Australian National University*

X windows: *Bob Dynes, Labtam*

Cruising the Internet: *Peter Elford, Australian Academic and Research Network*

Motif programming: *Jan Newmarch, University of Canberra*

SVR4: *Kevin Mayo, Sun Microsystems*

If any other topic is of interest, please contact the workshops organiser and make suggestions (but remember, we need to get presenters, titles on their own aren't a lot of use).

CONFERENCE

To be held on Wednesday the 17th of February at the National Convention Centre, Civic, Canberra. Lunch plus morning and afternoon tea provided.

Who Can that Be Now?
David Baldwin, Australian National University

This paper provides a description of a proposed implementation of an integrated identification card system including magnteic stripe, barcode and photo imaging. The system integrates enrolment data, library lending and access to buildings across various systems including several flavours of UNIX and mainframes. The system could be extended to include EFTPOS, on-line enrolments ...

XmFm - An X/MOTIF File Manager
Jan Newmarch, University of Canberra

XmFm is a file manager for Unix workstations that uses the Motif graphical user interface. Like other file managers it displays directory contents, using icons for files. The display organises files into groups

of directories, executable files (programs) and other files (data files) to aid in finding appropriate files quickly. It can display multiple directories, each appearing as an identical display without requirement of a "root" directory display (unlike the OpenLook file manager). It supports drag and drop actions using features of Motif 1.2, and can interact with other applications using the same protocol. Its major difference with other file managers are that it supports an object-oriented approach to the actions that can be performed upon files, and the high degree of user-level configurability of the system. This paper describes XmFm and compares it to other file managers.

POSIX and the Open Systems Environment
Peter Wishart, EASAMS Australia

The Open System Environment (OSE) is the international standards community view of Open Systems. POSIX and OSI are key components of OSE. POSIX.0 (the POSIX Guide) is proposing a reference model for OSE which is trying to bring together all the standards relevant to Open Systems and present them in a way that ensures consistency and completeness for users trying to implement Open Systems solutions. This paper looks at how OSE is developing within the standards community (through bodies like ISO and IEEE) and user organistions like the Commonwealth Government. It describes how POSIX, OSI and other standards fit into the POSIX proposed OSE reference model. It looks at the profiles being developed to help people navigate the standards maze.

MUDs - Serious Research Tool or Just Another Game
Lawrie Brown, University College, UNSW, Aust. Defence Force Academy

MUDs (Multi-User Dungeons) have been getting quite a bit of press recently, much of it heated, and much of it negative. This has generally run along the lines that they are simply a waste of bandwidth, and that people have better things to do with their time. Whilst there is certainly a large element of truth in this, does it mean we should write them off altogether. Well, I believe we should not, and that in fact MUDs are a very powerful tool for doing some quite interesting research. A MUD provides a controlled, user extensible environment in which a number of people can interact. The design of the programs running MUDs, the design of usage of the environments created in a MUD, and the way people interact, all involve some fascinating realms of inquiry. In this talk I intend to introduce the concepts and history of the MUDs, and then provide an overview of some uses for MUDs. These range from an on-line conference room in my own MUD, to trialing garbage collection algorithms, experimenting with economic models, simulating a Mars colony, and investigating the psychology of user interactions on MUDs. I'll try to conclude with an idea of where they are going, and some hints for responsible MUD management.

DOS access to UNIX Mail and News
Stephen Hodgman, Adept Software

There are a number of DOS packages (e.g. UUPC, FSUUCP, EZIMAIL, PCElm, SNEWS) which can provide access to UNIX news and mail from a DOS machine via dialup services. The packages provide a UUCP connection via dialup modems (or direct connection), various flavours of e-mail interfaces (including windows) and a USENET news interface. The packages are currently available for use by AUUG Canberra members wishing to connect to our UNIX machine. This presentation will look at several of the packages and share experiences on setting them up (on DOS and UNIX) and how the packages compare to standard UNIX interfaces to Mail and News.

UNIX resource management systems OR Back to the future (of batch systems)
Mathew Lim, Australian National University SuperComputer Facility

Traditionally, UNIX has always been a "free for all" system, users login and compete for as much of the available system resource as they can get. With the incursion of UNIX into the commercial (or vice versa), mainframe oriented world, batch and resource management systems for UNIX will become an

important issue. This paper describes various batch and resource management systems for UNIX. Different approches will be examined and some example products will be investigated. A case study will be presented on RASH, a system developed by the ANUSF for resource allocation and accounting on it's Fujitsu VP2200 supercomputer. While this implementation was developed for a supercomputer environment, many of the concepts used may be easily transposed into other "mainframe" UNIX environments where the fair sharing of a central resource is neccessary.

Capacity Planning
Alan Scott, Australian Technology Resources

(abstract still to be submitted)

The 1992 conference and workshops were attended by over 100 people from throughout the Canberra region. We look forward to seeing you at the 1993 conference and workshops.

For further details contact:

Presentations:	Peter Wishart ph 2612894 fax 2613806 email: pjw@lobo.canberra.edu.au
Workshops:	David Baldwin Ph 2495026 fax 2493992 email: David.Baldwin@anu.edu.au
Sponsorships/Advertising:	Elizabeth Keith Ph 2434818 fax 2434848

Summary of Annual General Meeting

On Tuesday the 10th of November, 1992, the Canberra Chapter of AUUG Inc held its AGM. 18 Registered AUUG Inc members attended, plus a few others. The new Committee is:

President	Liz Keith
Treasurer	David Baldwin
Secretary	John Barlow
Committee	Jeremy Bishop
Committee	Francois Debaecker
Committee	Merik Karman
Committee	Mathew Lim
Committee	Alan Scott

There was a general summing up of the past years activities (in a somewhat ad-hoc fashion), followed by a speedy nomination/voting session, then a summary of up and coming events (mostly the Summer Conference).

Important Dates:

19th January '93	Committee meeting at I-Block, ANU
16/17th February '93	Summer Conference and Workshops
9th March '93	General Meeting, TBA (possibly X windows)
20th April '93	General Meeting, TBA (possibly RAID technology)
11th May '93	General Meeting, TBA (possibly Windows NT)

Canberra AUUG dialup service

The Canberra Chapter operates a dialup service (a 386 machine running SCO unix, a modem and a news/email connection). Access to this service is free to chapter members (ie: AUUG members in the Canberra region). The service is not commercial (so no guarantees are given) and relies heavily upon volunteer support (many thanks to Stephen Hodgman !). Members who wish to use this service should contact the AUUG Canberra Chapter secretary (details below, I hope).

Secretary, Canberra Chapter of AUUG Inc.: Mr John Barlow,
ANU, Parallel Computing Research Facility, 2492930 (work) 2490747 (fax)
John.Barlow@anu.edu.au (email).

1993 AUUG Summer Technical Conference - Perth

April 16, 1993

The Perth 1993 AUUG Summer Conference promises to be an even bigger and better event than last year's, with two high-profile interstate speakers and, for the first time, a series of low-cost tutorials running in conjunction with the conference.

Our interstate speakers:

Greg Rose from the Australian Computing and Communications Institute will be re-presenting his paper, "A History of UNIX", fresh from presenting it at the 1993 USENIX Winter conference. Greg will also give us a report on that conference.

Chris Schoettle from UNIX System Laboratories Australia and New Zealand will be presenting a talk entitled "What's new in SVR4.2".

Each of these speakers will also be presenting a tutorial. Greg Rose will give a tutorial on public domain prototyping tools, including *perl*, Tcl, Tk, and Expect. Greg will also discuss how to obtain these tools, and describe their use in the implementation and testing of the remote terminal emulators in the TPC/C benchmark.

Chris Schoettle will be presenting a full-day tutorial, "UNIX System V Release 4 Tutorial: Technical Overview and Selected Internals Topics". This tutorial is identical to that recently presented at the 1992 AUUG Winter Conference in Melbourne.

Information about other talks and tutorials will be provided as details are finalised.

The 1993 Perth Summer conference promises to be an exciting event. Many of the quality local speakers have expressed interest in speaking again, and we should see some new faces there too.

For more information on attending, speaking, and/or presenting a tutorial, please contact the 1993 Perth Summer conference organiser:

Adrian Booth
Adrian Booth Computing Consultants
7 Glenrowan Place
Willetton WA 6155

Ph: (09) 354 4936
FAX: (09) 388 2171†
email: *abcc@dialix.oz.au*

† Thank-you to the W.A. branch of Sun Microsystems for their generosity in allowing the use of their FAX facilities to organise the Perth 1993 Summer Conference.

WAUG Meeting Reviews

[These reviews were originally published in YAUN, the newsletter of WAUG, and are reprinted here with their author's permission.]

(As usual, any errors in the summary are mine and not the speaker's).

October:
Storage Technologies
Alan Gregoire, 3M

Alan started his talk by quoting a survey that showed that if we ignore the ten very largest Australian companies, 60% of desktop platforms are PCs, and 40% dumb terminals.

This leads to the idea that there is more power and capacity in all of these PCs than in a traditional "large" system. (You can work out for yourself how much RAM and disk space 1,000 PCs collectively have).

Alan mentioned Apple's vision of this growth in desktop computing power - that RAM capacity increases four times every three years; mass storage doubles every three years; and processing power increases 70% per year.

These growth rates, if sustained, imply that by the end of the century, desktop machines will have 1Gb RAM, around 10Gb mass storage (I'm not sure of this figure - I can't read my own notes at this point), and will be capable of 1,000 MIPS.

These growth rates are being sustained by the R&D efforts of companies like 3M into "gee-whiz" technologies (I said that, by the way - I'm not quoting any marketing spiel).

An example of such technology is the "optical card". About the size of a credit card, it will hold 1Gb (by somehow achieving compression ratios of 30:1 - cynics in the audience suggested that the quoted 1Gb capacity might be a "best case" of 33.333Mb of zeroes compressed). This card costs around US$20.

Alan mentioned a new card he had been shown at a show currently happening in Perth. It stores 0.5Gb or 1Gb (I wasn't paying much attention, was I?), and costs $20 - $30. However, the drive to read it costs $14,000.

Another example of "new" technology is FRAM - ferro-electronic memory - which is basically a 1990s version of bubble memory (i.e. fast and cheap, but still non-volatile). This costs around $30/Mb.

What's wrong with traditional magnetic media? The advantages of electronic data storage include speed, the ability to run applications from RAM (and not page from a hard disk), its smaller size and greater reliability.

A very gee-whiz technology is "biological memory". A certain species of bacteria which lives in salt marshes is actually bistate - it changes its state when exposed to light of a certain frequency.

Given the size of your average bacteria, the density achievable should be around 1Gb/cubic centimetre, and prices for an *N*Gb drive should be around what we pay today for an *N*Mb drive. (Bacteria, after all, are very cheap).

There are two problems Alan mentioned - the 1ms access time (I'd love to know how they access it in the first place), and the fact that the bacteria die at temperatures above 30 degrees Celcius.

Other questions raised by the audience were "Does your storage media grow if you feed it?", and "What happens if the bacteria breed?".

Alan then briefly covered industry trends in media. Very surprisingly, Alan suggested that capacity and speed would improve, that costs would diminish, and that backups would become "more important".

One trend is the "floptical diskette", which holds 21Mb, but the drives are backward compatible with existing floppy technology. An audience member claimed that these drives were very slow, especially to format. Other audience members commented that you would expect a 21Mb disk of any type to take longer to format than a 1.44Mb floppy.

Alan then, I thought, spoilt what had been a very interesting technical talk by launching into a marketing spiel. I won't bother to report all of the rumours and innuendo he mentioned, but will let you know that helical scan technology is "inherently faulty" seeing as it has to do error correction constantly ("that's how TCP/IP works", a very astute member† †Actually, it wasn't an astute member, it was me :-) of the audience observed, and that the new QIC format cartridges (which *are* very cute) are the way to go in the future, since they are superior to DAT and 8mm technologies.

Still, the rest of the talk was excellent, and gave an interesting snapshot of the state of a very rapidly advancing area of technology.

November:

QPSX and Broadband Services

Dr. Dean Economou, QPSX Ltd.

Dean gave a rapid-fire technical talk on the status of asynchronous transfer mode (ATM) technologies today. The QPSX technology we have all heard of is a particular implementation of ATM technology.

QPSX started out in 1985 as a research contract at U.W.A., funded by Telecom Australia. This research resulted in a proposed *distributed queue dual bus* (DQDB) system. Telecom encouraged the commercialisation of this system, resulting in the formation of QPSX Ltd. in 1987. This technology has now been accepted by IEEE and made the 802.6 standard.

The key concept behind ATM is that all information is transferred in small, fixed-length *cells*. This in theory allows integrated service - for voice and data traffic to be transported over the same physical network. Dean hinted that this has not been as simple in practice as it is in theory, but did not elaborate.

As is the norm in the world today, there are *two* "standards" for ATM technology - the IEEE 802.6 standard already mentioned, which is being adopted by ISO as (I think) 8802-6; and CCITT's BISDN (Broadband ISDN) standard, which is in progress. A *third* standard is "on the way"

Dean then gave a brief description of the IEEE 802.6 DQDB (Distributed Queueing Dual Bus) technology - the "enabling technology" for metropolitan area networks. Its features include - cell-based, integrated services, broadband, robust and efficient across a wide area.

QPSX's DQDB goes further than IEEE 802.6, with multiple interconnected subnets, remote LAN bridging, circuit switching, comprehensive network management, and reliability enhancements ("self-healing"). QPSX's DQDB is deployed in 14 countries.

Services based on DQDB technology include FASTPAC in Australia, SMDS in the United States, and CBDS in Europe.

FASTPAC transports 802.6 MAC frames, much as Ethernet networks do. (The addressing scheme used - E.164 - is much like that already used for phone numbers. By 1998, phone numbers and E.164 addresses will be integrated in the same numbering scheme). FASTPAC offers "SMDS-like" service and virtual private networks.

Its two main services are FP2 - 2Mb/sec (with either a "raw" 802.6 or a LAN bridge interface), and FP10, a 10Mb/sec interface. FASTPAC is ideally suited to high-bandwidth requirements, especially bursty applications like file and image transfer, due to the essentially "pay as you use" tariff structure. One current application is the Bureau of Meteorology, which ships weather satellite images from Perth to its Cray in Sydney.

Dean concluded by pointing out that many technologies we hear about today - MANs, FASTPAC, DQDB, SMDS, and BISDN - are all implementations of ATM technology. Broadband service is available *now* via FASTPAC. The ATM infrastructure being put in place will serve us into the future. Finally, remember that all of this technology was designed here in Australia!

Adrian Booth, Adrian Booth Computing Consultants, <abcc@DIALix.oz.au>, (09) 354 4936

The ABCs of UNIX †

By Duane Bailey and John Hagerman

A is for Awk, which runs like a snail, and
B is for Biff, which reads all your mail.

C is for CC, as hackers recall, while
D is for DD, the command that does all.

E is for Emacs, which rebinds your keys, and F is for Fsck, which rebuilds your trees.

G is for Grep, a clever detective, while
H is for Halt, which may seem defective.

I is for Indent, which rarely amuses, and
J is for Join, which nobody uses.

K is for Kill, which makes you the boss, while
L is for Lex, which is missing from DOS.

M is for More, from which Less was begot, and
N is for Nice, which it really is not.

O is for Od, which prints out things nice, while
P is for Passwd, which reads in strings twice.

Q is for Quota, a Berkeley-type fable, and
R is for Ranlib, for sorting ar [sic] table.

S is for Spell, which attempts to belittle, while
T is for True, which does very little.

U is for Uniq, which is used after Sort, and
V is for Vi, which is hard to abort.

W is for Whoami, which tells you your name while
X is, well, X, of dubious fame.

Y is for Yes, which makes an impression, and Z is for Zcat, which handles compression.

† Reprinted from ;login Volume 17, Number 5

SESSPOOLE

SESSPOOLE is the South Eastern Suburbs Society for Programmers Or Other Local Enthusiasts. That's the South Eastern Suburbs of Melbourne, by the way.

SESSPOOLE is a group of programmers and friends who meet every six weeks or so for the purpose of discussing UNIX and open systems, drinking wines and ales (or fruit juices if alcohol is not their thing), and generally relaxing and socialising over dinner.

Anyone who subscribes to the aims of SESSPOOLE is welcome to attend SESSPOOLE meetings, even if they don't live or work in South Eastern Suburbs. The aims of SESSPOOLE are:

> To promote knowledge and understanding of Open System; and to promote knowledge and understanding of Open Bottles.

SESSPOOLE is also the first Chapter of the AUUG to be formed, and its members were involved in the staging of the AUUG Summer '90, '91 and '92 Melbourne Meetings.

SESSPOOLE meetings are held in the Bistro of the Oakleigh Hotel, 1555 Dandenong Road, Oakleigh, starting at 6:30pm. Dates for the next few meetings are:

Tuesday, 2 February 1993
Wednesday, 17 March 1993
Thursday, 29 April 1993
Tuesday, 8 June 1993
Wednesday, 21 July 1993

Hope we'll see you there!

To find out more about SESSPOOLE and SESSPOOLE activities, contact either **Stephen Prince** (ph. (03) 608-0911, e-mail: *sp@clcs.com.au*) or **John Carey** (ph. (03) 587-1444, e-mail: *john@labtam.oz.au*), or look for announcements in the newsgroup **aus.auug**.

Open System Publications

As a service to members, AUUG will source Open System Publications from around the world. This includes various proceeding and other publications from such organisations as

AUUG, UniForum, USENIX, EurOpen, Sinix, *etc.*

For example:

EurOpen Proceedings		USENIX Proceedings	
Dublin	Autumn'83	C++ Conference	Apr'91
Munich	Spring'90	UNIX and Supercomputers Workshop	Sept'88
Trosmo	Spring'90	Graphics Workshop IV	Oct'87

AUUG will provide these publications at cost (including freight), but with no handling charge. Delivery times will depend on method of freight which is at the discretion of AUUG and will be based on both freight times and cost.

To take advantage of this offer send, in writing, to the AUUG Secretariat, a list of the publications, making sure that you specify the organisation, an indication of the priority and the delivery address as well as the billing address (if different).

AUUG Inc.
Open System Publication Order
PO Box 366
Kensington, NSW, 2033
AUSTRALIA
Fax: (02) 332 4066

Following is a list of prices† provided by UniForum.

PUBLICATION ORDERS	Price		Postage/Handling		
	Member	Non-Member	Domestic	Canada	Overseas
CommUNIXations back issues*	$3.95	$5.00	$3	$5	$5
UniForum Monthly back issues*	3.95	5.00	3	5	5
UniNews Newsletter subscription	30.00	60.00	8	11	30
1992 UniForum Products Directory	45.00	95.00	7	15	55
1992 UniForum Proceedings	20.00	25.00	4	5	11
Your Guide to POSIX	5.00	10.00	3	4	9
POSIX Explored: System Interface	5.00	10.00	3	4	9
Network Substrata	5.00	10.00	2	3	6
Network Applications	5.00	10.00	2	3	6
The UniForum Guide To Graphical User Interfaces	4.95	9.95	2	3	6
Electronic Mail De-Mystified	5.00	10.00	3	4	9
The UniForum Guide To Distributed Computing(*)	4.95	9.95	2	3	6

† Prices in US dollars
(*) please specify issues

ACSnet Survey

1.1 Introduction

ACSnet is a computer network linking many UNIX hosts in Australia. It provides connections over various media and is linked to AARNet, Internet, USENET, CSnet and many other overseas networks. Until the formation of AARNet it was the only such network available in Australia, and is still the only network of its type available to commercial sites within Australia. The software used for these connections is usually either SUN III or SUN IV (or MHSnet). For the purposes of this survey other software such as UUCP or SLIP is also relevant.

At the AUUG Annual General Meeting held in Melbourne on September 27th, 1990, the members requested that the AUUG Executive investigate ways of making connection to ACSnet easier, especially for sites currently without connections. This survey is aimed at clearly defining what is available and what is needed.

Replies are invited both from sites requiring connections and sites that are willing to accept connections from new sites. Any other site that has relevant information is also welcome to reply (*e.g.* a site looking at reducing its *distance* from the *backbone*).

Please send replies to:

Mail: Attn: Network Survey
AUUG Inc
P.O. Box 366
Kensington N.S.W. 2033

FAX: (02) 332 4066
E-Mail: auug@atom.lhrl.oz

Technical enquiries to:

Michael Paddon (mwp@iconix.oz.au) (03) 571 4244
or
Frank Crawford (frank@atom.lhrl.oz) (02) 717 9404

Thank you

======

1.2 Contact Details

Name: _____
Address: _____
_____
_____
Phone: _____
Fax: _____
E-Mail: _____

1.3 Site Details

Host Name: _____
Hardware Type: _____
Operating System Version: _____
Location: _____
_____
_____

New Connections

If you require a network connection please complete the following section.

Please circle your choice (circle more than one if appropriate).

A1.	Do you currently have networking software?	Yes	No	
A2.	If **no**, do you require assistance in selecting a package?	Yes	No	
A3.	Are you willing to pay for networking software? If **yes**, approximately how much?	Yes	No	_____
A4.	Do you require assistance in setting up your network software?	Yes	No	
A5.	Type of software:	SUNIII TCP/IP Other (Please specify): _____	MHSnet SLIP	UUCP
A6.	Type of connection:	Direct X.25/Dialin Other (Please specify): _____	Modem/Dialin X.25/Dialout	Modem/Dialout
A7.	If **modem**, connection type:	V21 (300 baud) V22bis (2400) Other (Please specify): _____	V23 (1200/75) V32 (9600)	V22 (1200) Trailblazer
A8.	Estimated traffic volume (in KB/day): (not counting netnews)	< 1 > 100: estimated volume: _____	1-10	10-100
A9.	Do you require a news feed?	Yes Limited (Please specify): _____	No	
A10.	Any time restrictions on connection?	Please specify: _____		
A11.	If the connection requires STD charges (or equivalent) is this acceptable?	Yes	No	
A12.	Are you willing to pay for a connection (other than Telecom charges)? If **yes**, approximately how much (please also specify units, *e.g. $X/MB* or flat fee)?	Yes	No	_____
A13.	Once connected, are you willing to provide additional connections?	Yes	No	
A14.	Additional Comments:			

Existing Sites

If you are willing to accept a new network connection please complete the following section.

Please circle your choice (circle more than one if appropriate).

B1. Type of software:
SUNIII MHSnet UUCP
TCP/IP SLIP
Other (Please specify): _____

B2. Type of connection:
Direct Modem/Dialin Modem/Dialout
X.25/Dialin X.25/Dialout
Other (Please specify): _____

B3. If **modem**, connection type:
V21 (300 baud) V23 (1200/75) V22 (1200)
V22bis (2400) V32 (9600) Trailblazer
Other (Please specify): _____

B4. Maximum traffic volume (in KB/day): (not counting netnews)
< 1 1-10 10-100
> 100: acceptable volume: _____

B5. Will you supply a news feed?
Yes No
Limited (Please specify): _____

B6. Any time restrictions on connection?
Please specify: _____

B7. If the connection requires STD charges (or equivalent) is this acceptable?
Yes No

B8. Do you charge for connection?
If yes, approximately how much (please also specify units, *e.g. $X/MB* or flat fee)?
Yes No
_____

B9. Any other restrictions (*e.g.* educational connections only).?

B10. Additional Comments:

Book Reviews

ESSENTIAL SYSTEM ADMINISTRATION

by Aeleen Frisch
O'Reilly & Associates, Inc.,
Sebastopol CA USA
ISBN 0-937175-80-3

Reviewed by
Janet Jackson
Department of Computer Science
The University of Western Australia
<janet@cs.uwa.edu.au>

Over the past few years I've managed to become, I think, a reasonably good systems administrator without reading a single book on the topic. I've been lucky enough to be able to learn systems administration mainly by example, by working at a site that already does it. However, if you're suddenly confronted with a Unix system — or a whole network of them — and told to look after it, and no mentors are handy, you need a good book to show you the way. Ms Frisch has written such a book.

The book lives up to its name: it covers the basics, plus some. It refers you to other Nutshell Handbooks for in-depth information on particular topics, such as performance tuning, NFS and NIS. It includes a rather anorexic bibliography: for topics on which O'Reilly publish a book, it mentions only that one. I guess this is to be expected.

It's an introductory book that could also serve as a reference on details once you're more experienced. It assumes you're an ordinary user who knows next-to-nothing about administering multi-user systems.

The first chapter explains what a systems administrator does, and introduces a few basic tools, such as `ps`, `find` and `wall`. The second chapter then introduces you to those parts of the guts of Unix that you'll need to know about, such as processes, devices and the filesystem.

After that there are detailed chapters on most of the usual topics: startup and shutdown, user accounts, security, automating routine tasks, managing system resources, filesystems and disks, backups, terminals and modems, printers and spooling, managing TCP/IP networks (including NFS and NIS), and accounting. There's an appendix on Bourne shell programming.

The book's coverage of networking is pretty rudimentary. It doesn't tell you about administering mail, news, domain name service, UUCP or RFS. There's little mention of network management, which, to be fair, is a topic by itself, but is essential at most sites these days.

Another thing I think is missing is kernel configuration and tuning.

Nevertheless, what it does cover, it covers well. I thought the chapters on managing system resources, filesystems and disks, and terminals and modems were particularly informative. As well as specific instructions, the book contains something you won't find in most manuals: the philosophy, the lore, the reasons behind systems administration.

If you've heard of the book, you must be wondering when I'm going to mention its most striking feature: it describes BSD, SunOS 4.x, System V, Interactive, Xenix, AIX and System V Release 4, side-by-side. For each area of administration, Ms Frisch discusses the general philosophy and the basic "how-to", then describes in reasonable detail the specifics of the various Unix flavours. For some topics, such as print spooling, the flavours are so different that she gives them completely separate descriptions.

Now, if you look after only one kind of Unix, this could be annoying, but I didn't find it at all confusing. I think it'd be a real bonus if you're moving from one flavour to another (SunOS 4.x to Solaris immediately comes to mind), or if your site runs more than one flavour, or if you're merely interested in the differences and want to increase your knowledge. A lot of information is common to all Unix flavours — especially the *whys* of administration (as opposed to the *hows*).

The book is written in a clear, unpretentious style. It's very readable and in places quite entertaining. Ms Frisch comes across as an experienced, world-wise systems administrator — the mentor you wish you had.

Book Reviews

COMPUTER ARCHITECTURE: A QUANTITATIVE APPROACH

by J.L. Hennessy and D.A. Patterson
Morgan Kaufmann Publishers, Inc.

Reviewed by
Frank Crawford
Australian Supercomputing Technology
<frank@atom.ansto.gov.au>

This is one of the most useful books I have come across in years. It gives a detailed description of how computers work, from the fabrication of the chips to the design of the instruction set, from the complexity of modern superscalar CPU's to an overview of IBM channel architecture.

This book is written by the two instigators of the current RISC (*Reduced Instruction Set Computers*) systems. Patterson and colleagues at Berkeley coined the term *RISC* in 1980 and developed some of the first RISC systems, while Hennessy and colleagues at Stanford starting around the same time published work on the related aspects of efficient pipelining and compiler-assisted scheduling, key aspects of efficient use of RISC architecture.

Although the authors are proponents of RISC systems, the book covers all aspects of computer architecture, including an in depth study of the four very different architectures: the Dec VAX, IBM 360/370, the 8086 and the *DLX* (a hybrid RISC system). It doesn't try to push any one architecture but rather highlights both good and bad areas, often giving an indication of why they occurred.

The areas covered include:

- Instruction Set Design,
- Processor Implementation Techniques (including Pipelining and Vector Processors),
- Memory-Hierarchy Design,
- Input/Output,
- Future Directions, and
- Computer Arithmetic (in an appendix written by David Goldberg).

In all cases these are related back to concrete examples and heavy emphasis is placed on performance and cost. There is considerable use of quantitative measurements to evaluate performance decisions, with a set of programs used throughout the book for these measurements. The programs used are the Gnu C compiler, TeX and Spice, however the authors go to great lengths to point out that computer architects should choose examples appropriate to the intended system. The software used, including various other simulators and measuring tools, are all available for use in conjunction with the book.

It should be obvious by now that this is intended as text book for courses on computer architecture, and as such has the obligatory exercises and references. However, they have also structured the book so that it can be used at a number of levels, *i.e.* introductory, intermediate and advanced, with recommendations about which sections to select. Further, each chapter is subdivided into various standard parts, which are:

- Introduction,
- Detail,
- *Putting It All Together*, *i.e.* a sort of summary showing how the concepts are used in a real machine,
- *Fallacies and Pitfalls*, *i.e.* which gives examples of past mistakes,
- *Concluding Remarks*, and
- *Historical Perspective and References*.

This structure means that to quickly cover concepts it is only necessary to read the *Putting It All Together* sections, with some study of the detail where appropriate. As such it is suitable for just about any one who needs knowledge of what goes on inside that *black box* (or is it a *blue box*). Even a computer salesman could read the sections on *Fallacies and Pitfalls* and the *Historical Perspective* to get a better understanding of the industry.

If I have any complaints about the technical content, it is that the section on *Input/Output* is a

bit skimpy. As the authors say:

> *Input/output has been the orphan of computer architecture. Historically neglected by CPU enthusiasts, ...*
> *While this single chapter cannot fully vindicate I/O, it may at least atone for some of the sins of the past and restore some balance.*

It certainly gives a start, but it lacks much of the detail that you find in the rest of the book. My only other compliant is about the soft cover, if you can find a hard cover version, get it, as the amount of use it will get, quickly takes its toll on a soft cover.

Despite being written as a text, the information contained in it makes it very valuable as a reference, especially to anyone at or near the forefront of technology (and who isn't, with the rate of change in the industry). Although specific information will date, the book has been written in such a way that most of it will be useful for many years. It is certainly a very valuable addition to my collection.

C++ AND C EFFICIENCY

by David Spuler
Prentice Hall
ISBN 0-13-096595-2

Reviewed by
Ian Crakanthorp
Australian Nuclear Science and Technology organisation
<ian@atom.ansto.gov.au>

As the title suggests this book is written for C and C++ programmers interested in making their programs more efficient. Efficiency being defined as improvement in program speed and memory usage. The book doesn't promise to make your programs faster than a speeding bullet, but offers a number of useful and practical methods on how to achieve greater efficiency in your code. The methods discussed cover how to write more efficient C or C++ source code, rather than delving into theoretical discussions about algorithm development. Algorithms are touched on, but the author David Spuler leaves the algorithm to the programmer, mentioning that having a fast algorithm is important. The reader is assumed to be fluent in C or C++, as no introductory discussion is given about the respective languages.

The First Chapter of the book discusses the topic of efficiency, how to go about achieving it and the trade-offs you might encounter. The author also outlines a range of related books on the subject of efficient programming by other authors for the readers benefit. The second Chapter covers methods on how to measure the amount of time and space being used by the program. A few Unix utilities are discussed such as "prof" and how to use them. If the programmer is not using a Unix platform, other methods to time code are mentioned and demonstrated. Chapter 3 covers making efficient use of data structures and algorithms without having to make fundamental changes to the program. The next two chapters relate to the C and C++ languages and specific methods that can be used with each to improve efficiency. The book goes on in Chapter 6 to cover making efficient use of the ANSI C library, and some functions that can be used instead of the standard library routines. Then Chapter 7 discusses methods for improving space-efficiency. Chapter 8 looks at efficiency from the data structure and algorithm level, and using examples, shows benefits and disadvantages of different implementations. Chapter 9 poses some small programming problems, which are then coded as efficiently as possible using the methods learned in previous chapters. The last chapter assumes the reader is an implementor of a C or C++ compiler and looks at methods the compiler can use to improve efficiency.

In conclusion, I would recommend this book to anyone interested in writing more efficient code. Or even just better or more readable code, as the author promotes good programming style throughout the book. As a "get it working" worry about cleaning it up later type programmer, this book gives simple and practical methods to write efficient code from scratch. Even a more seasoned programmer would find this book an excellent reference.

Note: AUUG Inc. and various publishers/distributors have agreed to give AUUG members discounts. For more details see the announcements in this issue. An order form for Prentice Hall Australia is provided on the next page.

The AUUG 1992 FaceSaver
A Report

Michael Paddon
mwp@iconix.oz.au

December 3, 1992

1 Introduction

The FaceSaver Project at the AUUG 1992 Winter Conference was extremely successful. The digital visages of 170 delegates were captured for posterity; each a PAL resolution 24 bit image. In total, this represents nearly one fifth of a gigabyte of raw data.

This document is intended to describe the system used to acquire this data to a reasonable level of detail. Hopefully, it will satisfy those curious enough to wonder what went on behind the scenes at the FaceSaver stand. If you have any further queries, please don't hesitate to contact me.

2 Putting It All Together

The project was sponsored by Digital Equipment Corporation, who provided:

- a DECstation 5000, running Ultrix 4.2, with 24 bit graphics
- a PAL video camera
- hardware and software to interface the camera to the host
- personnel to run the FaceSaver stand

In short, we started with a powerful Unix workstation able to accept a video signal and display the data in an X11 window in real-time.

3 The FaceSaver Application

A number of tasks needed to be addressed by a custom FaceSaver application:

- collect details about each subject
- capture an image frame from a designated X11 window

- convert the image into a standard format
- output the image and associated details to disk

This process is summarized in Figure 1.

The FaceSaver consists of 1362 lines of commented ANSI C code. It provides all of the abovementioned functionality behind a Motif-based operator interface.

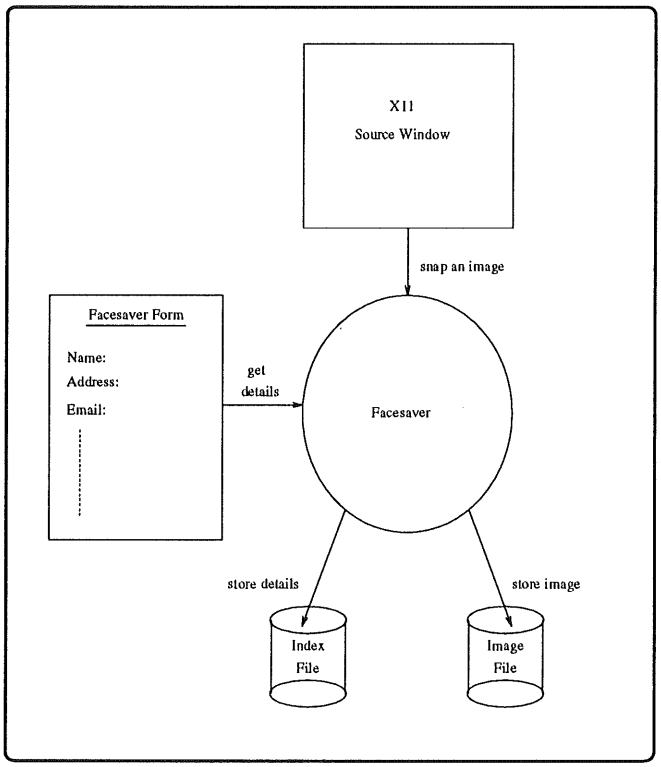


Figure 1: Face Saver Architecture

3.1 Operation

The operator runs the whole show from the form shown in Figure 2. He or she fills in each person's details, positions them in front of the camera and hits the *snap* button.

The image is sucked out of the source window and redisplayed in a confirmation dialog, as shown in Figure 3. If the subject isn't happy with his or her face (and let's be honest, who is?), the image may be rejected, and another captured.

Once accepted, the image is stored on disk in 24 bit PPM format and a record is written to an index file of the subject's details.

There are two other buttons on the operator's form: *clear* returns the form to a state of tabula rasa, and *window* allows the operator to redesignate the image source window at any time.

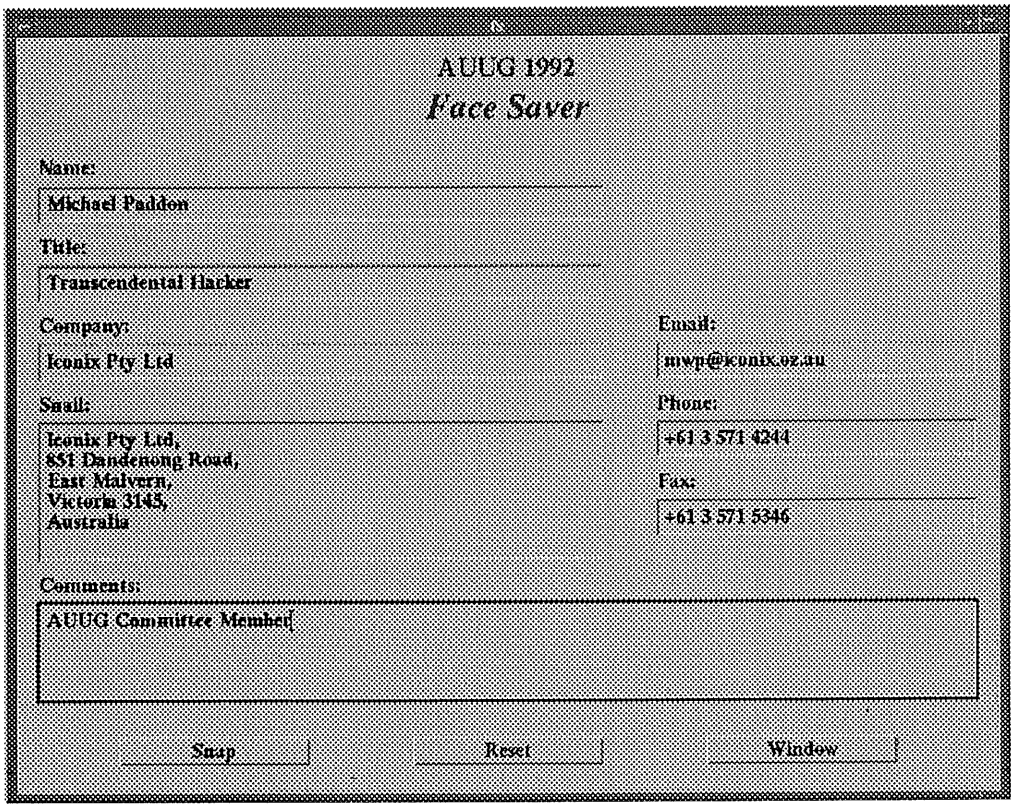


Figure 2: Operator Interface

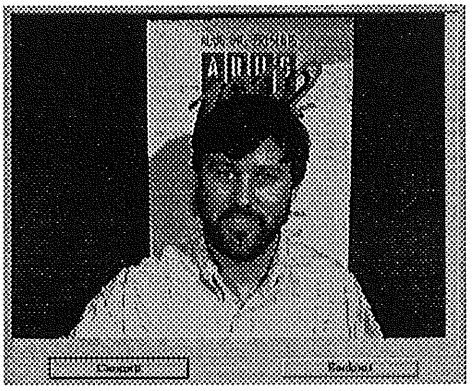


Figure 3: Commit Dialog

4 What Has Been Done With the Data?

All the faces are available for public FTP from **ftp.adelaide.edu.au**. Have a look at the file **pub/auug/faces/README** for more information.

In order to conserve disk space, the images were JPEG encoded; this reduced the size of each image from 1.2 megabytes down to 50 or 60 kilobytes. It is interesting to note that there is no visible quality degradation at this compression ratio.

Individual images have been emailed to all FaceSaver subjects who provided a valid address when their face was captured.

Lastly, a hardcopy directory of all the faces follows this paper.

5 Can I Get The Source Code?

Yes. The source is available from the abovementioned FTP site. Feel free to use or modify it (subject to the license restrictions affixed to the code).

Code to convert images to and from JPEG format is freely available via ftp from **archie.oz.au**. Grab the file **graphics/jpegsrc.v3.tar.Z**.

6 Acknowledgements

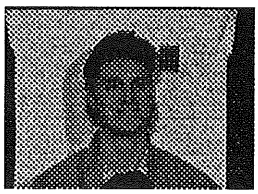
Without the generous sponsorship of Digital, the AUUG FaceSaver would not have been possible. Thanks are also due to the Digital staff who helped out on the stand (when they could have been hunting down the ubiquitous free ice cream and popcorn).

Eng Teoh coordinated the DEC end of things, making sure that hardware was in the right place at the right time.

Mark Prior helped out enormously by providing FTP space for the public archive.

Lastly I'd like to thank my employer, Iconix Pty Ltd, for allowing (and even occasionally encouraging) me to devote time and effort to AUUG.

AUUG Faces 1992

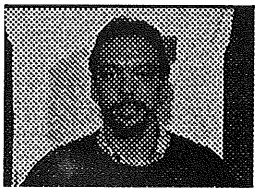


Name: Tyson Ackland
Position: Mr.
Company: Dept of Defence
Snail: Building L
Russell Offices
Canberra ACT 2600
Phone: (06) 265-5985
Fax: (06) 265-6086
Comments: Systems Administration, Networks

Name: Des Albert
Position: Senior Consultant
Company: Coles Myer Information Systems
Email: des@storesys.coles.oz.au
Snail: Level 3 Module 8
800 Toorak Road
Tooronga VIC 3146
Phone: (03) 829-6271
Fax: (03) 829-6211
Comments: Open Systems Prophet

Name: Simon Alsop
Position: Client Services Manager
Company: Praxa Ltd
Snail: 147 Eastern Rd
South Melbourne
Vic 3205
Phone: 03-9603811
Comments: Unix Support

Name: Nick Andrew
Position: Wizard
Company: Zeta Microcomputer Software
Email: nick@kralizec.zeta.org.au
Snail: PO Box 177
Riverstone NSW 2765
Comments: Networking,



Name: Sean Appleby
Position: Consultant
Company: Sean Appleby Consulting
Snail: 34 Orchid St
Enoggera QLD 4051
Phone: (07) 354-1663
Fax: (07) 855-1940
Comments: Unix, C, Progess Development,
Systems Engineering

Name: Sallyanne Astill
Position: Software Technology
Company: Mincom Pty Ltd
Email: sally@mincom.oz.au
Snail: PO Box 72
Stones Corner QLD 4120
Phone: (07) 364-9999
Fax: (07) 394-2844
Comments: Unix, R&D, Systems Programming

Name: Justin Baker
Position: Information Tech. Officer
Company: Bureau of Meteorology
Email: justinb@bom.gov.au
Snail: 150 Lonsdale St
Melbourne
Vic 3000
Phone: 03 669 4720
Fax: 03 669 4128
Comments: void

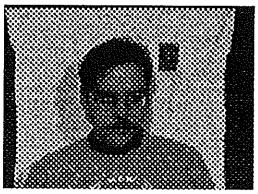
Name: David Baldwin
Position: Head, Faculties Computer Unit
Company: Australian National University
Email: David.Baldwin@anu.edu.au
Snail: GPO Box 4
Canberra ACT 2601
Phone: 06 249 5026
Fax: 06 249 3992

Name: Ramesh C. Balgovind
Position: Information Technology Officer
Company: Bureau of Meteorology
Research Centre
Email: rcb@meteorology.ho.bom.gov.au
Snail: GPO Box 1289K
Melbourne VIC 3001
Phone: (03) 669-4404
Fax: (03) 669-4660
Comments: Scientific Programming, Fortran,
Cray, SunOS

Name: Sheldon Barr
Position: System Administrator
Company: Queensland TAB
Snail: 240 Sandgate Rd
Albion QLD
Phone: (07) 862-0267
Fax: (07) 262-8505

Name: Kirk Barrett
Position: Consultant Programmer
Company: University of Wollongong, ITS
Email: kirk@uow.edu.au
Phone: (042) 21 3121
Fax: (042) 21 3262

Name: Sean Batt
Position: Open Systems Programmer :-)
Company: Australian National University
Email: sean@coombs.anu.edu.au
Snail: GPO Box 4
Canberra City
2601
Phone: (06) 249 3296
Fax: (06) 257 1893
Comments: Distributed Ray tracing



Name: Mike Benson
Position: Systems Engineer
Company: Alcatel Australia
Email: mikb@titan.alcatel.oz.au
Phone: (02) 690-5434
Fax: (02) 690-5111
Comments: Office Automation,
Network Management

Name: Greg Birnie
Position: Senior Software Design Engineer
Company: Leeds & Northrup Australia
Email: greg@lna.oz.au
Snail: 42 McKechnie Drive,
Eight Mile Plains,
Australia 4113.
Phone: 07 3402167
Fax: 07 3402100

Name: Craig Bishop
Position: Unix Systems Manager
Company: Geelong and District Water Board
Email: csb@gdwb.oz.au
Snail: 61-67 Ryrie Street
Geelong, Victoria, 3220
Phone: (052) 26 2506
Fax: (052) 21 8236

Name: Greg Black
Position: Director
Company: Greg Black & Associates
Email: gjb@bullit.void.oz.au
Snail: 681 Park Street
Brunswick
Vic. 3056
Phone: (018) 537 484
Fax: (03) 380 4716
Comments: Software development in C
and C++ for UNIX

AUUG Faces 1992

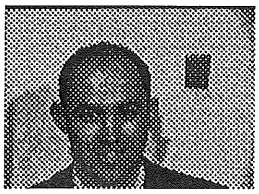
Name: Darren Bock
Position: Senior Unix Programmer
Company: Dept DEVETIR, Qld
Email: sgccdeb@citecuc.citec.oz.au
Snail: Darren Bock
Floor 1, Porbes House
Makerston St, Brisbane, 4001
Phone: (07) 227 5726
Comments: C, 4GL
Cycling, People

Name: Greg Bond
Position: Chief Pointy Head
Company: Burdett Buckeridge & Young
Email: gnd@bby.com.au
Phone: (03) 614 8922
Fax: (03) 614 8742

Name: Stephen Boucher
Company: Metal Trades Industry Association
Email: stephen@mtiame.mtia.oz.au
Snail: Level 2, 509 St Kilda Rd
Melbourne, 3004

Phone: 0328001 11
Fax: 0328001 99

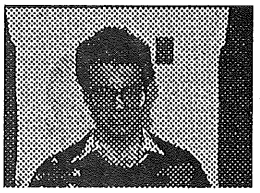
Name: Dr Lawrie Brown
Position: Lecturer
Company: Australian Defence Force Academy
Email: Lawrie.Brown@adfa.oz.au
Snail: Department of Computer Science
University College, UNSW
Canberra ACT 2600
Phone: 06 2688816
Fax: 06 2688581



Name: Robin Brown
Position: System Administrator
Company: BHP Research
Email: robin@resmel.bhp.com.au
Snail: 245 Wellington Rd
Mulgrave VIC 3170
Phone: +61 3 560 7066
Fax: +61 3 561 6709

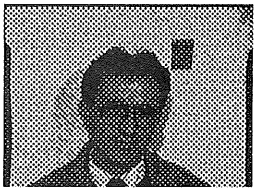
Name: David Burren
Position: Systems Manager,
Network Representation Systems
Company: Telecom Australia
Email: davidb@img.com.au
Snail: 5/459 Lt Collins St
Melbourne
VIC 3000
Phone: +61 3 634 3635
Fax: +61 3 670 1189

Name: Julian Byrne
Position: Lecturer
Company: Monash University
Email: Julian.Byrne@eng.monsah.edu.au
Snail: Electrical and Computer Systems
Engineering Dept
Monash University
Clayton VIC 3168
Phone: (03) 565-3458
Fax: (03) 565-3454
Comments: Computer vision, robot vehicles,
parallel computing

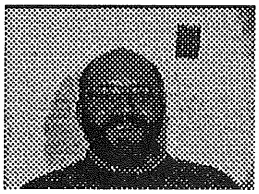


Name: John Carey
Position: Senior Systems Programmer
Company: Labtam Australia
Email: john@labtam.oz.au
Snail: 41 Malcolm Rd
Braeside VIC 3195
Phone: (03) 587-1444
Fax: (03) 580-5581
Comments: Labtam X terminal
Research and Development

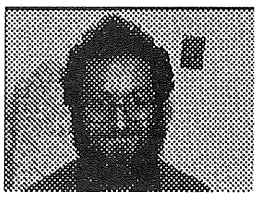
Name: Tony Cataldo
Position: Systems & Network Administrator
Company: Telecom Research Labs.
Email: t.cataldo@trl.oz.au
Snail: 770 Blackburn Rd.,
Clayton VIC 3168
PO Box 249
Clayton VIC 3168
Phone: (03) 253 6725
Fax: (03) 253 6664
Comments: Conference has been great,
waiting on my DG
umbrella and PI Chairs



Name: Sab Celik
Position: Systems Manager
Company: University of Technology
Email: sab@ccsd.uts.edu.au
Snail: PO Box 123
Broadway NSW 2007
Phone: (02) 330-2112
Fax: (02) 330-1994
Comments: Operating Systems Software

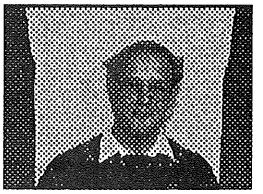


Name: Chris Chittleborough
Position: Chief Techie
Company: Syscorp
Snail: Level 4,
30 Collins St,
Melbourne 3000
Phone: 03 654 7466
Fax: 03 659 2163

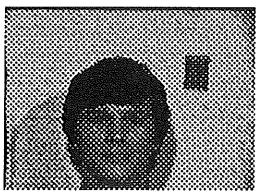


Name: Peter Chubb
Position: Senior Software Engineer
Company: Softway Pty Ltd
Email: peterc@softway.sw.oz.au
Snail: po box 305
Strawberry Hills
NSW 2005
Phone: (02) 698 2322
Fax: (02) 6999174

Name: Melinda Church
Position: Product Manager
Company: Hitachi Data Systems
Snail: 11-17 Khartoum Road
North Ryde, Nsw 2113
Phone: (02) 887-4455
Fax: (02) 887-4899



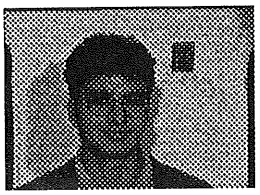
Name: Scott Colwell
Position: Senior Design Engineer
Company: Labtam Australia
Email: scott@labtam.oz.au
Snail: 41 Malcolm Rd
Braeside, VIC 3195
Phone: (03) 587-1444
Fax: (03) 580-5581



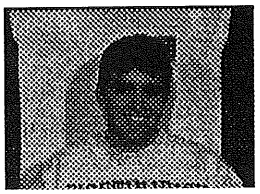
Name: Tim Cook
Position: Systems Programmer
Company: Deakin University
Email: tim@deakin.oz.au
Snail: Waum Ponds VIC
Phone: +61 52 27 2601
Fax: +61 52 ?? ????

Name: Barbara Covington
Position: Dealer Manager
Company: AIPC
Email: barb@aipc.oz.au
Snail: 331 Johnston St
Abbotsford VIC 3067
Phone: (03) 419-7988
Fax: (03) 416-0060
Comments: Office Automation, Aster*x

AUUG Faces 1992



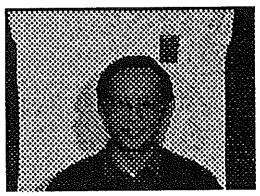
Name: Ralph Cowan
Position: Facilities Analyst
Company: Woodside Petroleum
Snail: 1 Adelaide Terrace
Perth WA 6000
Phone: (09) 224-4965
Fax: (09) 325-8178
Comments: Sun System Administration



Name: Ian Crankanthorp
Position: Systems Administrator
Company: ANSTO
Email: ian@atom.ansto.gov.au
Snail: Private Mail Bag 1
Menai NSW 2234
Phone: (02) 717-3365
Fax: (02) 717-9273
Comments: Systems Spud

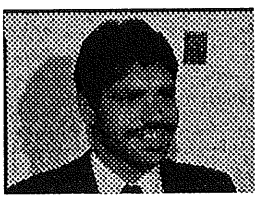
Name: Mike Crooks
Position: Senior Designer
Company: Computer Power Group
Email: mike@bohra.cpg.oz.au
Phone: (03) 823 0208
Fax: (03) 824 8068

Name: Paul Cutt
Position: VP Engineering
Company: Xtensory Inc
Email: fong@world.std.com
Snail: 140 Sunridge Drive
Scotts Valley, CA 95066 USA
Phone: 408/439-0600
Fax: 408/439-0600

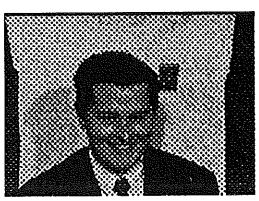


Name: Freeman Deng
Position: System Programmer
Company: Moss Products
Snail: 6/28 Wilgah Street
St Kilda East VIC 3183
Phone: (03) 551-6188
Comments: Networking, System Management

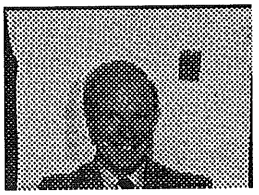
Name: Julian Dryden
Position: Researcher
Company: CSIRO
Email: julian@dwt.csiro.au
Snail: Po Box 7 Ryde 2112
Phone: 02 809 9345
Fax: 02 809 9476
Comments: I wanna better job!!!



Name: Peter Elford
Position: Network Coordinator: Technical
Company: AARNet
Email: pte@aarnet.edu.au
Snail: GPO Box 1142
Canberra ACT 2601
Phone: 06 249 3542
Fax: 06 249 1369

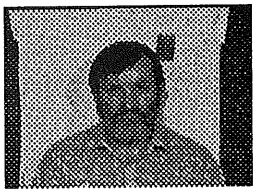


Name: Simon Flowers
Position: Development Officer
Company: Mobil
Snail: 2 City Road
South Melbourne VIC 3205
Phone: (03) 252-3365
Fax: (03) 252-3669

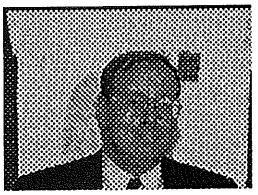


Name: Ricky Cox
Position: CADD Development Manager
Company: Queensland Transport
Snail: GPO Box 1412
Brisbane QLD 4001
Phone: (07) 834-2285
Fax: (07) 834-2998
Comments: SunOS, Unix, C, CADD, Fortran

Name: Frank Crawford
Position: System Support Manager
Company: Aust Supercomputing Technology
Email: frank@atom.ansto.gov.au
Snail: Woods Centre
Private Mail Bag 1
Menai NSW 2234
Phone: (02) 717-9404
Fax: (02) 717-9429
Comments: Systems Admin, Super Computing

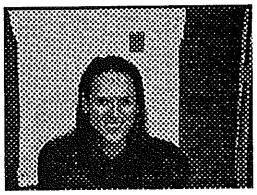


Name: Geoff Crossley
Position: Mr
Company: Attorney-General's Dept
Snail: PO Box 246
Belconnen ACT 2616
Phone: (06) 264-4441
Fax: (06) 264-2721
Comments: Systems Planning



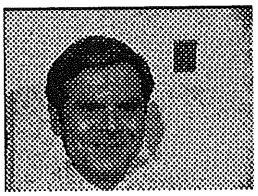
Name: Francois Debaecker
Position: Information Systems Manager
Company: Aust. Institute of Criminology
Email: fd@aic.act.crime.oz.a
Snail: GPO Box 2944
Canberra ACT 2601
Phone: (06) 274-0234
Fax: (06) 274-0201
Comments: Unix, Unify/ACCELL, SPSS-X

Name: Ian Donaldson
Position: System Programmer
Company: Labtam Australia
Email: iand@labtam.oz.au
Snail: 41 Malcolm Rd
Braeside VIC 3195
Phone: (03) 587-1444
Fax: (03) 580-5581



Name: Ruth Earwaker
Position: Senior Consultant
Company: Andersen Consulting
Snail: 141 Walker St
North Sydney NSW 2060
Phone: (02) 964-6900
Fax: (02) 922-2065
Comments: Championing UNIX Causes!!, AI

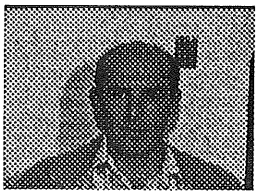
Name: Michael Flower
Position: Project Leader
Company: Telecom Research Labs
Email: mcf@munnari.oz.au
Snail: 770 Blackburn Rd
Clayton
Phone: +61 3 253-6179
Fax: +61 3 253-6173



Name: Wael Foda
Position: Managing Director
Company: ACMS
Email: ???
Snail: PO 468
Paddington NSW 2021
Phone: +61 2 332 4622
Fax: +61 2 332 4066

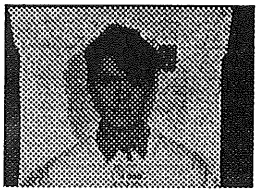
AUUG Faces 1992

Name: Liz Fraumann
Position: Business Manager
Company: AUUG Inc
Email: eaf@sw.oz.au
Snail: PO Box 366
Kensington NSW 2033
Phone: +61 2 361 5994
Fax: +61 2 332 4066



Name: Richard Fullford
Position: Staff Officer Systems
Company: Material Division - Army
Snail: J-1-08 Russell Offices
Canberra ACT 2600
Phone: (06) 265 5787
Fax: (06) 265 5410

Name: Miles Gillham
Position: Software Engineering Manager
Company: TRAC Systems Australia
Email: mhg@ptcburp.ptcbu.oz.au
Snail: PO BOX 83
NERANG QLD 4211
Phone: (075) 784 111
Fax: (075) 964 592
Comments: EFTPOS, End of Line Equipment, Smart cards, MMC units

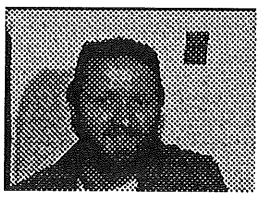


Name: Rupert G. Goldie
Position: Research Scientist
Company: Australian AI Institute
Email: ash@aaii.oz.au
Snail: 1 Grattan St
Carlton VIC 3053
Phone: (03) 663-7922
Fax: (03) 663-7937
Comments: System Admin

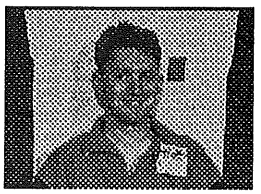
Name: Stan Gorr
Position: Director
Company: Vivstan
Snail: 105 Vanessa St
Kingsgrove NSW 2208
Phone: (02) 502-2477
Fax: (02) 50-4603
Comments: Process Printing

Name: Chris Green
Position: Unix Systems Manager
Company: Biomolecular Research Institute
Email: chrisg@tigger.mel.dbe.csiro.au
Snail: 343 Royal Pde
Parkville VIC 3052
Phone: +61 3 342 4300
Fax: +61 3 342 4301

Name: Robert Groom
Position: Senior Support Engineer
Company: Labtam Australia
Email: robert@labtam.oz.au
Snail: 41 Malcolm Rd
Braeside VIC 3195
Phone: (03) 587-1444
Fax: (03) 580-5581

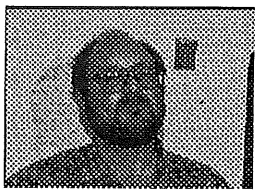


Name: Alan Hargreaves
Position: System Manager
Company: University of Newcastle
Email: alan@frey.newcastle.edu.au
Snail: Computer Systems Group
University of Newcastle
University Drive
Callaghan, 2308
Phone: 049 215512
Fax: 049 684742

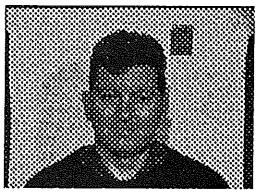


Name: Rolf Freeman
Position: Trainee Engineer
Company: B.H.P. Westernport
Snail: 14 St. Ives Gve
Mt. Martha
3934
Phone: (059) 796911
Comments: Student Eng at Monash Caulfield
No Modem Yet!!!

Name: Dennis Garnsey
Position: Principal Engineer
Company: Telecom Australia
Snail: P.O. Box A138
Sydney South
2000
Phone: 02 287 1150
Fax: 02 261 8657
Comments: Alarm Management



Name: Peter Goddard
Position: HOD, Computing & Inf Sci.
Company: LaTrobe Univ College Northern Vic.
Email: goddard@redgum.ucnv.edu.au
Snail: Box 199
Bendigo
VIC 3550
Phone: (054) 447426
Fax: (054)447777
Comments: Education (tertiary), computer science.



Name: Andrew Gollan
Position: Senior Consultant
Company: Softway Pty. Ltd.
Email: adjg@sw.oz.au
Snail: Box 305
Strawberry Hills
NSW 2012
Phone: +61 2 698 2322
Fax: +61 2 699 9174

Name: Peter D. Gray
Position: Professional Officer
Company: University of Wollongong, Computer Science
Email: pdg@cs.uow.edu.au
Phone: (042) 21 3770
Fax: (042) 21 3262
Comments: Death to DOS!

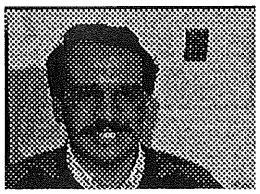
Name: Joanne Gregory
Position: Application Support Specialist
Company: AIPC
Email: joanne@aipc.oz.au
Snail: 331 Johnston Street
Abbotsford VIC 3067
Phone: (03) 419-7988
Fax: (03) 416-0060
Comments: Office Automation, Aster*x

Name: Ty Ha
Position: Computer Systems Officer
Company: Land Titles Office
Snail: Queens Square
Sydney NSW 2000
Phone: (02) 228-6968
Comments: DataBase Administration, Application Programming

Name: Ken Harmsworth
Position: Mgr Open Systems Group
Company: Coles Myer
Snail: 800 Toorak Rd
Tooronga Vic
3146
Phone: 829 6515
Fax: 829 6211

AUUG Faces 1992

Name: Kylie Hawkins
Position: Computer Systems Officer
Company: University of Newcastle
Email: kylie@cs.newcastle.edu.au
Snail: Computer Science Department
The University of Newcastle
University Drive
Callaghan 2308
Phone: 049 21 6175
Fax: 049 21 6929
Comments: Educational software and applications



Name: Michael Homsey
Position: R&D Contractor
Company: OTC Australia
Email: msh@otc.otca.oz.au
Snail: GPO Box 7000
Sydney 2001
Phone: (02) 287-3225
Fax: (02) 287-3299

Name: Colin Hoskins
Position: Senior Programmer Analyst
Company: BHP IT
Email: colin@bhpese.oz.au
Phone: (049) 40 2101
Fax: (049) 40 2165

Name: Tim Hudson
Position: Unix System Specialist
Company: Mincom
Email: tjh@mincom.oz.au
Snail: PO Box 72
Stones Corner QLD 4120
Phone: (07) 364-9999
Fax: (07) 394-2844

Name: Janet Jackson
Position: Systems Administrator
Company: University of Western Australia
Email: janet@cs.uwa.edu.au
Snail: NEDLANDS WA 6009
Phone: (09) 380 2231
Fax: (09) 380 1089
Comments: I am a systems admin in the Department of Computer Science. I am also on the committee of WAUG and editor of its newsletter YAUN.

Name: Graham Jenkins
Position: Clients Support Specialist
Company: CBIS
Email: gkj@main.com.au
Snail: Level 3
71 Queens Rd
Melbourne VIC 3004
Phone: (03) 526-3700
Fax: (03) 526-3799
Comments: RDBMS, Communications, 4GL

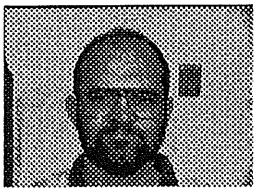
Name: Rolf Jester
Position: Open Systems Mktg Manager
Company: Digital Equipment Australia
Email: jester@sno.mts.dec.com
Snail: 410 Concord Rd
Rhodes NSW 2138
Phone: +61 2 561 5168
Fax: +61 2 561 5850

Name: Andre Joanisse
Position: Computer System Engineer
Company: BHP Research - Newcastle
Email: andrej@resntl.bhp.com.au
Snail: PO Box 188
Wallsend, NSW 2287
Phone: (049)510 572
Fax: (049)513 740

Name: Andrew Hodgson
Position: Computer Manager
Company: Australian AI Institute
Email: ash@aaii.oz.au
Snail: 1 Grattan St
Carlton VIC 3053
Phone: (03) 663-7922
Fax: (03) 663-7937
Comments: System Admin

Name: Ron Hoskin
Position: Open Systems Specialist
Company: Systems Services Pty Ltd
Snail: 32 Grenfell St
Adelaide SA 5000
Phone: (08) 212-2800
Fax: (08) 231-0321
Comments: Unix and PC Operating Systems Support

Name: Ian Hoyle
Position: Senior Research Scientist
Company: BHP Research
Email: ianh@bhp.com.au
Phone: (03) 560 7066
Fax: (03) 561 6709
Comments: Chief good guy



Name: Glenn Huxtable
Position: Computer System Manager
Company: University of Western Australia
Email: glenn@cs.uwa.edu.au
Snail: Department of Computer Science
University of Western Australia
Nedlands, 6009
Phone: (09) 380 2878
Fax: (09) 380 1089
Comments: AUUG committee member, WAUG chairman

Name: Loo Jansen
Position: Manager, Library Systems
Company: University of Newcastle
Email: ccalj@cc.newcastle.edu.au
Snail: University Drive
Callaghan NSW 2309
Phone: (049) 21-5842
Fax: (049) 21-5833
Comments: Programming, X-ternminals

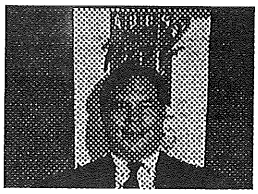
Name: Murray Jensen
Position: Computer Scientist
Company: CSIRO
Div of Manufacturing Technology
Email: mjj@mlb.dmt.csiro.au
Snail: Locked Bag No. 9
Preston, Vic, 3072
Phone: +61 3 487 9263
Fax: +61 3 484 0878
Comments: Hi

Name: Haritar Jha
Position: Senior Info-Tech Officer
Company: Bureau of Meteorology
Email: hatrj@??
Snail: 150 Lonsdale St
Level 2
Melbourne VIC 3000
Phone: +61 3 669 4029
Fax: +61 3 669 4500

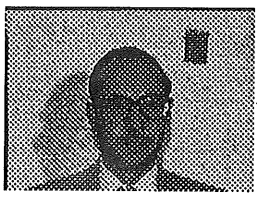
Name: Ian C. Johnston
Position: UNIX System Administrator
Company: Colonial Mutual
Email: ijohnston@cmutual.com.au
Snail: 330 Collins St
Melbourne VIC 3000
Phone: (03) 607-6448
Fax: (03) 607-6198
Comments: Networking, Admin,

AUUG Faces 1992

Name: Julie Jones
Position: President
Company: Uniforum NZ
Email: julie@jj.co.nz
Snail: PO Box 27149
Mt Roskill
Auckland
NEW ZEALAND
Phone: +64 25 958 245
Fax: +64 9 629 2015

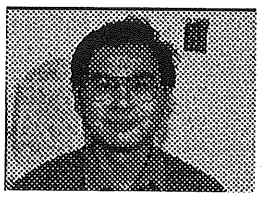


Name: Peter Karr
Company: Computer Magazine Publications
Phone: (02) 901 3622
Fax: (02) 901 3863

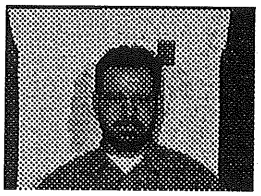


Name: Michael Keen
Position: Manager Projects Office Systems
Company: Dept of Premier and Cabinet
Snail: Corporate Services
Dept of the Premier and Cabinet
G.P.O. Box 2343
Adelaide SA 5001
Phone: 08 2263604
Fax: 08 231 0724
Comments: Integration of Unix using PC's.
Information sharing amongst
multivendor systems to assist
business decisions.

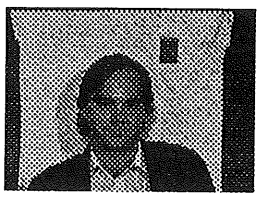
Name: David Kimpton
Position: Systems Programmer
Company: Vic Roads
Snail: 560 Lygon St
Carlton 3053
8thFloor South Building
Phone: 345-4536



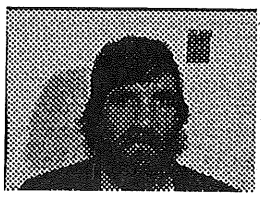
Name: Mathew LIM
Position: Systems Programmer
Company: ANU Supercomputer Facility
Email: M.Lim@anu.edu.au
Snail: GPO Box 4,
Canberra City,
ACT 2601
Phone: 06 249 2750
Fax: 06 247 3425



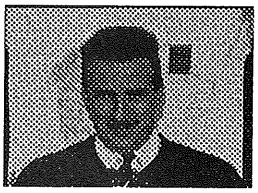
Name: Steve Landers
Position: Director
Company: Functional Software
Email: scl@fs.com.au
Snail: Suite 6 / 173 High Rd
PO Box 266
Willetton WA 6155
Phone: (09) 246-3331
Fax: (09) 307-7442
Comments: Unix System Management Software



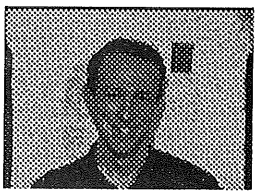
Name: Peter N. Lewis
Position: Computer Systems Officer
Company: NCRPDA, Curtin University
Email: peter@cujo.curtin.edu.au
Snail: GPO Box U1987
Perth WA 6001
Phone: (09) 368-2055
Fax: (09) 367-8141
Comments: Macintosh, Pascal



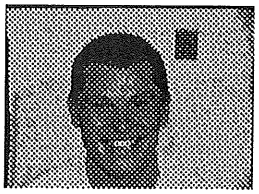
Name: Michael Lightfoot
Position: Director
Company: SysIX
Email: michal@sysix.oz.au
Snail: P.O. Box 155
Charnwood
ACT
2615
Phone: 06 258 8185
Fax: 06 258 8185
Comments: Email ineperative til November



Name: Merik Karman
Position: Technical Support Manager
Company: Defence Service Homes
Email: merik@blackadder.dsh.oz.au
Snail: PO Box 21
Woden ACT 2606
Phone: (06) 289-6790
Fax: (06) 285-2608
Comments: Graphics, Oracle RDBMS,



Name: Christopher Keane
Position: Unix Systems Manager
Company: State Bank NSW
Email: chris@rufus.state.com.au
Snail: Level 40
225 George Street
Sydney NSW 2000
Phone: (02) 259-4459
Fax: (02) 251-8009
Comments: Highly available systems,
Disaster recovery,
Systems administration



Name: Brad Keifer
Position: Analyst Programmer
Company: BHP IT
Email: brad@bhpese.oz.au
Phone: (049) 40 2101
Fax: (049) 40 2165

Name: Patrick Ko
Position: Manager
Company: LanTech
Email: patrick@marsh.cs.curtin.edu.au
Snail: 77 Belmont Ave
Belmont WA 6104
Phone: +61 9 478 3388
Fax: +61 9 277 1382

Name: Geoff Lamb
Position: Computer Scientist
Company: CSIRO,
Div of Manufacturing Technology
Email: gsl@mlb.dmt.csiro.au
Snail: Locked Bag No. 9
Preston, Victoria
3072
Phone: (03) 487 9295
Fax: (03) 484 0878

Name: Sue Landers
Position: Administrator
Company: Functional Software
Email: sgl@fs.com.au
Snail: Suite 6 / 173 High Rd
PO Box 266
Willetton WA 6155
Phone: (09) 246-3331
Fax: (09) 307-7442
Comments: Unix System Management Software

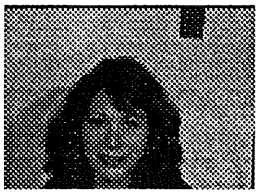
Name: Hok Min Lie
Position: Software Engineer
Company: BHP Research
Email: hok@resntl.bhp.com.au
Snail: PO Box 188
Wallsend NSW 2287
Phone: (049) 51-2444
Fax: (049) 51-3740
Comments: Expert System, Data Base, GUI

Name: John Lions
Position: Associate Professor
Company: University of New South Wales
Email: johnl@cs.unsw.oz.au
Snail: Kensington, Nsw 2033
Phone: (02)697-4071
Comments: Life member

AUUG Faces 1992

Name: Richard Liu
Position: Senior Scientist
Company: Telecom Research Lab
Email: r.liu@trl.oz.au
Snail: 770 Blackburn Rd
Clayton VIC 3160
Phone: (03) 253-6336
Fax: (03) 253-6664
Comments: Systems Administration, SUNos, Ultrix

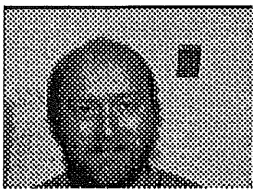
Name: Rob Logie
Position: Network Operations and Support
Company: Telecom Australia, Computer Support and Services
Email: logier@cheops.qld.tne.oz.au
Phone: (07) 837 5174
Fax: (07) 837 4704



Name: Mary Lopes
Position: Technical Services
Company: EDS (Australia)
Snail: P.O. Box 223
Mulgrave North, Vic. 3068
Phone: (03) 262-2651
Fax: (03)262-2611

Name: John Lowry
Position: Unix Support Manager
Company: Dept. Treasury
Snail: Parkes Place
Parkes ACT 2600
Phone: (06) 263-2996
Fax: (06) 263-3104
Comments: Unix System Admin, Networking

Name: Steven Lynch
Position: Unix and Hardware Hacker
Company: Iconix Pty Ltd
Email: sl@iconix.oz.au
Snail: 851 Dandenong Road
East Malvern
Victoria, 3145
Phone: (03) 571 4244
Fax: (03) 571 5346

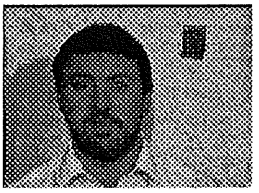


Name: Chris Maltby
Position: Technical Director
Company: Softway Pty Limited
Email: chris@sw.oz.au
Snail: PO Box 305
Strawberry Hills NSW 2012
Phone: +61 2 698 2322
Fax: +61 2 699 9174

Name: Brian Marriott
Position: Technical Services Manager
Company: Department of Computer Science, University of Tasmania
Email: B.W.Marriott@cs.utas.edu.au
Snail: GPO Box 252C
Hobart Tas 7001
Phone: 002 202929
Fax: 002 202913

Name: Mike McCauley
Position: Consultant
Company: TechNIX
Email: mike@technix.oz
Snail: 197 Riversdale Rd
Hawthorn, VIC
3122
Phone: (03)8193339
Fax: (03)8193278
Comments: X, Motif, GUI, UIMS, Unix, C++

Name: Glenn McNally
Position: Analyst/Programmer
Company: BHP IT
Email: glenn@bhpese.oz.au
Phone: (049) 402 163
Comments: Unix/C programmer

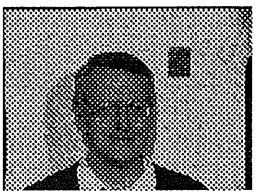


Name: Andrew McRae
Position: Software Engineer
Company: Megadata Pty Ltd.
Email: andrew@megadata.mega.oz.au
Snail: 2/37 Waterloo Rd
North Ryde 2113
NSW AUSTRALIA
Phone: +61 2 805 0899
Fax: +61 2 887 4847

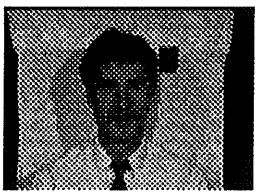
Name: Scott Merrilees
Position: Systems Programmer
Company: BHP IT
Email: Sm@bhpese.oz.au
Snail: PO BOX 216
Hamilton
2303
Phone: +61 49 402132
Fax: +61 49 402165

Name: Peter Miller
Position: S.I.T.O C
Company: B.M.R.
Email: pmiller@bmr.gov.au
Snail: GPO Box 378
Canberra ACT 2601
Phone: 06 2499656
Fax: 06 2499977
Comments: Graphics
Image Processing

Name: Neil Mitchell
Position: SO1 Information Systems Operations
Company: Headquarters Logistic Command - Army
Snail: Systems Branch (T-18)
350 St Kilda Rd
Melbourne VIC 3004
Phone: (03) 282-6660
Fax: (03) 282-6265
Comments: Unix, Networks, Data Bases, Application Development, Platform Managing



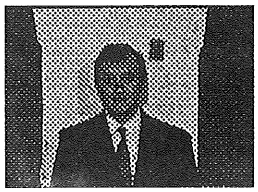
Name: Ian Moran
Position: Communications Manager
Company: Telecom Australia
Email: i.moran@trl.oz.au
Phone: (03) 541 6194
Fax: (03) 543 4127



Name: Christopher Mugdan
Position: Mr
Company: Musys
Email: chrism@runxtsa.runx.oz.au
Snail: 7 Innes Ave
Hornsby NSW 2077
Phone: (02) 477-7556
Comments: Unix Development, Networks

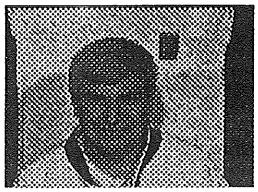
Name: Neil Murray
Position: Analyst / Programmer
Company: Webster Computer Corporation
Email: neil@wcc.oz.au
Snail: 1270 Ferntree Gully Road
Scoresby, Victoria, 3179
Phone: +61 3 764 1100
Fax: +61 3 764 1179
Comments: Embedded Systems, Communications, Amateur Radio

AUUG Faces 1992

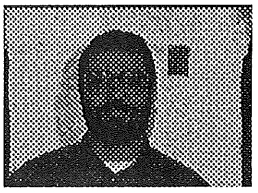


Name: Dave Newton
Position: Senior Consultant
Company: Andersen Consulting
Email: auugn@munari.oz.au
Snail: 141 Walker St
North Sydney NSW 2060
Phone: (02) 964-6900
Fax: (02) 922-2065
Comments: Championing UNIX Causes!!

Name: Hanh Nguyen
Company: Burdett, Buckeridge & Young
Email: shn@bby.com.au
Snail: 2nd Floor, 405 Collins St
Melbourne VIC 3000
Phone: +61 3 614 8922
Fax: +61 3 614 8742



Name: John O'Brien
Position: Managing Director
Company: Whitesmiths, Australia P/L
Email: john@wsa.oz.au
Snail: Suite 5, Woods Centre
ANSTO Business & Technology Park
Lucas Heights Research Labs
New Illawarra Road
Lucas Heights NSW 2234
Phone: +61-2-717-9444
Fax: +61-2-717-9445
Comments: Rugby, Sailing,
Sex, Macintosh UNIX etc



Name: Ernst Van Oeveren
Position: Information Systems Manager
Company: Menzies School of Health Research
Email: ernst@menzies.su.edu.au
Phone: (089) 22 8684
Fax: (089) 27 5187

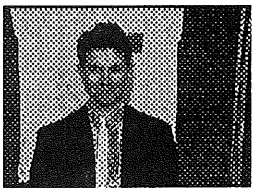
Name: Michael Offringa
Position: Technical Services
Company: Functional Software
Email: moff@fs.com.au
Snail: PO Box 266 Willetton
WA 6155
Phone: +61 9 246 3331
Fax: +61 9 307 7442

Name: Jorg Ottensmeyer
Position: Computer Technician
Company: Any offers ?
Snail: 66 Bay Rd.
Mt. Martha
Vic. 3934
Comments: Currently studying
"Associate Diploma of Engineering"

Name: Michael Paddon
Position: Unix Hacker
Company: Iconix Pty Ltd
Email: mwp@iconix.oz.au
Snail: 851 Dandenong Road
East Malvern
Victoria, 3145
Phone: (03) 571 4244
Fax: (03) 571 5346
Comments: Interests: Unix kernel, networking,
graphics, image processing

Name: Les Pall
Position: Director
Company: I.T.C. Pty Ltd
Email: les@itcsyd.itc.oz.au
Snail: 27 Hampden St
Paddington NSW 2021
Phone: (02) 360-6999
Fax: (02) 360-6695
Comments: X Windows, Graphics, Real-Time

Name: James Patterson
Position: System Administrator
Company: Univ of Tasmania
Email: jim.patterson@cc.utas.edu.au
Snail: GPO Box 252C
Hobart TAS 7001
Phone: +61 02 20 2811
Fax: +61 02 23 1172



Name: Tom Pledger
Position: Software Developer
Company: Peace Computers NZ
Snail: PO Box 37580
Parnell NZ 1001
Phone: +64 9 307-3974
Fax: +64 9 307-3973
Comments: Graph Theory, and Croquet

Name: Michael Podhorodecki
Position: R&D Manager
Company: Labtam Australia
Email: michael@labtam.oz.au
Snail: 41 Malcolm Rd
Braeside VIC 3195
Phone: (03) 587-1444
Fax: (03) 580-5581

Name: Chris Ramsay
Position: Technical Support
Company: Sequoia Systems (Australia) Pty Ltd
Snail: 1155 Malvern Road
Malvern VIC 3144
Phone: (03) 823-2202
Fax: (03) 823-2290
Comments: Sales/Product Support
UNIX
Communications

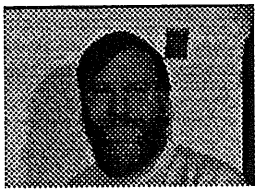
Name: Roy Rankin
Position: Systems Engineer
Company: Alcatel Australia
Snail: 231 Trafalgar St
Petersham, NSW
2049
Phone: 02 6905251
Comments: Business Systems,
Post sales support

Name: Andrew Raphael
Position: Systems Administrator
Company: CISRA
Email: raphael@research.canon.oz.au
Snail: 1 Thomas Holt Drive
North Ryde NSW 2113
Phone: +61-2-805-2915
Fax: +61-2-805-2929
Comments: When does the band start?

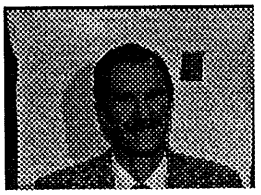
Name: Glenn Rieger
Position: Technical Programmer
Company: Coles Myer
Snail: 800 Toorak Rd
Tooronga VIC 3146
Phone: (03) 829-6514
Fax: (03) 829-6211
Comments: System Management, VMS, UNIX

Name: Arno Rizzuto
Position: Manager,
Information Computer Sevices
Company: D.M.I.D.
Snail: 228 Victoria Pde
East Melbourne VIC 3002
Phone: (03) 412-8286
Fax: (03) 419-0770
Comments: UNIX

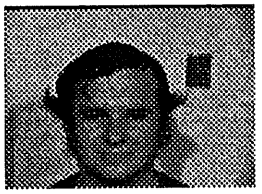
AUUG Faces 1992



Name: Greg Rose
Position: Open Distributed Computing
Company: ACCI
Email: ggr@acci.com.au
Snail: 8 Meadow Street
Concord, NSW, 2137
Phone: (018) 17 4842
Fax: (02) 736 3262



Name: Grahame Rumballe
Position: Senior Systems Support Officer
Company: Queensland Transport
Snail: GPO Box 1412
Brisbane QLD 4001
Phone: (07) 834-2845
Fax: (07) 834-2998
Comments: SunOS, Ultrix,

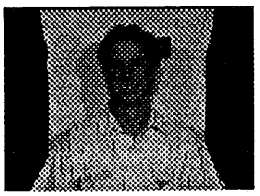


Name: Brendan Searle
Position: Unix Administrator
Company: Defence Dept
Email: -
Snail: Russell Offices L-4-06
Canberra ACT 2600
Phone: +61 6 265 6150
Fax: +61 6 265 6086

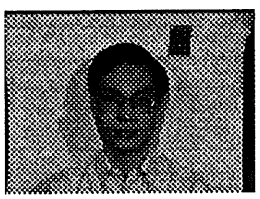
Name: Graeme Speak
Position: Managing Director
Company: Quality Software
Snail: 1st Floor
72 Melville Street
South Perth, 6151
P.O. Box 788
Western Australia
Phone: (09) 474 1477
Fax: (09) 474 1485

Name: Michael Spink
Position: Account Manager,
Mills and Production Planning
Company: BHP IT
Email: dv8@bhpese.oz.au
Phone: (049) 40 2101
Fax: (049) 40 2165

Name: Terence Steele
Position: Systems Engineer
Company: BULL
Email: tps@melb.bull.oz.au
Snail: 677 Victoria St
Abbotsford
3067
Phone: 2464982
Comments: Open systems
transaction processing, CICS
Australian Unix Expert



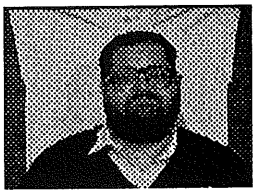
Name: Rick Stevenson
Position: Project Leader
Company: Pyramid Technology
Email: rick@ptcburp.oz.au
Snail: Research Park Centre
Bond University
Gold Coast, QLD 4229
Phone: (075) 950-249
Fax: (075) 722-475



Name: Kam Tam
Position: Network Manager
Company: CSIRO DMS
Email: christam@syd.dms.csiro.au
Snail: PO Box 218
Lindfield NSW 2070
Phone: 02 413 7598
Fax: 02 416 9317
Comments: System & Network Management,
Software Development

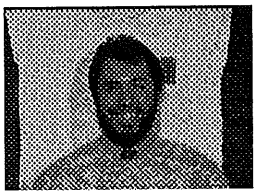
Name: Michael Rourke
Position: Project Manager
Company: Softway
Email: michaelr@softway.sw.oz.au
Snail: PO Box 305
Strawberry Hills, NSW, 2012
Phone: (02) 698-2322
Fax: (02) 699-9174
Comments: Operating Systems, Networks,
Software Engineering

Name: Alan Scott
Position: Manager, Technical Services
Company: Australian Technology Resources
Snail: Trevor Pearcey House, Traegar Court
Fern Hill Technology Park,
Bruce, ACT, 2617
P.O. Box 729
Jamison Centre, ACT 2164
Phone: (06) 251 1100
Fax: (06) 251 2464



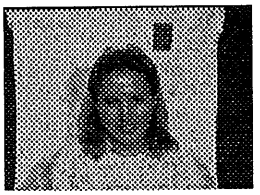
Name: Zoltan Somogyi
Position: Lecturer
Company: University of Melbourne
Email: zs@cs.mu.oz.au
Snail: Parkville VIC 3052
Phone: (03) 282-2401
Fax: (03) 282-2444
Comments: Data Bases

Name: Stephen Spence
Position: System Administration
Company: CITEC
Email: sgccses@citecuc.citec.oz.au
Snail: 317 Edward St
Brisbane QLD 4000
Phone: (07) 227-6969
Fax: (07) 222-2249
Comments: Facilities Management

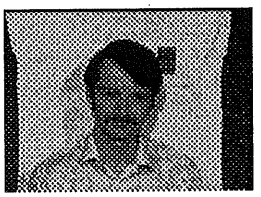


Name: Bart Steanes
Position: System Administrator
Company: Wormald
Snail: 176 South Creek Rd
Dee Why NSW 2099
Phone: (02) 981-0640
Fax: (02) 971-1759
Comments: Networking, C++, DataBase

Name: Andrew Steele
Position: Progammer Analyst
Company: BHP Information Technology
Email: fozzy@bhpese.oz.au
Snail: PO Box 216
Hamilton NSW 2303
Phone: (049) 40-2101
Fax: (049) 40-2165
Comments: Manufacturing Systems, Real Time

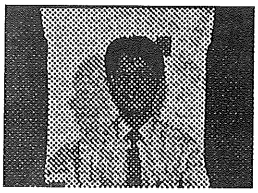


Name: Narelle Stone
Position: Systems Administrator
Company: Griffith University
Email: n.stone@itc.gu.edu.au
Snail: Kessles Rd
Nathan QLD 4111
Phone: (07) 875-7993



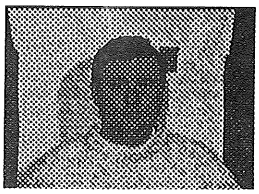
Name: Digby Tarvin
Position: Technical Director
Company: Tesseract Pty Ltd
Email: digbyt@runxtsa.runx.oz.au
Snail: 53 George St
Redfern NSW 2016
Phone: (02) 214-6694
Fax: (02) 698-8881
Comments: System Programming (OS 9 and Unix),
Real-time Progamming

AUUG Faces 1992

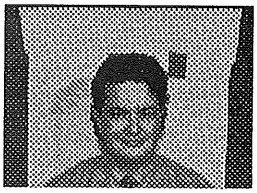


Name: Eng Sin Teoh
Position: UNIX Support
Company: DEC
Email: engsin@anjing.enet.dec.com
Snail: 836 Whitehorse Road
Box hill
Victoria 3128
Phone: +61 3 8959416

Name: John H. Terpstra
Position: Managing Director
Company: Aquasoft Pty Ltd
Snail: PO Box 105
Miranda NSW 2228
Phone: (02) 540-3154
Fax: (02) 540-4016
Comments: C, C++, Unix Development, X Windows Networking,
Integration of Disparate Systems



Name: Chris Terry
Position: Facilities Manager
Company: QLD Dept Resource Industries
Snail: PO Box 194
Brisbane QLD 4001
Phone: (07) 237-1394
Fax: (07) 221-9517
Comments: Systems Administration, Networks
X Windows



Name: Stephen Thompson
Position: Technical Officer
Company: Labtam Australia
Email: stephen@labtam.oz.au
Snail: 41 Malcolm Rd
Braeside VIC 3195
Phone: (03) 587-1444
Fax: (03) 580-5581

Name: Mark Treacy
Position: Systems Programmer
Company: Labtam Australia Ltd
Email: mark@labtam.oz.au
Snail: 41 Malcolm Rd
Braeside VIC 3195
Phone: +61 3 587 1444
Fax: +61 3 580 5581

Name: Michael Tuke
Position: System Software Manager
Company: ANL Limited
Email: mjt@anl.oz.au
Snail: 432 St Kilda Rd
Melbourne VIC 3004
Phone: +61 3 869 5364
Fax: +61 3 869 5110

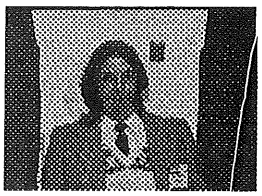
Name: Christopher Vance
Position: Associate Lecturer
Company: University College, UNSW
Email: Christopher.Vance@adfa.oz.au
Snail: ADFA, Northcott Drive
Canberra ACT 2601

Phone: +61 6 268 8051
Fax: +61 6 268 8581

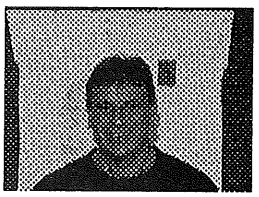
Name: Michael Wagner
Position: Open Systems Specialist
Company: Systems Services
Snail: Level 6
32 Grenfell Street
Adelaide SA 5000
Phone: (08) 212-2800
Fax: (08) 231-0321
Comments: Unix Systems Support,
Networking, System Integration

Name: Gavin Watson
Position: Systems Administration
Company: Mincom Pty Ltd
Email: gavin@mincom.oz.au
Snail: PO Box 72
Stones Corner QLD 4120
Phone: (07) 364-9999
Fax: (07) 394-2844
Comments: Systems Admin, UNIX

Name: Mark Watson
Position: Unix Manager
Company: Walter & Eliza Hall Inst.
Email: mark@wehi.edu.au
Snail: PO Royal Melbourne Hospital
VIC 3050
Phone: +61 3 345 2555
Fax: +61 3 347 0852



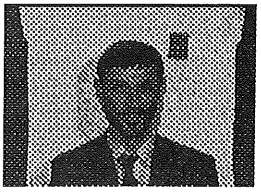
Name: Graeme Wightman
Position: Systems Administration
Company: SEQEB
Email: bss_graeme@seqeb.gov.au
Snail: 150 Charlotte St
Brisbane QLD 4000
Phone: (07) 223-4150
Fax: (07) 221-7556
Comments: Unix, Real Time, Networks



Name: Bruce N. Winzar
Position: Systems Manager
Company: La Trobe University
Email: bruce@sheoak.bcae.edu.au
Snail: PO Box 199
Bendigo VIC 3550
Phone: (054) 447-220
Fax: (054) 447-777
Comments: Systems Management

Name: Arnold Wong
Position: Systems Consultant
Company: Australian Technology Resources
Snail: P.O. Box 1105,
West Perth,
W.A. 6872
Phone: (09) 483 8111
Fax: (09) 321 5021

Name: James Woods
Position: Data Base Administrator
Company: Defence Service Homes
Email: jamesw@thingy.dsh.oz.au
Snail: PO Box 21
Woden ACT 2606
Phone: (06) 289-6672
Fax: (06) 285-2608
Comments: Data Base, 4GL



Name: Geoff Woolfenden
Position: Systems Engineer
Company: AIPC
Email: geoff@aipc.oz.au
Snail: 331 Johnston St
Abbotsford VIC 3067
Phone: (03) 419-7988
Fax: (03) 416-0060
Comments: Office Automation, Aster*x

Name: James Worsley
Position: Technical Applications Support
Company: Ericsson Australia
Email: epajmw@epa.ericsson.se
Snail: 61 Riggall St
Broadmeadows VIC 3047
Phone: (03) 301-1787
Fax: (03) 301-1361

AUUG Faces 1992

Name: Chan Yong
Company: Labtam Australia
Email: chan@labtam.oz.au
Phone: 03 5871444
Fax: 03 5805581

Name: John Young
Position: Marketing Director
Company: Yindi Systems
Email: 100026.2046
Snail: 28 Congewoi Road
Mosman NSW 2088
Phone: (02) 960-4357
Fax: (02) 960-4357

Name: Tracey

Name: Virtual Reality Demonstration

Name: The organisers

SUN - The Sydney Unix Net †

Piers Lauder

Basser Department of Computer Science
Sydney University

ABSTRACT

The Sydney Unix Net is a simple implementation of a user initiated file transfer facility. SUN is a "host supported" network, and considerations of low overhead have lead to an efficient design. The network is self configuring with an optimised routing algorithm.

Introduction

SUN consists of linked hosts (nodes) with unique names, where links between nodes may be unreliable and quite slow. Using this network, a simple system has been developed to provide reliable host-host file transfer. The system is implemented in two levels. Level one provides an error free path between nodes; and level two implements a host-host protocol, and maintains a network topology file for routing calculations. Files are transmitted through the network until they reach their destination where they are spooled for later collection by the remote user, who is informed by mail of the arrival of files from the network.

User Interface

The user interface is as simple as possible. A network address is defined as a username:host pair, i.e. the name of the user on the remote host is separated from the remote host name by a colon. This form of address is understood by the mail programs, some print commands, and the netsend command.

Two special file transfer types are recognised in addition to user-to-user file transfers, namely mail and print. Mail is automatically delivered on the remote host by invoking mail, and print files are handled by invoking the print spooler, but user files are spooled in a holding directory for later collection by the designated user. The arrival of files is notified by mail to the user. To complete the file transfer (and assume ownership of the file), a user must invoke the netget program to retrieve the file from the holding directory. Uncollected files can be easily

† Reprinted from AUUGN Volume 2 Number 6

discarded from time to time.

Design

The network programs operate on two levels with clearly defined interfaces. The top level (implemented by the program net) accepts files for transmission, calculates the next node in the path, and spools the file together with a host/host protocol header in a directory naming the next host. The lower level (implemented by the program netd) accepts files for transmission to an immediately connected host, negotiates with the lower level on the remote host for file transfer to start, and then uses a node/node protocol to transfer the file reliably. On arrival at the next node, the file is passed to the upper level for any further routing.

The file transfer mechanism is half-duplex at the lower level, and store-and-forward at the higher level.

Node/Node Interface

The link between hosts provided by the operating system must be a full duplex, byte oriented special file. The name of this special file is that of the node to which it is connected at the remote end. This file is opened for reading and writing by the network daemon responsible for communicating with the next node.

Actual links between hosts may be physical RS232 type lines, either directly connected, or via telecommunications modems, and be handled by the standard Unix tty interface. Many hosts multiplex the physical link using the mx tty interface (mentioned elsewhere) to which the network requires only one port.

Node/Node Protocol

The network daemons communicate with each other over the node-node interface. They use a half-duplex, multi-buffered, positive acknowledgment data transfer protocol. Before file transfer starts, the daemons negotiate the direction of the next transfer. File transfer proceeds with short data blocks enclosed in a protocol envelope consisting of a header and a trailer. The header contains a sequence number used to provide the multi-buffered message flow, and the trailer implements a simple low-cost error detection capability. Messages must be acknowledged (positively or negatively) with a two byte reply consisting of ACK or NAK and the relevant sequence number. Errors cause the re-transmission of all un-acknowledged blocks. Catastrophic error conditions cause a negotiation for file tranfer restart.

Daemon/Spooler Interface

The interface between spooler and daemon is defined by queued files. Each daemon maintains a command directory which it scans for command files. Each command file specifies the path names of

files for transmission. The spooler chooses the next host for transmission by the fact that the name of the host is also the name of the command directory for the appropriate daemon. On the other hand, files received by the daemon are passed directly to the spooler program.

Host/Host Protocol

The network routing program net (the "spooler" referred to above) prepends a host-host protocol header to each file before spooling it in a daemon directory for transmission to the next host. This header contains the source and destination network addresses together with routing information and file parameters. Each file arriving from the net is examined for this header and re-routed if the destination is not yet reached. Each host through which the file passes adds a host/time record to the routing information in the header. Amongst other things, this information can be used for network performance analysis.

Topology File

The routing information in each file header is used to maintain the toplogy file. Each host mentioned in the route is connected to the preceding and succeeding hosts, and these links are maintained in the topology file. It is possible for a topology file to become aware of new hosts in this way, however there are special files whose purpose is to maintain network toplogy files generally. Thus there are "host-up" messages to inform the network of a new hosts, and "host-down" messages to inform the network of broken links. These messages are broadcast around the network by the net programs who also stop any loops. New hosts coming up receive a special file from any immediately connected hosts containing a copy of their topology files, thus immediately informing a new host of the latest network state.

In order to send a file to a remote host, its name must exist in the topology file so that the routing program can find it, unless it is directly connected.

Conclusions

The initial effort producing the software for a usable network took about 2 man-weeks. Since then many requested enhancements have been implemented involving a further 4 man-weeks of work. As it now exists, SUN has proved very helpful in bringing together many people involved in cooperative work in support of our various teaching systems.

October 22, 1980

Security and Enterprise Computing

Chris Schoettle
UNIX System Laboratories
Australia and New Zealand
Tel: +61-2-906-8953
Fax: +61-2-436-4673
Email: cts@usl.com

Abstract

This paper describes the new security paradigm required to secure the corporate computing environment in the 1990's. The existing security paradigm utilizing physical security and add-on security packages is no longer sufficient. A new paradigm is required in which security is embodied in every object in the computing environment. The hardware and operating system software must be secured to provide a suitable foundation for secure applications. Hardware has traditionally been viewed as secure, and now UNIX® System V Release 4.1 Enhanced Security provides security in the operating system that satisfies the new security paradigm. Additionally, a security policy and management controls must be defined for the computing environment. Utilizing the mechanisms that follow the new security paradigm will result in a secure enterprise computing environment.

UNIX is a registered trademark of UNIX System Laboratories in the USA and other countries.

1. Overview

The evolution of computer technology has made it possible for commercial institutions and government agencies to process and store a large volume of sensitive data and to transmit these data among computers. This sensitive information is stored in databases and passed via internal communications using electronic mail. Critical to business is the ability to provide immediate access to this information. However, control of this access is critical to ensure success. Increasingly, computers of different companies communicate with each other, performing sensitive interactions such as Electronic Data Interchange (EDI) type transfers, simultaneously enhancing the efficiency of inter-business communication. A new security paradigm must be introduced that secures the enterprise computing environment. Key to achieving this is the use of a secure operating system.

Secure systems are required in all government and commercial sectors. But the need for secure computer systems is traditionally recognized only by a minority of the commercial and government sectors, such as banks and insurance companies. Many installations mistakenly believe that they are not in danger of having their systems compromised. However, the increased interconnection of computers and the increased number of employees with access to computers dramatically increases the risk of security penetration. As the level of expertise in operating the computer has increased among the general public, so has the potential for computer abuse.

The distributed nature of today's computing environment means that sufficient security cannot be provided by the physical protection of locking systems in a computer room. Putting an add-on security package on top of an existing operating system cannot promise a sufficient level of confidence, since there is no guarantee of security provided by the underlying operating system functions. A new security paradigm must be introduced that secures the entire computing environment. A secure computing environment requires a secure operating system, augmented with a security policy tailored to the environment in which it is used. The operating system must have security as an integral part of its design and implementation, not just an add-on to the system. All functions must be well defined and perform only the actions for which they are intended, without side effects. This secure operating system then provides a foundation upon which secure applications can be constructed. UNIX System V Release 4.1 Enhanced Security is the only general-purpose, open operating system that satisfies the new security paradigm needed to provide a secure enterprise computing environment.

2. Evolution of the Corporate Computing Environment

Before the computer, processing and storing information in the corporate environment consisted of recording it on paper and storing it in a file cabinet. Information was accessed and handled according to procedures, and physical restrictions limited access to the information. Sensitive information in particular was kept in a locked file cabinet when it was not being used.

The introduction of computer systems in the corporate environment was intended to increase the productivity of the work force. As systems evolved, the size of machines decreased and the processing power increased. These changes resulted in significant increases in the use of these systems and in the number of individuals who accessed them. However, the benefits brought to information processing and storage by the evolution of computing also have been tightly coupled with the creation of new problems. The most important of these problems is the security of the computing environment.

In the 1960s, the typical corporate computing environment consisted of large mainframes. These systems ran proprietary operating systems that were provided by the hardware vendor. No security was provided by the operating system, so the system was housed in special, secured rooms which provided security by limiting physical access.

The 1970s and 1980s brought dramatic changes to the corporate computing environment. Advances in hardware technology brought into existence smaller machines that could rival the larger machines produced just a few years before. The mainframe continued to be centrally located and maintained by the corporate MIS Department. Individual departments acquired mini-computers and set up their own facilities for maintaining the systems, outside the scope of the MIS department.

The need for security mechanisms in the system were realized, and basic security measures such as identification and authentication were introduced. These mechanisms were built into new systems and made available as add-ons to existing systems.

The introduction of the PC to the corporate computing environment is considered to be the greatest leap forward in the utilization of computers. Unfortunately, it also represented a tremendous step backward in terms of security. PCs were spread out over companies, and offered little security short of physically locking up the machine. Anyone with minimal knowledge of how to use the PC and the ability to gain physical access to the machine could easily obtain the information it contained. There were no policy statements that outlined how to securely manage the information on these machines. Although there may have been strict rules in a company stating that all files must be stored in a locked file cabinet when not in use, the PC was left out in the open for anyone to access.

The wide spread use of mini-computers and PCs lead to a decentralized environment of autonomous systems, preventing information from being shared. To facilitate the sharing of information and provide increased productivity, systems were connected in a network. This networked configuration, with mainframes serving as the corporate hub for mini-computers or servers, and workstations or PCs, has evolved into the corporate enterprise computing environment of the 90s.

In the late 1980s and continuing into the 1990s, we see systems residing in every part of a company. Computer systems are used by personnel in the mailroom, the clerical staff, managers, all the way up to the CEO. The data on these systems includes personnel information, sales figures, financial information, and R&D information on new products. These systems are accessed by many users. The systems are not only networked throughout a company, but also connected to a global network. Global networks utilize fiber optic cables and satellite feeds to beam information around the world in seconds. This networking goes beyond the bounds of a single company and provides interconnectivity among businesses around the world, supporting Electronic Data Interchange (EDI) type business transactions.

It is now more critical than ever to secure the corporate-wide enterprise computing environment. In order to guarantee security across the network of systems, each autonomous system must be secure. In a networked environment, the security of the network will only be as strong as the security of the weakest system.

3. Computer Crime

Trespassing in the system, altering data, and theft of information, services, or money are all forms of computer abuse or, more correctly, computer crime. The first recorded computer

abuse occurred in 1958. In 1966, the first federally prosecuted computer-related crime, the alteration of bank records by computer, was identified in Minneapolis. Recently, reports of computer crime in the news media have become commonplace. The rapid proliferation of computers in all sectors of commerce and government, combined with the increased computer literacy among the public, has caused the incidences of computer crime to grow exponentially. The probability of being convicted of a computer crime is approximately 1 in 22,000, and many computer crimes involve significant value. This combination of low risk and potentially high profits means that computer crime can be lucrative. In 1988, the estimated cost for computer crime in the US was a staggering $555 million, according to the National Center for Computer Crime Data (NCCCD). Additionally, the NCCCD estimates that only 6% of computer crime is actually reported. Companies often believe that it is better not to disclose these incidences because they would lead to a loss of customer confidence.

A common misconception is that computer crime is usually caused by a hacker trying to break into a company's computer. In fact, according to the Data Processing Management Association (DPMA), 81 percent of all computer crimes are committed by the company's own employees. Theft of information or services and alteration of data or software account for 54 percent of computer crime. However, 44 percent of all computer crimes involve theft of money. Although it is important to keep *unauthorized* users off the system, security mechanisms are also needed to ensure that *authorized* users do not abuse the system. Another misconception is that the perpetrator of a computer crime is a person very knowledgeable about computer systems. This is not necessarily the case: only 24 percent are committed by programmers while 39 percent are committed by clerical personnel, managers, and other users.

4. The Need for Security

Increased concerns regarding the disclosure or modification of computerized information and fear of unauthorized system access have expanded the market for secure systems well beyond the US Government. The need for secure computer systems is being realized internationally in both the commercial and government sectors. Security is a concern for all organizations with assets that are controlled by computers. It is imperative that unauthorized users be stopped from gaining access to the system and that valid users be stopped from accessing information for which they are not authorized.

Computer systems may exhibit vulnerabilities due to poor design and insufficient quality control. However, it is most common for system vulnerabilities to result from poor administrative practices. A secure system must do its best to remedy these situations. Efforts within international standards bodies, governments, and commercial sectors, are actively defining guidelines and criteria to provide secure systems. The primary governmental influence has been from the US and several European countries defining the *Trusted Computer Systems Evaluation Criteria* (TCSEC), and the *Information Technology Security Evaluation Criteria* (ITSEC) respectively. One of the better known efforts in the commercial sector is the *Commercial International Security Requirements (CISR)*, the result of a consortium headed by American Express Travel Services and Electronic Data System Corp. (EDS). Bellcore is conducting an effort in the telecommunications industry to define security requirements for the Regional Bell Operating Companies. Other efforts are also being conducted by X/Open and the European Economic Community (EEC).

Most of these standards define the features that should be included in the system, and some define the procedures that must be employed in the development of secure systems. However,

the security of the system's software and hardware must be augmented by a security policy and proper management control of the policy, if a secure environment is to be maintained.

4.1 Security Policy and Management Control

Since information is the lifeblood of most companies, protecting a company's information has always been critical, whether it is stored in file cabinets or on computers, However, a secure computer system alone will not solve the problem. A security policy must be defined that addresses company-specific needs and the implementation of management controls to make sure that security is provided. This policy must define information values, information protection responsibilities, and organizational commitment. The policy must be carefully explained, understood by all employees, and enforced by proper management controls. If the policy is correctly implemented, it will assure that access to information is properly controlled, information and programs are changed only in a specified and authorized manner, and authorized users have continued access to information and resources. For instance, we all know that when we approach a stop light we may go if the light is green and we should not go if the light is red. This behavior is based upon a policy, and is enforced by the police.

In many business environments, access to information is restricted, such as storing it in a locked file cabinet. The file cabinet must be reasonably constructed so that it cannot easily be broken into. The people who have access to the information must understand the proper methods for handling this data, such as returning the information to the appropriate folder and locking the file cabinet when they are done accessing the information. Again, this policy must be defined for the company or organization and must be supported and enforced from the top-level management on down.

The rules described in the previous examples also hold for a company utilizing computers to store and process information. First, a policy must be defined to describe how information is to be handled and stored. For instance, storing the information on a diskette and leaving the diskette on your desk is not very secure. The policy must be supported by upper-level management. Depending upon the complexities associated with the environment, it may take up to a year to define this policy. The policy must then be explained to all personnel. All levels of management must support the policy and they must also abide by and enforce the policy. Existing computer systems must be examined to make sure that they abide by the policy, or new systems that do abide by the policy must be introduced. Even when a secure system is being used, there are still choices that must be made in the system configuration to make sure that the system conforms to the specific policy of the company. Whenever new software is introduced into the company's computers, it must be scrutinized to make sure that they, too, abide by the security policy.

The definition of an organizational security policy and associated management controls is the essential first step towards a secure environment. Only by combining the policy and controls with a secure operating system and secure applications can a secure computing environment be obtained.

5. UNIX System Security

The UNIX operating system is often perceived as lacking security. This perception stems from press coverage of bugs and/or break-ins on UNIX Systems. This has led many people to believe that open systems contain bugs and security holes. On the contrary, since the source

code is available and scrutinized by a wide audience, there are likely to be less bugs than in a proprietary system. Furthermore, there are many variants of the UNIX System and not all receive the rigorous testing and quality control as UNIX System V. For instance, machines running UNIX System V were not affected by the legendary *internet worm*, which infected many UNIX Systems on the internet.

The UNIX System was originally developed in an open R&D environment in which a paramount concern was the free and easy exchange of information. Guest logins without passwords, unprotected system files, and unrestricted dial-in lines were typical in such an environment. Although the system was designed with fundamental security features, such as user defined protection for files, they were usually viewed as unfriendly and consequently were rarely utilized. The predominant problem, however, has been lax or improper system administration. This was further compounded by an inadequate amount of security and administrative documentation, software holes, and the ability of unprivileged users to read the password file (which contained encrypted versions of the passwords).

Systems with a high degree of security have become a popular topic. However, in 1985 USL started the investigation and engineering of a release that would provide a high degree of security in the standard UNIX System V product. The result of this effort is UNIX System V Release 4.1 Enhanced Security, the only general-purpose, open system designed to satisfy the new security paradigm of the 90s.

6. The Security Paradigm

Existing security methods are no longer sufficient to meet the security needs of the enterprise computing environment. The existing security paradigm associated with the corporate computing environment must be examined and redefined to keep pace with the evolution of enterprise computing. In order to achieve a truly secure system, the existing security paradigm must be shifted.

6.1 Shifting The Security Paradigm

Limited physical access, once the sole means to secure computer systems, is not adequate in todays corporate computing environment. Locking a system in a room does little to protect the system, when it is networked and is easily reached from outside. Physical security is a necessary component of the corporate enterprise computing environment; however, it is not adequate by itself.

The realization that physical security alone was not sufficient caused security to be introduced into the computer operating system. Most secure general-purpose operating systems are created by putting an add-on package containing security features on top of an existing system. This add-on approach to the operating system depends on underlying operating system functions that are not guaranteed to be secure.

For instance, many people believe that if files are encrypted, they have sufficient security. However, the information must be handled by the operating system when the encryption is performed (e.g., read in from the terminal). If the operating system functions have not been verified to be secure, side effects can occur that compromise the data. For instance, the buffer used to hold a copy of the terminal input may be allocated to another user without being cleared. In this case, the operating system functions and the encryption routine worked as documented. However, an undesirable side effect occurred: the plain text information could be

read by the next person to allocate the buffer. The add-on encryption mechanism failed to provide the expected security.

In contrast to the current add-on security approach, the shifting security paradigm dictates that security must be designed as an integral part of the system architecture. The system must be designed and built in a modular fashion, and each module must itself be secured so that it contains protective firewalls. This practice must start in the operating system and be continued in all functions and applications added to the system.

This approach to security provides the foundation to build secure applications on the system. If a secure foundation is not provided, individual applications will try to enforce security. This will result in each application having its own security policy. To maintain proper security on a system, a single point of mediation must exist and should be called whenever a security-relevant decision is required. This mediation task should be handled by the operating system. With a secure operating system in place, applications can request mediation by the operating system and always receive the same, correct answer, based upon a single system security policy.

This approach provides further benefits by reducing the amount of overhead commonly incurred with secure systems. This includes both system performance and the amount of time required to administer the system.

In order to provide the security and degree of trust needed to protect a company's valuable assets, the information on their computers, the entire enterprise configuration must be secured. All objects in the computing environment must be secured and properly managed.

6.2 The New Security Paradigm

The new security paradigm dictates that security be provided in every object in the enterprise computing environment. Utilizing this object-oriented approach, the hardware, all functions of the operating system, applications, and even users are viewed as objects. Security must be designed as an integral part of these objects (for users, the best that can be achieved is to provide a policy and proper education). Each object must have a specific set of permitted operations and clearly defined procedures for its use.

The security model is analogous to the layering of an onion. Security is first achieved at the lowest level object or inner-most layer. Once this foundation is provided, each successive object, or layer, built above is secured until all objects are secure.

The lowest layer in a system is the hardware. The hardware, such as the microprocessor, is generally viewed as secure, but the hardware alone can not provide the firewalls to assure security. The hardware merely provides a secure foundation upon which to build secure software.

The objects that comprise the operating system must be secured. First, the most elementary or lowest level of functional objects, which communicates with the hardware, must be secured. Once these low level objects have been certified to operate correctly and in a secure fashion, there is a high degree of confidence in these objects. Then, the next level of objects which provide more sophisticated functions, such as the memory management subsystem, and utilize the lowest level objects, must undergo the same certification process. This process is repeated, working its way out into other areas of the operating system, until the entire operating system has been certified to operate correctly.

Utilizing this new approach the entire system is not only secure, but has also been assured to work the way it is supposed to, without side effects, resulting in a higher quality system. The operating system now provides a high quality, secure foundation upon which higher level objects, such as applications, can be built securely. For instance, many database products provide their own security features. The same problem described above, where an encryption mechanism did not provide adequate security are true for the database application. Security cannot be guaranteed unless the database is utilizing a secure operating system.

This functional, or modular, approach also results in improved performance, as evidenced by UNIX System V Release 4.1 Enhanced Security. Add-on security packages and special-purpose secure systems generally result in significant performance degradation, sometimes as much as 30 to 50 percent. While system performance will degrade in conjunction with the level of auditing selected for an individual system, the performance of UNIX System V Release 4.1 Enhanced Security is exceptional. The performance of UNIX System V Release 4.1 Enhanced Security is very close to System V Release 4: it ranges within 96 to 97 percent with all security features enabled but no auditing enabled, within 93 to 94 percent with all security features enabled and default auditing enabled, and within 85 to 90 percent with all security features and full auditing enabled.

The combination of a security policy enforced by management controls, proper user education, and secure applications built on a secure operating system -- such as UNIX System V Release 4.1 Enhanced Security -- and hardware platform will provide the secure enterprise computing environment required by the integrated business computing model of the 90's. UNIX System V Release 4.1 Enhanced Security is the only system available that delivers the new security paradigm.

7. Summary

The existing security paradigm is not sufficient for the corporate computing environment of the 1990s. A new paradigm is required in which security is embodied in every object in the computing environment. The hardware and operating system software must be secured to provide a suitable foundation for secure applications. UNIX System V Release 4.1 Enhanced Security is the only general-purpose, open operating system that can satisfy the new security paradigm. Also, applications must be developed in a secure manner and a security policy must be defined and enforced by management controls. Users must understand the security policy and receive education about how to maximize the security of the computing environment. Utilizing these mechanisms that follow the new security paradigm will result in a secure enterprise computing environment.

Achieving Real-Time Unix through Kernel Replacement

Mitchell Bunnell

Lynx Real-Time Systems Inc.
16780 Lark Avenue
Los Gatos, CA 95030

Abstract

This paper describes how a Unix time-sharing system can be transformed into a real-time computing system by replacing the standard Unix kernel that is supplied as part of the operating system with a LynxOS kernel. The distinguishing characteristic of this approach is the combination of extremely high real-time performance and complete compatibility to Unix systems of the subsequent system. Compatibility was achieved by providing the LynxOS kernel with a binary interface exactly the same as that of the Unix kernel it replaces and equivalent functionality for all system calls. Real-Time performance was achieved, not by modifying the existing Unix kernel, but by designing and writing a brand new kernel using the appropriate data structures and algorithms for predictable real-time response. Actual performance measurements are provided in this paper for three different computers with different CPU architectures. These measurements show the throughput and real-time response for both a standard Unix Operating System and a Unix Operating System running with the Lynx kernel.

1. Introduction

Over the past few years it has become clear that Unix has become the open standard for multitasking operating systems. There are many attractive reasons for going with an open [2], standard operating system. These include vendor independence, portability of applications across different hardware, availability of off-the-shelf applications, the large number of trained programmers and users on the O.S., connectivity between dissimilar computers, support for standard graphics user interfaces.

The world of real-time computer applications has so far resisted adopting Unix as a standard platform in favor of many incompatible proprietary solutions. Despite the advantages of open systems, the overriding factor in the choice of an operating system in a real-time environment is performance. Standard Unix fails to meet the performance needs of most real-time applications.

Real-time applications include such things as high speed data acquisition, chemical process control, robot control, machine automation, and real-time simulation. What differentiates these applications from non-real-time applications is the need to respond to timed or external events within a fast, predictable period of time. A real-time operating system is one that is guarantees application code will be

executed in a fast predictable amount of time. Unix is not a real-time operating system. Even so there is a strong and growing interest in using Unix for real-time applications.

2. Why replace the Unix kernel for real-time?

The Unix Operating System consists of many parts. There are over 200 utilities that are included as part of Unix. These utilities provide Unix with its user interface, networking, system administration, software development, and system monitoring. Unix contains many servers, as well, for file serving, graphics interface serving, network serving, etc.. Despite the label of monolithic which is sometimes given to the Unix kernel, the kernel is but one small part of the full Unix environment. It is however the kernel that binds all the other parts of Unix together; it is interface between all programs and the hardware, and it provides the low level multitasking and scheduling of when things will be done. Because of the way the Unix kernel handles these responsibilities, it inhibits the predictable response of even properly written real-time applications that would attempt to run under it. The kernel is the one part of the Unix operating system that keeps a Unix based system from being used for hard real-time.

2.1 Unpredictable response

In a computing system where some tasks need to be run at predictable times and some tasks do not, it is the operating system kernel that is responsible for scheduling and executing time critical tasks on time. The Unix kernel was not designed to deal with tasks that have hard timing deadlines. It was designed for multi-user, time-sharing work and endeavors to be fair with its task scheduling while striving for maximum total system throughput.

There are several major problems in the Unix kernel design which preclude it from being used for real-time applications. The Unix kernel does not bound the amount of time to preempt a task using system services; it does not bound the amount of time tasks are delayed by interrupt servicing, and it does not schedule tasks based directly on a user set priority.

The Unix kernel itself is non-preemptive [2]. This means a task, no matter how low priority, executing a system call cannot be preempted to run higher priority tasks until the system call is completed. The maximum time to execute some system calls has no known bound but will sometimes extend for several seconds. A real-time kernel must have fast bounded times for executing all system calls that real-time tasks may use. Longer system calls that don't have known bounds would have to be preemptive. If the length of time to execute a system call is based on the arguments passed, the system call must be re-entrant because it may be used by high and low priority tasks simultaneously. The Unix kernel relies on its non-preemptibility and its non-reentrancy to protect its data structures; it would take major changes to make it fully preemptive.

The Unix kernel does not deal with interrupts in an effective way for real-time systems. Device I/O produces asynchronous interrupts that must be handled by a kernel and its device drivers. All interrupt processing under a Unix kernel runs at higher priority than all user tasks. Most Unix systems have

interrupt routines that can execute for many milliseconds. Even worse, the number of interrupts generated by any single device is not bounded. Task execution could, then, be delayed for long, unpredictable periods of time while these interrupts are being serviced. It would be impossible to run a real-time application in the presence of such interrupts because real-time tasks must be executed at precise intervals or within a certain period of time after some external event.

The Unix task scheduling is also not appropriate for real-time. True task priorities are set via an algorithm based on the user's desired priority for the task, what resources the task has had to wait on in the recent past, and the percentage of CPU time the task has used. This is in direct conflict with the types of scheduling that need to be implemented in order to prove that real-time tasks will always meet their timing deadlines, such as rate-monotonic scheduling [1]. These scheduling algorithms call for the user to be able to set the true task priority for all tasks in the real-time scheduable set.

2.2 Modified Unix kernels have not worked

There have been many attempts at modifying the Unix kernel so that it would be more suitable for real-time tasks. None of these has been entirely satisfactory due to a number of reasons. Most of the attempts have been based on providing preemption points or simply adding semaphores to protect the normal Unix kernel data structures. The problem of overly long and unbounded interrupt routines has not been addressed. They do not offer ROMability and so are not popular for real-time embedded systems. Performance of interprocess communication and I/O under these modified kernels is typically the same as the standard Unix kernel which is normally judged to be rather slow.

The method of adding preemption points has been the most popular way of improving the real-time performance of Unix. It involves the least amount of change to the kernel. Usually between 30 and 300 preemption points are added. The normal goal is to make the system better for "soft" real-time applications like transaction processing. The worst case preemption time can be improved from several seconds to the tens of milliseconds range. But even this preemption time is not guaranteed.

Adding semaphores throughout the Unix kernel is a newer method for improving preemption latency that is becoming popular. It got its start from companies trying to make Unix work for symmetrical multiprocessing. With this method the longest non-preemptive region is made much shorter. The problem with simply providing semaphores for all data structures to improve real-time response is that long task preemption delay is simply traded for long blocking regions. Since both the preemption delay and the time in blocking regions [4] affect the schedulablity of real-time tasks that interact, very little actual improvement is achieved for most real-time applications. This is due to the fact that the Unix kernel data structures were never designed properly for fast access.

3. LynxOS as a kernel replacement

Because of the problems with using a standard or even modified Unix kernel for real-time a different approach becomes necessary. The approach studied involves replacing the Unix kernel with a completely different kernel that meets the needs of real-time applications but can also run all the applications, tools,

and interfaces that normally run as part of Unix. The kernel which was designed for this purpose is the LynxOS kernel.

We began the LynxOS project [2] in May of 1985. Our goals were to design a real-time operating system from the ground up to support hard real-time applications but give it an industry standard interface, namely Unix. Other goals included expanding the Unix I/O system, portability, improved robustness, ROMability, flexible configuration, and IEEE POSIX 1003 conformance. To meet these goals we were forced to provide many more features than any real-time kernel has ever had, and better real-time performance than any Unix system has ever known.

3.1 Real-Time by design

The model for most real-time applications is that of multiple tasks, each with its own response needs. LynxOS supports these applications by providing priority preemptive scheduling, allowing true user set task priorities and task preemption even in the kernel. LynxOS goes even further by executing extended asynchronous interrupt processing at task priority levels. The worst case preemption delay and blocking times are known for the kernel and can be used in conjunction with task execution times to ensure that independent tasks will always meet their timing deadlines.

The LynxOS kernel was designed to be fully preemptive without adding long blocking regions. Data structures used in the kernel which are implicitly shared, that is, shared without the application programmer being aware of it, are protected by temporarily disabling preemption during access (in effect a priority ceiling protocol [2]). To ensure that preemption is disabled for a very short period of time, Lynx data structures were built for very fast, deterministic access. Data structures that are explicitly shared, such as a data file or I/O channel, have fast, but longer, access times and so are protected with semaphores.

LynxOS has a unique interrupt handling system. Most operating system simply execute interrupt processing to completion, allowing it to be preempted only by higher priority interrupts. This traditional method of dealing with interrupts does not work well with today's computing systems used for real-time. Because today's computers may be attached to a network, access mass storage, or handle a friendly user interface, the computer can receive many hardware interrupts from many different sources. These interrupts would essentially steal time from user tasks. In a worst-case scenario where interrupts come one right after another, user tasks would never have a chance to run at all. To solve this problem LynxOS executes the bulk of interrupt servicing at a user task priority level through the use of dedicated kernel threads. An interrupt thread priority is based on the highest priority user task that accesses the device generating the interrupt. Furthermore, interrupts are re-enabled by their kernel thread. This puts a bound on the amount of time high priority tasks are delayed by interrupts. Thus, under LynxOS, a predictable system can be built in the presence of the unpredictable interrupts typical in today's computing environments.

The known blocking and preemption delays of the LynxOS kernel, even in the presence of interrupts, make it possible to use analytical methods to ensure a set of real-time tasks will always meet their

deadlines. For example the rate-monotonic scheduling algorithm [1] can be used quite effectively under LynxOS. Under this algorithm each task is assigned a fixed priority based on its period. A task with a shorter period is given higher priority. The modified theorem for n independent periodic tasks that will always meet there deadlines running under a Lynx environment and in the presence of interrupts is given by:

$$\frac{C_1}{T_1} + \quad \cdots \quad + \frac{C_n}{T_n} \leq n\,(2^{1/n} - 1)$$

where C is the task execution time + worst-case preemption time + context switch time, and T is the task period. Only the worst case preemption time for the highest priority task must include the interrupt handling time (only the part run at interrupt priority) for each interrupting device.

3. 2 Binary compatibility with Unix

Despite the need for supporting real-time tasks, complete real-time applications need much of the same functionality from the operating system as normal applications. In addition, it is sometimes most efficient to share a computer between real-time and non-real-time application programs. The old strategy of providing a real-time kernel with only a small subset of the functionality found in a time-sharing kernel does not meet these needs. That is why it was determined to provide all the functionality found in the Unix kernel within LynxOS. Specificly the functions in both BSD 4.2 Unix and System V.3 Unix kernels have been placed into the Lynx kernel. Its Unix compatible system call interface meets ABI specifications and therefore gives LynxOS the capability of running Unix utilities and off-the-shelf applications.

The Unix kernel contains many functions. It provides for a hierarchical file system, process creation, program loading, task control via signals, networking, etc.. All of these functions were given to the Lynx kernel. They operate in the same way as under Unix, but LynxOS provides these functions using different underlying mechanisms due to the fact LynxOS was independently designed with real-time constraints placed on its design.

The interface to a kernel such as the Unix or the LynxOS kernel consists primarily of system calls. A well defined signal context and execution context are also part of this interface. Most CPU manufacturers publish an Application Binary Interface for their CPUs. This is supposed to be a general interface but really specifies an interface to a Unix kernel because it describes the parameters and system call numbers that match only the Unix kernel interface. LynxOS has become the first non-Unix derived operating system to actually meet these ABI specifications. This allows LynxOS to run Unix utilities, GUIs, daemons, and off-the-shelf applications. The only programs that don't run are the few system utilities that use the memory device to interrogate tables in the kernel. Because of this, replacing the Unix kernel with a LynxOS kernel also involves replacing these system utilities.

3.3 Important Extras

Despite the completeness of the Unix kernel for time sharing applications, there are a number of things that real-time applications would find missing. Many real-time applications are realized by embedded systems where all software must run out of ROM. The types of devices supported under Unix is only a subset of the types of devices used in many real-time applications. Although Unix can handle software faults much better than most real-time kernels, it lacks robustness when dealing with, say, exhausted software resources. The standard Unix file system does not allow the preallocation of contiguous data files that are very useful for high speed data acquisition and real-time data bases. Despite the fact that Unix is the most ported time-sharing operating system, a real-time operating system must be ported to more hardware platforms even easier.

The LynxOS kernel provides more than real-time response and a Unix interface in order to support a wide variety of real-time applications. LynxOS is ROMable. Both the kernel and applications can be booted from or executed from ROM without modification. The Lynx I/O system was designed to deal with devices found in real-time computer systems, analog and digital I/O, servomotor controllers, instrument bus controllers. The Lynx file system has contiguous files that are preallocated. A more modular design makes LynxOS more portable than Unix.

LynxOS was chosen as the Data Management System operating system for use in computers on-board United States Space Station Freedom. The already robust LynxOS kernel was brought up to an even higher level of software quality assurance, fault tolerance and safety in order to meet stringent NASA specifications for flight software. This project also added support for real-time Ada [3] under LynxOS.

4. Results

The LynxOS kernel was actually ported to four distinct computer architectures -- the Motorola 147 based on a 68030 CPU, an IBM PC-AT compatible based on an Intel 80386 CPU, the Data General Aviion 5000 based on the Motorola 88000 RISC CPU, and the CDC 4360 based on the MIPS R3000 RISC CPU. In each case a Unix System V.3 binary compatible interface that meets the ABI specifications for the CPU was built into the LynxOS kernel (except for the 68030 because no System V exists which meets the spec.. The same binary interface that Motorola's System V uses was built in instead). The systems were tested on how well Unix software ran on them, and how much interprocess communication was improved. The real-time task response was measured to see just how good a real-time operating system was created.

On all four platforms virtually all the Unix utilities executed perfectly. We ran the Bourne Shell and the Korn shell, the compilers, debuggers (although we had some problem with *adb*), *ls*, *find*, etc.. Only those few utilities that search through the O.S. symbol table in order to gleam information directly from the kernel data structures failed to work. These utilities, like *ps* and *netstat*, are available in LynxOS specific versions. We tested off-the-shelf software including LPI Fortran, Oracle, Informix, WordPerfect, Qcalc, and 20/20. All operated just as they did under the Unix kernel. Network based applications that used both the Streams and Socket based networking were tested and worked well (note: socket based

networking is in BSD Unix not System V, but somehow it made it into some of the ABIs).

Several benchmarks were run to evaluate interprocess communication speed on both standard Unix and LynxOS. An important thing to note is that the exact same binary programs were run under both kernels. A sample of the data follows:

System V messages – messages per second	50 byte	100 bytes	200 bytes
20 MHZ 386 64k memory cache (LynxOS)	186,390	187,872	157,774
20 MHZ 386 64k memory cache (Unix)	90,279	88,742	85,774
25 MHZ 68030 no memory cache (LynxOS)	186,848	179,328	142,701
25 MHZ 68030 no memory cache (Unix)	78,655	76,400	71,301
			256 bytes
20 MHZ Aviion 88K (LynxOS)			166,120
20 MHZ Aviion 88K (Unix)			42,093

Pipes – transfers per second	100 byte	1000 bytes	4000 bytes
20 MHZ 386 64k memory cache (LynxOS)	277,790	133,754	48,360
20 MHZ 386 64k memory cache (Unix)	84,916	50,661	21,566
25 MHZ 68030 no memory cache (LynxOS)	237,320	95,233	34,076
25 MHZ 68030 no memory cache (Unix)	68,757	37,332	14,605

The main purpose for replacing the Unix kernel with a LynxOS kernel is to get better and more predictable task response. In order to measure the real-time response an application program could hope to have under each kernel, the time from an interrupt to the execution of user task code was measured . This measurement was taken many times while the computer was under a load. The load was in the form of many low priority tasks accessing a disk drive, network, and terminal while a high priority task responded to the sample interrupt (including the real-time clock this makes a total of 5 interrupting devices). The typical and worst-case times were recorded:

Task response in microseconds	Typical	Worst-case
20 MHZ 386 64k memory cache (LynxOS)	168	445
20 MHZ 386 64k memory cache (Unix)	340	> 2,000,000 [1]
25 MHZ 68030 no memory cache (LynxOS)	153	433
25 MHZ 68030 no memory cache (Unix)	311	> 2,000,000 [1]
25 MHZ MIPS 65K d&I cache (LynxOS)	38	58[2]
25 MHZ MIPS 65K d&I cache (Unix)	228	> 2,000,000 [1]

(1) Timing hardware used could not measure intervals above 2 seconds

(2) This test was done with a total of 3 interrupting devices, not 5

5. Conclusion

A complete real-time operating system can be provided by replacing the Unix kernel of a Unix operating System with the LynxOS kernel. The resulting operating system displays a high degree of compatibility with Unix by providing all the Unix tools and utilities as well as access to off-the-shelf Unix applications. Applications that make heavy use of inter-process communication will run noticeably faster under this "real-time Unix."

The task response under the LynxOS kernel is significantly better than under the Unix kernel in the typical case and even more so in the worst-case. The worst-case task response is very fast and bounded under the Lynx kernel but not under the Unix kernel. This makes a transformed Unix system running with a Lynx kernel a viable platform for hard real-time applications while standard Unix is not. LynxOS bounded task response makes it possible to prove real-time tasks will never miss their deadlines using analytical methods such as rate monotonic scheduling.

References

[1] C.L. Liu and J.W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard Real Time Envronment," J. ACM, Vol. 20, No. 1, 1973

[2] Inder M. Singh and Mitchell Bunnell "LynxOS: Unix Rewritten for Real-Time", *Seventh IEEE Workshop on Real-Time Operating Systems and Software*, May, 1990

[3] Lui Sha and John B. Goodenough, "Real-Time Scheduling Theory and Ada,"*Computer*, April 1990

[4] L. Sha, R. Rajkumar, and J.P. Lehoczky, "Priority Inheritance Protocols, An approach to Real-Time Synchronization", technical report, Dept of Computer Science, Carnegie Mellon University, November 1987

MHS Column

MHSnet for the Insurance Industry

by
Murray Seymour
Network Manager
DBA Limited

DBA is a company based solely around the insurance industry. We provide a number of services to the industry; we market a software package to insurance brokers called IBS (Insurance Broking System), we sell Unix computers to run the software, and we provide a wide area network called BrokerLink to connect the insurance brokers to insurance underwriters.

The BrokerLink network consists of eight insurance underwriters and around 150 insurance brokers. Each has a Unix computer, but the hardware platforms vary. There are NCR Towers, ACER Counterpoint 19K's, Toshiba Laptops, IBM RS/6000's, and Motorola 3000 and 8000 series computers. All are running MHSnet as the networking software.

There are a number of different types of links which make up BrokerLink. The majority of links are dedicated Telecom lines, over which we run X.25. There are some hardwired serial links, a couple of dial-up links, and some high speed Ethernet links. Recently, we have been experimenting with SLIP (Serial Line Internet Protocol). MHSnet runs over all these types of links transparently.

BrokerLink provides three significant services. The primary purpose is to pass insurance policy information from broker computers to the underwriter systems, and to return confirmations.

Electronic mail is an important means of communication between the two parties, and also internally within DBA.

Finally, remote printing allows reports generated at DBA to be sent to underwriter printers directly. Each of these facilities is implemented using MHSnet.

One particular problem which had to be solved was the method of transmitting binary files over X.25. Since we were using PAD's at either end, with XON/XOFF flow control, and the standard Control-P escape character, we needed a method of file transfer which allowed us to configure exactly which characters should and should not be transmitted. MHSnet's VCdaemon allows us to do that on a character by character basis. This means that binary files can be transmitted without conversion to printable characters, and we retain the throughput advantages of using almost the entire ASCII character set.

BrokerLink is close to a star network, with one major hub having almost all nodes directly connected. This node at times runs up to 350 MHSnet processes. MHSnet's flexible routing policies have enabled us to keep the management of network topology to a minimum. All leaf nodes have minimal information which does not never changes. Only the main routing database on the central machine must be updated when new links are added. The MHSnet initialisation process "netinit" makes it easy to start links at regular intervals and bring up permanent links automatically which go down due to link failures.

In summary, MHSnet provides a solid software basis for building a reliable, easily managed network!

The Return of Doc Strange

Colston Sanger
doc.strange@gid.co.uk

GID Ltd

Colston Sanger is a senior consultant with GID, a notorious UK-based software engineering consulting firm. He is also joint editor of *Open Systems for Europe: towards 1992* (Chapman & Hall, 1991) and *CSCW in Practice: an Introduction and Case Studies* (Springer-Verlag, forthcoming).

Software Quality: There's Not a Lot of it About

Well there isn't is there? Consider this little gem:

```
Return_Stat stopKid()    /* Return_Stat is typedef'd elsewhere */
{
  Return_Stat rc = OK_KID;
  int pid;

  if (kid.pid != NO_KID)
  {
    /* prevent race condition with the SIGCHLD signal */
    pid = kid.pid;
    kid.pid = NO_KID;
    /* Workaround for execv(2) bug. */
    /* We have to kill both the sh and the comms process */
    /* Where there's a 'bug', there's likely to be more than one bug */
    (void)kill(pid, SIGTERM);
    (void)kill(pid + 1, SIGTERM);
  }
  /* Reset restart count */
  kid.count = 0;

  return(rc);
}
```

I mean, even if you had written it, would you admit it?

The Logic of (Horrible) Programming

Now, since you didn't write it, and I *certainly* didn't write it, there's no great rush to search for the guilty party, punish the innocent or praise and honour the non-participants. So we can talk about it, right?

But...how could *ANYONE* write such a horrible (as in 'Horrible, horrible!') piece of code? Two possibilities come to mind:

a. the programmer was asleep

b. he or she simply didn't know any better.

No, that's too easy. The first thing to say is that nobody makes mistakes on purpose: there's always some bizarre logic behind what they do — or, in this case, the way they write programs. The problem here (the '`pid + 1`' problem) seems to be that the programmer has an incomplete, perhaps even

fundamentally mistaken conceptual model of the UNIX multi-tasking environment.

I can imagine how it might come about. Everyone knows that most programmers are just glorified typists who spend most of their day sitting bleary-eyed in front of **vi** or **emacs**. A quick compile and round trip to the office coffee machine, a chat with colleagues, is a memorable event. How to relieve the tedium? Sharpen pencils? Rearrange icons on desktop? Twiddle with **ps** command?

Ah, the **ps** command — so many options. Oh what joy! Almost as good as **ls**.

Maybe, just maybe, the programmer responsible for this fragment of code has discovered the **ps** command. He (or she) has observed an interesting thing with **ps**: that when you start up a shell script or a command that in turn invokes another command, the process-id of the invoked command is indeed often (but not necessarily always) one greater than that of the invoking command. In other words, observing only surface characteristics, he or she has inferred a conceptual model of how processes are created and scheduled in the UNIX environment that is blazingly, obviously incorrect to *those of us who know better.*

No, I don't really buy that. I don't believe that stupidity or silliness come into it.

OK, let's try again. We know that this is being developed on a Sun workstation, so presumably under some sort of windowing system, say **X.11** and **Open Look** or **Motif**. Now, assume that our programmer is more used to programming in C under Microsoft **Windows**. Further, let's assume that he or she has a PC running **Windows** at home. Since **Windows**, and MS-DOS in general, has no notion of a process, and since **Motif** with its three-dimensional boxes *could* conceivably be mistaken for Microsoft **Windows** — is it possible that there was a sort of carry-over effect, what Donald Norman calls a 'description error'?[1]

No, I don't believe that either.

Are we, heaven forbid, in the presence of an unethical programmer, a 'recession-hit software writer' — one who, according to a recent UK Sunday newspaper 'deliberately [adds] errors to clients' programs in an attempt to ensure that they obtain followup work'?[2]

No `<shudder>`, certainly not that.

What else is there? Perhaps only that the `(void)kill(pid + 1, SIGTERM)` was an attempt to solve what seemed at that point to be an insoluble problem: a kludge, but the only way out. Of course it doesn't work (what about the children of the child, for instance? That would need recursion.). Of course it's dangerous, but ... it's late. Maybe he or she meant to come back to it tomorrow.

So what's the moral of this sorry little tale? I guess it could be phrased as:

> *It's OK not to know the right way to do something straight off. That sometimes days of contemplation (== 'not doing anything') are needed before a real understanding of a problem is reached.*
>
> *And that if you do find yourself shovelling layer upon layer of code, riddled with special cases and mutually cancelling areas of complexity, there is almost certainly a better way of doing things.*

Looking at this horrible thing, this shard of code, it's clear that the real problem lies higher up, in a function called `startKid`:

---

1. *Donald A.Norman, The Psychology of Everyday Things, New York (Basic Books), 1988, pp.107-9.*
2. *Susan Watts, 'Software bugs put byte on bosses', Independent on Sunday, 16 August 1992, p.2.*

```
Return_Stat startKid()
{
  Return_Stat rc;

  rc = OK_KID;

  if (kid.pid != NO_KID )
    rc = KID_ALREADY;
  else
  {
    if ((kid.pid = vfork()) == -1)
    {
      Errlog(LOG_ERR, "failed to fork process");
      rc = KID_ERROR;
    }
    else if (kid.pid == 0)
    {
      /* This is the kid */
      /* And now, the strange case of the 'bug' that never was... */
      /* Fix for Sun bug. */
      /* Fork shell and have that run the shell script */
      execlp("sh", "sh", "-c", kid.filename, (char *)0);
      Errlog(LOG_ERR, "failed to execv %s", kid.filename);
      _exit(-1);
    }
    /* This is the parent */
  }

  return(rc);
}
```

God only knows what this 'Sun bug' is. Is the programmer saying, perhaps, that you cannot exec a shell script with **execlp**? That's simply not true. As far as I know, it works in all (but not 4.2?) BSD releases and derivatives, and certainly works in SunOS 4.1.2. It also works in System V.[3] Moreover, if it were true, it's like saying that the fork-exec process creation mechanism doesn't work, which would nullify the existence and procreation of all the UNIX systems in the known universe.

Leaving aside other infelicities such as _exit(-1) (*Why* `_exit`?), the `execlp` line above becomes:

```
execlp(kid.filename, kid.filename, (char *)0);
```

The shell script that is exec'd is prefaced by:

```
#!/bin/sh
#
# Is the !/bin/sh really necessary on BSD-ish systems?
# Shouldn't execlp alone work?
#
# Ensure default behaviour of SIGTERM
trap 15
...
```

3. *See Marc Rochkind, Advanced UNIX Programming, Englewood Cliffs, NJ (Prentice-Hall), 1985, p.106.*

End of story, end of 'bug', end of `pid + 1`.

End Note

This is a slightly revised version of an article that first appeared in *news@UK*, Vol.1 No.4 (September 1992). When it was published, I received several e-mail comments along the lines of 'I thought you were a bit too lenient with the hapless originator of that code.'

However, the point I'd rather make here is that writing code is difficult, and that we all make mistakes:

> Date: Wed, 4 Nov 92 17:56:53 GMT
> >From: andy@eurovi.uucp (Andy Sparrow)
> Message-Id: <9211041756.AA10311@eurovi>
> To: colston@gid.co.uk
> Subject: Re: Software Quality (Doc Strange)
>
> Hi Colston,
>
> ...
> The thing that always gets me about coding is that you look at code you wrote a year ago, and you think ''Oooh, I wouldn't do it that way now'', and you look at code you wrote three years ago, and deny that you ever had anything to do with it! Mind you, I guess that that is just a sign that you are still developing as a programmer, and I would put that down to reading other people's code and trying to work out *WHY* they did it like that... (This only applies to quality code of course).
>
> The one that is *REALLY* embarrassing, is when some program that you wrote years ago gets re-compiled under a later release of the OS, and no longer runs. You generally go in and look at it, and think ''how did that EVER work?''
> ...

Also, that reading code can be a useful, even salutary exercise. Maybe that was the real value, in days of old, of having the source on-line.

Mind you, just spending time with the fragments presented here, together with the program of which they are (anonymous) parts, has been interesting. At times, I've felt almost like an archaeologist digging down through the layers, deciphering, seeing how one bit fits with another.

Terrible metaphor. Gerald Weinberg said what I'm trying to say much better:

> Writing a program is a process of *learning* — both for the programmer and the person who commissions the program. Moreover, this learning takes place in the context of a particular machine, a particular programming language, a particular programmer or programming team in a particular working environment, and a particular set of historical events that determine not just the form of the code but also *what the code does* ...
>
> There are many reasons why programs are built the way they are, although we may fail to recognize the multiplicity of reasons because we usually look at code from the outside rather than by reading it. When we do read code, we find that some of it gets written because of machine limitations, some because of language limitations, some because of programmer limitations, some because of historical accidents, and some because of specifications ... which leads us to believe that studying programming as [a human activity] will bear numerous and not always expected fruits.[4]

4. Gerald M.Weinberg, *The Psychology of Computer Programming*, New York (Van Nostrand Reinhold), 1971, ch.1 'Reading Programs'.

It's these latter points — programming as a process of practical learning, and programming as a multi-faceted, multi-dimensional human activity — that I'd like to take up in future articles.

Acknowledgements

My thanks to my fellow GID-ers, Alan Carter, Andy Greener, David Purdue and Neil Todd, for discussions of earlier drafts of this little harangue.

Australian Systems Administrators' Guild

UPDATE

As was announced in the previous edition of AUUGN, USENIX has established SAGE, the System Administrators' Guild, and AUUG was looking at forming a similar group in Australia. Following further discussion both locally and with the US organisation it has been decided to establish the Australian group as a separate organisation, rather than as a chapter of AUUG.

At a BOF at the Networkshop in Brisbane in December, a draft of the charter was presented and an interim committee was formed. The role of this committee is to oversee the formation of SAGE in Australia, and consists of:

Frank Crawford	frank@atom.ansto.gov.au
Glenn Huxtable	glenn@cs.uwa.edu.au
Greg Rose	ggr@acci.com.au
Hal Miller	Hal.Miller@mel.dit.csiro.au
Keith Haberle	haberle@rivett.mst.csiro.au
Peter Gray	pdg@cs.uow.edu.au

As this is an important topic to AUUG members, AUUG will continue to publish information on both SAGE and SAGE Australia.

Whither The Customer? - or for Whom Do We Administer Systems?

by Kevin Smallwood
<kcs@staff.cc.purdue.edu>

[This is the first of what we hope will be a regular series of articles on issues of a less technical nature, but, nonetheless, very important to system administrators. Free discusion is encouraged and solicited, if you are interested in participating, please contact me. – Bryan, SAGE *Ed.]*

What's in a name? Was Shakespeare correct when he wrote, "A rose by any other name still smells as sweet" ? If a new variety of rose was developed and called "Putrid Stench," would you be very eager to give it a sniff and do you think you would appreciate the smell? What do you think of when you hear the name skunk cabbage? Yes, the crushed leaves do have a skunklike odor, but when thoroughly dried, the leaves are a tasty addition to soups and stews. I suggest that a name IS important; in many cases it provides a mindset and a frame of reference. Yes, it often allows us human beings to pre-judge and anticipate, but we are only human.

I love to visit Walt Disney World and Disneyland, and I am not alone. One of the features about Walt Disney World that totally fascinates me is how clean the facility is. Once I watched as a person loaded film into their camera and then dropped the empty film box on the ground. Within thirty seconds a Disney "cast member" swept up that litter. You blink your eyes and the park is clean. Why is that so difficult for other theme parks?

How is it that Walt Disney Productions can keep a park so clean? I found the answer to this question in "A Passion For Excellence," coauthored by Tom Peters and Nancy Austin. "At Disneyland and Disney World, every person who comes onto the property (the "set") is called a guest. Moreover, should you ever write the word at Disney, heaven help you if you don't capitalize the G."

Is this tripe? I suggest that it is not. Let's look at a contrasting example. Tom Peters describes, "On an all-night flight to Denver our plane stops briefly in Salt Lake City. It is on the ground for only about nine minutes, and then the Salt Lake passengers begin to board. As the new people begin to come down the ramp, the head stewardess turns to her associate and says, 'Here come the animals.'" Think about this: there are a lot of things we do to "animals" that we would not consider appropriate for "Guests." We all know how bad airline service can be.

In his book, "It's Not My Department!," Peter Glen tells the following story:

I had landed late at a hub city and missed my connecting flight. There was another one in four hours, so I decided to try my luck on a commuter airline that had a flight leaving for my destination in less than an hour.

All of the check-in counters for this airline were located in one area, so I waited in line in front of a sign that listed the city of my destination. After about 10 minutes I was told that this particular flight was an earlier one that had already left and that the passengers for the flight I wanted were being checked in at another counter. After another 10 to 15 minutes at that counter, the agent told me my flight check-in had been moved to another counter. I went to the new counter.

I handed my ticket to the agent, and she said, "Sir, you should check in at the next counter." Now since the next counter was part of her counter and she could actually reach over and touch the computer, I said, "Look lady, I am a Customer! This is the third line I've waited in and I'm not waiting in another one. I am a Customer! You take two steps to the left and check me in."

She was a little startled, but she did what I requested. When she handed me my boarding pass, she said, "Sir, we try not to use the 'C' word around here."

They don't try to use the 'C' word because they don't try to treat their Customers as Customers, and it shows.

I would imagine that many of you have similar horror stories. I doubt that many of you would be happy in the above situation; you are a Customer, and you want to be treated as one – your money is being used to employ this person. Yet, how many systems administrators make their "users" perform equally demeaning tasks and jumping through hoops?

What do you require from a "user" to restore a file? A complete dossier neatly typed (on the department's only manual typewriter) in triplicate, of course, hoping that the sheer complexity of the request will dissuade this "user" from ever requesting a file restoration again?

How do you handle notification of system downtime? Do you often say to yourself, "Oh, heck, those stupid users don't know how to use this computer, so they won't even miss it for this fifteen minute reboot. Let 'er rip!"

And, of course, we all know what is best for our "users" in all cases, don't we? We all know that a researcher in biology doing X-Ray Crystallography needs the latest version of "bison" instead of an optimizing FORTRAN compiler. I mean, why would anyone be so stupid as to program in FORTRAN? How many of you have said that once or twice?

So, again, I suggest that a name is important. For this reason, I implore you to refer to the people for whom you administer systems not as "users," but as "Customers", "Clients", or "Patrons". Why? What image do you conjure up when you think of a "user" ? Maybe my high school drug education class left a lasting impression on me, but I have a very vivid image of a junkie huddled in a corner with a needle sticking out of his arm.

There are some sites proud of the fact that they call their Customers "lusers"; they even print it in their newsletters. And, who hasn't heard of "Stupid User Tricks"? With all of this imagery and mindset, it is difficult to display much common (or should I say uncommon) courtesy toward those who justify our jobs and often indirectly pay our salaries. Tom Peters refers to this as "thinly disguised contempt for the Customer." In many cases, I don't think it is all that thinly disguised, either. It is easy to deny maintenance on a FORTRAN compiler for a "user" (or "luser"). Yet, how would we act if we all got paid on commission? Satisfying that Customer would mean a whole different thing, wouldn't it?

How secure are you and your systems administration staff in your positions? Treating the people you administer systems for as valued Customers isn't really necessary, or is it? I know of a couple organizations where the systems administration staff was so smug to think that their "users" couldn't live without their talents, that they treated the "users" as antagonists rather than Customers or even equal partners. This was a clear battle of ego; there were well educated, talented people on both sides of the issue. The "users" of those organizations won in the long run. With enough properly placed complaints about work-stopping road-blocks put in the way of the "users," management was forced to look for alternatives. In both cases, management brought in an outside organization that realized that they had to provide both good technical solutions and perceived quality service to the Customers. Initially, it cost the organizations a little more, but in the long-run, the Customers were pleased with the level of service they received, finished the projects, and the organizations were flourishing in the businesses they were in.

Don't think that can happen to you? I know that the two systems administration organizations felt the same way right up to the day the pink slips were handed out. As more and more independent systems administration organizations that know the value of the "Customer" come into being, the higher the chances of the same thing happening to you if you continue to show "thinly disguised contempt" for your Customers.

Now, I don't mean to imply that simply calling someone your "Customer" is a quick fix. If you don't believe that as a systems administrator you provide a technical service and that those people you administer systems for are your Customers, then you will still exhibit that "thinly disguised contempt for the Customer." However, getting into the correct mindset is a good first step. Furthermore, if you expect to provide service excellence, you must treat your employees the way you want them to treat the Customers. If you don't have employees under you, share this article with your boss. Expert after expert agree that the Customer will be treated no better than the employee is treated.

Would you try a little experiment for me? For just one day, while you are at work, tell yourself over and over "Customer, Customer, Customer." Don't let the word "user" enter your mind. Use "Customer" in your writing, speaking and thinking. At the end of the day, look back and see if you treated people differently. Then, ask yourself how would you want to be treated: as a Customer or an animal?

In future articles, I hope to expand on some of the issues I only touched upon in this article. Many of you know the technical knowledge to be competent systems administrators, but lack an equally important skill: providing quality Customer service. Also in future articles, I hope to respond to your comments, criticism, suggestions, and questions. So, please write.

Counterpoint

by Rob Kolstad
<kolstad@bsdi.com>

Kevin Smallwood points up a fine method of improving the image of system administrators in his article *Whither the Customer?*. I get the idea while reading the article, though, that there's a bit of a power-play or adversary relationship that we, as system administrators, are fighting.

I think there's no denying that a power-based relationship ("I can put your job to the bottom of the printer queue" or "I'm going to tell your supervisor that you didn't get my workstation installed") is probably the least productive kind of interaction that a system administrator can have.

I fear, though, that the notion of user-as-customer, with its subtle implication that "the customer is (always) right" and the (too often one-way) deference to the customer by the "clerk" (a.k.a. salesperson, administrator) pushes the pendulum too far in the other direction.

I have spent the majority of my professional life in industrial computer centers (as opposed to academic ones). I believe that one particularly good tone for user-administrator relationships in the industrial setting revolves more around teamwork than around one co-worker serving the other. When people believe they are part of a team, particularly a team with a common goal, it is amazing how smoothly things can proceed.

Bob Paluck, President and CEO of Convex Computer Corporation, is a master at focusing engineering teams on a common goal. He built a 30 person organization that built a mini-supercomputer from scratch in 18 months. The project included then-revolutionary 20,000-gate gate-arrays, circuits and cabinets, and software (including both an operating system and vectorizing compilers). His key strength was the sharpening of the group's focus. The group responded by working as a team – with members using their strengths to help other members who needed the help. To me, this same kind of teamwork exemplifies the zenith in user-administrator relations.

It is important, though, to avoid going too far in this 'help each other' motif. In the extreme example, team members line up outside some particularly competent person's office, each waiting their turn for the talented person to solve the problem assigned to them as their primary work-task. This situation signifies a talent mis-match inside the working group that requires remedy.

In summary, I think that peer-peer working relationships (typified by the teamwork approach) may have even more benefits than relationships which might appear to be based upon power or presumed superiority.

SAGE Views

Whither the Customer, Part Two

by Wendy Nather
Swiss Bank Corporation
Zurich, Switzerland
<wendy@sbcoc.com>

I would like to respond to the points made by Kevin Smallwood and Rob Kolstad in this column in the previous issue of *;login:*. In general, I agree with both of them: users should be treated as customers, and the customer isn't always right.

As part of a systems administration team responsible for supporting an 1800-node options trading network in sixteen cities around the world, I have seen that a higher level of commitment to the customer does pay off. The willingness to (1) be available and (2) tackle a problem even if it's not your job is a valued trait that is not necessarily found in the corporate cultures in other countries.

I believe that more system administrators are making themselves more available than ever before. Witness the number of portable phones and beepers at the San Antonio USENIX conference. Sure, they're cool toys, but the mystique wears off quickly after a couple of 3:00 am calls from Tokyo. As UNIX spreads even further in the industry, more time-critical systems are using it, and we can no longer afford the luxury of maintaining research systems in a quiet corner of a university during spring break.

Nevertheless, the greater the UNIX presence, the more often we find ourselves maintaining systems for non-technical customers. There is perhaps no greater problem facing system administrators today than the fact that our managers don't know what we do. They don't appreciate the effort, the skills involved (how many managers have suggested that instead of hiring a new UNIX administrator, you just train the filing clerk who has a little extra time on his hands?), or the number of personnel needed to provide the best service possible. The average customer these days sees his computer as a pencil: he doesn't care why it's broken; he wants a new one, and he wants it NOW.

But I've found a simple phrase that often makes all the difference to a customer, no matter how little he knows about the system, and no matter whether the problem falls within your jurisdiction or not: "I'll do my best." It does not promise the customer the moon and the stars; it does not even necessarily promise that you'll fix the problem within a certain time. It does convey your commitment to service, and it is an amazing contrast to the phrases I hear all around me in other cultures:

"He's in a meeting." "He's on vacation for three weeks. No, there's no one taking his place." "It's after five; he's gone home." "I don't know how to do that." "I can't do that." "That's not allowed." "It's impossible."

When you indicate that you're willing to tackle the problem WITH the customer, you create the teamwork that Rob Kolstad describes and defuse the power play and adversarial situations. Customers become immediately more flexible and tolerant when they realize that you really are on their side.

Note that this does not mean accepting rude or abusive treatment from a customer (user). My managers all the way up the ladder have come to the defense of the support groups and quietly insisted on consideration and respect. Both customers and system administrators deserve the same treatment.

Not only does the attitude "I'll do my best" improve life for the customer, but for support staff working together as well. My job would be intolerable without the customer orientation of the staff in the other support groups at my firm. When I ask them for help, I feel like a customer myself.

If you are a system administrator, then supporting the users of that system, whether directly or indirectly, is part of your job. How willing are you to do your job? The answer means the difference between a Mickey-Mouse operation and Disneyland.

The Customer Isn't Always Right; the Customer Isn't Always Even a Customer

by Elizabeth Zwicky
SRI International
<zwicky@erg.sri.com>

Kevin Smallwood makes some interesting arguments in the previous issue's column for calling the people who use the computers you are responsible for "customers" instead of "users." At my site, at least, the correct name for these people is "colleagues." My customer – the entity that pays me to administer computers – is SRI International. There's a very important distinction, there.

Kevin points out that if you go to Disneyland and drop an empty film box on the ground, a Disney employee will smilingly whisk it away, and correctly identifies this as an example of Disneyland's excellent customer service, part of what makes Disneyland such a popular place to go. If you go to Yosemite and drop an empty film box on the ground, and a park employee is standing nearby, do not expect the same smiling service; expect a ticket for littering. Yosemite and Disneyland have chosen different sets of priorities. Disneyland has chosen to take a specific set of people, charge them large amounts of money, and for this money provide them with a bright, clean, happy place. Yosemite has chosen to be open to as many people as possible, at as low a cost as possible, to provide them with the great outdoors with as little modification as possible, and to protect the wilderness for other future uses at the same time.

Most systems work more like Yosemite than like Disneyland. A public-access UNIX system has a Disneyland-style problem; the people who pay for the machine are the people who use it, and the goal is to make them happy. A corporate or educational UNIX system has a Yosemite-style problem; making the people who are using it happy is an important sub-goal, but may be secondary to other things (like cost) in the eyes of the machine's owners.

SRI, my customer, has chosen a set of priorities on the Yosemite side. Many of the things that I need to do in order to further SRI's interests do not particularly please the people who use SRI's computers. For instance, SRI believes that security is a high enough priority that it should be allowed to override convenience; it believes that it is more important that disk space be allocated cost-effectively than that users never run into disk space limitations; it believes that Macintosh software should be bought for the entire division and not for individual users, and that this should be enforced by making packages legally available to everyone from a centralized space. Each of these decisions results in a certain amount of unpleasantness, which I am expected to subject other people to in order to please our joint employer. It's tough to think of someone as a customer when you're being paid to be mean to them as nicely as possible.

Furthermore, it's not productive. Treating these people as customers, instead of as colleagues, encourages them to ignore my areas of expertise. In this society, "the customer is always right." In fact, as we all know, the customer is often wrong. We deal with that partly by assigning new names to customers who can be expected to be wrong; if you buy medical assistance, you are a patient; if you buy teaching, you are a student; if you buy transportation, you are a passenger. These roles come with the expectation that you will defer to a doctor, a teacher, or a pilot, who will apply expertise that you do not have or choose not to exercise. The role of customer comes with the expectation that someone in the sales role will defer to you. It is not appropriate for the people who use the computers that I am responsible for to expect me to serve them, rather than advising and instructing them, when it comes to matters that involve those computers. Just as the passenger doesn't fly the plane – even if the passenger owns the plane – the user does not control the computer.

Fortunately for system administrators, managing computers involves considerably less risk to life and limb than flying airplanes. Unfortunately, this makes roles much less clear- cut. Calling people "users" encourages one extreme, where the computer belongs to the system administrator and everybody else serves as a source of stupid user stories. Calling people "customers" encourages the other extreme, where the computer belongs to the people who use it, and everybody else serves as a source of fascist administrator stories. In truth, the computer belongs to whoever bought it, and we're all in this mess together.

"What one quality do you value most in a System Administrator?"

by Paul Moriarty
cisco Systems, Inc.
<pmm@cisco.com>

[I was talking with people outside the main ballroom at LISA VI this week when I was asked about the best qualities of a Systems Administrator. I thought about this, and decided to pose it as a question for the newsletter, since it is a topic of interest to us all. Paul Moriarty submitted the following sessions. – SAGE Editor]

Since making the transition to the management side of systems administration, I have had the opportunity to interview many people as potential members of the Engineering Computer Services team at cisco Systems, Inc. In addition to the typical laundry list of technical skills, the two skills that I value most in a potential candidate are articulateness and a strong desire to work closely with the user community.

As systems administrators, our most visible interactions from the perspective of our customers (the user community) are either those where we must interface with them directly (i.e., solving their specific problem or answering a question) or those where a resource upon which they depend has suddenly become unavailable. The key to the success of an organization lies in how well the customers perceive that you interact in these situations.

The engineering user community at cisco comprises people with a wide variety of technical expertise, ranging from the extremely knowledgeable to those who only wish to use a computer to get their job done and couldn't care less about the underlying operating system as long as it doesn't get in their way. The successful systems administrator must understand these differences and be able to adapt his/her interactions in such a way as to neither offend the technical user by responding too simply nor overwhelm and baffle the novice with too much underlying detail. Responses that are clearly and effectively expressed will not only leave the user with an answer to their question, it will also make them feel that you truly understand them and what they are trying to do. This fosters a sense that you are a member of their team as opposed to simply an answering service of some sort.

Every user knows what they want their computing environment to do for them and it is important for us as systems administrators to ensure that they get the most productive environment that we can provide. However, the challenge lies in the fact that the users often cannot express their desires in a way that is easy for us to understand. It is not up to them to figure out how to communicate this effectively to you; they have tasks and commitments that already fill their day. It is up to you to develop an understanding of how they use the computing environment and devise ways to maximize their use. This can only be accomplished by talking with the user community and providing them with a forum where they can explain to you just what it is they do and how they use computers to do their jobs. The successful systems administrator will proactively establish dialogues with his/her customers and not merely try to deduce what it is they do from fixing their problems when they occur.

Computers don't always work correctly and most users understand and accept this. However, when the machines are not working properly, it is imperative that we let them know that something is wrong and that we are trying to remedy it. It is comforting and reassuring to them that the problem did not magically go away (and will likely come back) – yet I have seen many instances where a systems administrator will identify and fix a problem but not tell anybody about it. This communication is especially important when the problem is transient in nature and difficult to troubleshoot. Update the user community regularly on what you are doing to fix the problem, even if it means telling them that you haven't made any significant progress. It is surprising to see just how understanding and patient they will be if they know that you haven't forgotten about it (and if you fail to update them regularly, I can assure you that this is exactly what they will assume).

For many organizations, the days when a service organization could exist solely on its service metrics are gone. To be a vital part of the organization, we must add value as well. From the organization's perspective, the only way that we as systems administrators add value is if our customers perceive us in that way. Thus, in order to be successful systems administrators we must not only be technically competent, we must understand our customer's needs and be able to articulately interact with them on their respective levels. The best way to accomplish this is to work closely with them, identifying their problems rather than acting as a background process, quietly fixing things or waiting for them to come to us with a problem or question before interacting with them.

An Update of UNIX-Related Standards Activities

by Stephen Walli
Report Editor < *stephe@mks.com*>
USENIX Standards Watchdog Committee

You are in a Maze of Twisty Profiles — All Different

[Warning — Profiles are poorly understood, ill-defined specifications that are being drafted as full standards in various corners of the standards community. If at first the article seems twisty and convoluted, that is because the topic is twisty and convoluted, mired in a lot of historical context. The article presents the historical context for profiling activities, and the traps lying in wait for unsuspecting applications developers. It finishes with a few recommendations.]

Profiles are the latest confusion to appear on the open systems standards scene. They are supposed to define a view on one or more standards in a coherent way to fulfill a general need. This need may be something like: "a programming platform for general multi-user, multi-processing business applications" or maybe: "supercomputing applications typically require the following services." This seems reasonable. It also seems to feel right. So what happened?

In the Beginning ...

The POSIX.1 (ISO/IEC 9945-1:1990 == IEEE Std. 1003.1-1990) standard standing alone is not enough. By its own definition, it requires C language support. This can be either Common Usage C or Standard C (ISO/IEC 9899:1989 == ANSI X3.159-1989). These two standards together provide a reasonable programming environment. They are not complete; there are many things missing. To move the standard forward, things were left out that were too contentious at the time. It was better to have some kind of standard than none at all.

POSIX.1 also has optional functionality. Some of this functionality is called out by "Big-O" options, such as {NGROUPS_MAX}, {_POSIX_JOB_CONTROL}, or {_POSIX_CHOWN_RESTRICTED}. These are implementation level options, and a vendor could choose not to implement them and still be conforming. There are other named options, such as {_POSIX_NO_TRUNC}, and {_POSIX_SAVED_IDS}, which may or may not be implemented. A strictly conforming application should never count on such functionality being present.

Using this simple model, the National Institute of Standards and Technology (NIST) created the U.S. government procurement document, FIPS PUB 151-1. In it, NIST specified what options and limits must be supported from POSIX.1, and how the C language support should be done. The intent was to provide as functional a platform as possible by mandating as much of the POSIX.1 standard as possible, something upon which U.S. government applications developers could depend. Simple.

A long time ago, relatively speaking, X/Open was formed. It described a collection of specifications that all of its member vendor organizations would adhere to. Thereby it provided a Common Application Environment (CAE) for applications portability. This specification of a platforms functionality was written down in the X/Open Portability Guide (XPG). They have made a point of adopting POSIX standard interfaces where possible, moving away from the original SVID definitions. Perhaps not complete, but still relatively simple. Both FIPS and the X/Open XPG feel kind of like something you, as an applications developer, might want to point to when describing the environment you want.

Now let's move on to where things start getting messy. A number of things start happening in parallel, which means the confusion factor goes up exponentially. POSIX began doing some things. ISO was doing others. The industry consortia were doing something else. And remember, the industry consortia are the ones backed by vendor money, and have a stake in selling you their solution. Industry consortia == A vendor once removed.

POSIX

A few years ago, at the beginning of the Great Project Proliferation in POSIX, two projects began which would develop Applications Environment Profiles (AEPs) for Supercomputing (POSIX.10) and Transaction Processing (POSIX.11). The intent was to describe how to use POSIX in building applications in these two particular domains. In the last two years, two more AEP projects developed in POSIX, one for Real-time applications (POSIX.13) and one for Multi-processor applications (POSIX.14).

These last two are illustrative of many of the problems encountered. The POSIX.4 (Real-time) and POSIX.4a (Threads) standards will become addendums to the POSIX.1 base standard. All function interfaces defined by POSIX.4 and POSIX.4a will need to be provided by future implementations of POSIX.1, although they may be just stubs returning ENOSUPPORT (or some such), if the implementation does not support the added functionality. This functionality will be called out by named options. Hmmmm. Getting a little muddy.

POSIX.13 and POSIX.14 would hopefully define an applications domain that must be provided by an implementation for the appropriate class of applications. By pointing at the appropriate base standards and choosing options, we can clearly define the requirements of a class of real-time or multi-processing applications. It is unclear whether the base standards are POSIX.1, POSIX.4, and POSIX.4a, or some future, as yet completed ISO/IEC 9945-1:2001.

That's perplexing enough. Now consider the following. POSIX.6 (Security), POSIX.8 (Transparent File Access), POSIX.12 (Protocol Independent Interfaces), POSIX.15 (Batch), and POSIX.17 (Directory Services) functions will all be grafted onto POSIX.1, with options, as they are approved. All of these base API standards, some of which are nothing more than option-labelled "diffs" to POSIX.1 (i.e., POSIX.8), will somehow be fit together into one BIG book. (And people thought POSIX.2 was big!)

Remember, all of the function interfaces will need to be provided by an implementation, even if only as stubs because the "option" is not provided by the implementation. A portable application will spend all of its start-up time querying sysconf() to determine if the underlying support is present. Profiles, which strategic management believes will provide some wonderful shorthand notation to discuss procurement packages with vendors, will do nothing for the applications developers actually writing applications.

Application Environment Profiles

I made reference to AEPs defining an environment that must be provided by an implementation to support an application domain. This is another source of confusion. Are we specifying an application domain where the implementation supports far more? It likely does anyway, but in a non-standard fashion. Or are we specifying a "platform" environment, so it provides a broad base of functionality typically required by an application domain. I believe this ambiguity lies at the heart of the "sub-setting" problem between POSIX.1 and POSIX.13.

The "sub-setting" argument arises because the real-time AEP (POSIX.13) wants the ability to call out parts of POSIX.1 as options, e.g. the file system. Some people feel this is a horrible idea, since POSIX.1 specifies a good general purpose base upon which to build applications. The profile specifiers, however, don't need the rest of the standard to describe their application domain. This has been a constant source of argument and confusion in the POSIX world. What can a profile point to, and how?

And then there are other specifications, outside of POSIX and TCOS, that would be obvious to include in certain application domain profiles. The POSIX.14 Multi-processing AEP would like to point to the X3 parallel language extensions work. The IEEE has no problem with pointing to other specifications, even incomplete early drafts such is the case here. It might even be an algorithm in a textbook.

If the point is to define a standards based environment, why would anyone want a profile standard to point to an indeterminate draft of a standard which is very unstable, even once it is mature enough to ballot? De facto specifications from vendors and vendor consortia (such as PostScript or OSF/Motif) are more stable than this!

ISO, on the other hand, has very strict rules about what can be pointed to in a profile. This leads us to another fine source of information and confusion. Let's look at ISO's contribution.

ISO and the OSI Stack

ISO has a little more experience with profiles, or maybe one should say longer experience. If I understand things correctly [salt warning]:

> The ISO SC21 working groups defined the now famous seven layer stack. This was an anticipatory model, telling us how things should/would be done in the future, rather than one cluttered by implementations.
>
> Vendors were somewhat horrified when governments started leaning in this well defined, robust direction. They still wanted to be able to play in the lucrative government sandbox. They started demonstrating how, if you interpret things in one light, their product fits the model here, or really fulfills these two layers together over there, and so on. The stack mutated a little.

> The wonderful situation arose, that it was now possible to draw an entire path through the stack, top to bottom, which wouldn't communicate with another line through the stack. People even gave this a name: *conforming incompatible implementations.*
>
> The procurement agencies weren't too thrilled by this turn of events, and profiles were born. U.S. GOSIP (Government OSI Profile) specified a known implementable path through the maze, and they used this for procurement specifications to ensure that one government OSI installation could communicate with another.

So we now have this concept of a profile. Choose a set of API and protocol specifications that will work together to form the OSI communications models. ISO even developed a document specifying how to do this. Technical Report 10000 (TR10000, or TR10K) defines a set of rules for how to define an OSI profile.

TR10K has very strict ideas about how OSI profiles are to be constructed, what they can point to and how. Profiles can only point to ISO standards (or other ISO profiles), if they are to have normative weight. Otherwise, the references are just informative.

Chaos Sets In ...

When the full complexity of this profiling problem began to appear, a number of different working groups began investigating the problem from various angles.

The profiling groups within POSIX were identifying problems as they built their drafts almost from the time they started meeting. The groups operated fairly autonomously, however, and initially never got together.

Appeals were made to the POSIX.0 working group for help. The POSIX.0 Guide to Open Systems Environments defines a model for how strategic management views standards being used, identifies many standards and where they fit into the model, and even has a couple of chapters on profiling activities and how they should be done. The POSIX.0 working group argued, however, that it was not responsible for setting profiling policy. Go figure.

After much pain and gnashing of teeth by the four POSIX profiling groups, a TCOS steering committee was formed to help solve the problems they had been having for about two years at this point. The group is made up officially of one member from each of the working groups defining profiles, and a few members of POSIX.0. Really.

The Profiling Steering Committee has been meeting for a year now. They were immediately lost in a forest of liaison points, and information gathering, trying to determine the state of profiling in the world. Now to my poor naive way of thinking, someone is not doing their job here. If the POSIX.0 members of the PSC did not already have all of the profiling documents that could be found, upon what is the profiling material in POSIX.0 based? Conversely, if they did have the profiling information and experience, then why has it taken a year to define a set of rules by which IEEE POSIX working groups should be defining profiles?

And even with a Profiling Steering Committee, they were so busy investigating what everyone else was doing, no one noticed that the POSIX.13 Real-time profiles were in ballot. Takes your breath away.

On the ISO front, things aren't much better. True to their anticipatory nature of late, a few different groups have been formed to investigate and comment upon something which doesn't yet exist. Technical Specification Group 1 (TSG1) has taken a kick at the cat.

The Special Group on Functional Specifications (SGFS) is also giving it a try. SGFS is attempting to take the TR10K document and modify it in a couple of places so as to make it applicable to the functional API standards, such as POSIX.

The European Workshop on Open Systems (EWOS), a CEN/CENELEC sponsored body, has set-up a working group to investigate a Common Application Environment (CAE). This work may be the most pertinent to date. There are people in this working group that have actually spent time attempting to specify real profiles in the commercial world. X/Open is involved in the work, lending its experience with defining specifications such as XPG3.

The EWOS work attempts to define a method of investigating the user requirements, building up the definitions and interfaces (informational rather than actual programming interfaces), and only as the very last step investigating how standards might be applied to the requirements model.

I was careful in the last paragraph to not say what type of profile was being defined. There is still a lot of discussion with respect to what is an application environment profile, versus a platform

environment profile, and there is even a new concept of a component profile. There is grey, and then there are shades of grey.

Wrap Up

```
> GET SENSE
 I SEE NO SENSE HERE.
> XYZZY
 XYZZY DOES NOT WORK HERE.
> DROP THE BIRD
```

Where do we go from here?

Profiles are poorly defined specifications (despite the many attempts at writing rules for their creation,) based for the most part on unstable documents in ballot, and there is no real experience at defining and implementing formal profiles in the open systems world. (The OSI profiles appear to be a well-defined, well-structured set of specifications which developed after there was experience with the stack — not before.)

Why do people feel that these documents should be standards? Why build castles on foundations of sand? We do NOT know what shape some of the key standards will be in until they finish ballot. Simply pointing to an interim draft of a document in ballot, even if the IEEE is willing to archive the draft, is silly.

The point is to specify an environment (application specific or not) which applications developers can count on. These environment specifications are supposed to be standards based. That's the whole point! The draft documents in ballot will change. Saying we'll modify the profile standard later, when the base documents complete ballot, is naive! People will have used it in procurements. Applications will have been written. What if the functionality in the base standard is gone? Or mutated so as to be useless to the profile? What's the rush for a useless standard?

There is the desire to specify how a set of standards can be used together to define a known environment to solve a set of applications portability problems. The simple extremes of a single standard profile (FIPS PUB 151-1), or a suite of specifications (X/Open's Portability Guide), have proven to be useful. It is useful for people to put down on paper the definition of a set of requirements for a particular applications domain, as is described by documents like the EWOS CAE working group's work. This should be done in a less formal way.

The Paul Masson Method should be applied. (We will define no standard before its time.) Make the profiling work either "guidelines" or "recommended practices" at the IEEE level, or "technical reports" at the ISO level. Until people have REAL experience putting these complex, subtle API definitions together with appropriate other functional and language standards, many of which are still in ballot or under definition, profiles should not be given the weight of full standards.

If you're an application developer, get involved. Follow the POSIX mailings. Determine what your national standards organization is doing. Ask questions, or make yourself heard through your institutional representatives to POSIX. (USENIX, EurOpen, Uniforum, DECUS, CUG, and SHARE are all represented in IEEE TCOS. X/Open, Unix International, and the OSF are also present.)

Before some strategic thinking manager above you makes the decision for you, you should fully appreciate the enormity of the confusion being unleashed on you as you quietly contemplate that POSIX.1 and ANSI C are probably useful after all.

Report on POSIX.0: Guide to Open Systems Environments

Kevin Lewis <klewis@gucci.enet.dec.com> reports on the April 8-12, 1992 meeting in Dallas, TX:

As I reported in the January Snitch for POSIX.0, the POSIX Guide to Open Systems Environments (OSE) is going to formal ballot (finally, as someone in the SEC said to me...). If you are in the TCOS Balloting Pool, you should have received an invitation to join the ballot group for this work.

The formal ballot will be on draft 15 which is being produced presently. The changes submitted by the group to produce this draft strictly addressed the mock ballot comments. The group agreed (after I placed a gag order on them) not to surface or open any issues that had previously been considered closed. This went a long way towards our moving through the comments and objections. (By the way, if you were one of our mock balloters, please be patient. I will be sending out a summary along with detailed ballot resolutions after I have completed the formal ballot package for IEEE.)

The formal ballot closure date has not yet been determined, although it appears that the end of August is the likely time frame. Our goal is to have a significant number of ballot responses into IEEE and the ballot coordinator (i.e. me) prior to the October meeting in Europe so we can use that time for ballot resolution, as well as share the results with our European counterparts.

Two issues remain as we move toward formal ballot. One is a rationale document. It became apparent during the April meeting that our attempts to use our Issues Log, along with the minutes and institutional memory of the core participants of the group, were lacking when it came to actually documenting our rationale for certain issues. So the July meeting has been dedicated to the task of developing and writing this document.

The other issue is that of public specifications. Our document is moving into the international formal standards forums. Looking out to the horizon, we expect there will be a lot of consternation over the group's choice to include informal specifications in the guide.

This will undoubtedly shape into another battle of significant proportions. The formal ballot period should offer the current warriors a chance to take a breather.

Report on POSIX.1: System Service API

Peter Collinson <pc@hillside.co.uk> reports on the April 8-12, 1992 meeting in Dallas, TX:

[*Ed. — Peter is the USENIX Institutional Representative to TCOS-SS, the IEEE group responsible for drafting the POSIX family of standards.*]

Overview

Theoretically, I spent most of the week in POSIX.1, the working group for the "original" system interface standard. It's still meeting because it has several extant projects:

POSIX.1LIS, programming language independent POSIX.1;

POSIX.16, the C binding to POSIX.1LIS;

POSIX.1a, the place where bug fixes and new features for POSIX.1 are being put while the language independence work is being done;

POSIX.18, the POSIX Environment Profile. It's a profile (or list of other standards) intended to describe something close to a complete UNIX system.

I tend only to attend the work for the first of these because I also go to many other steering committee meetings. Here's an idea of what happened in the bits that I managed to get to ....

The Report...

The ISO standards working group on POSIX, WG15, requires that the IEEE POSIX working groups produce a programming language independent version of the existing POSIX.1 standard (ISO 9945-1). This language independent specification (LIS) is referred to as POSIX.1LIS.

The POSIX.1 standard has been re-cast in two sections: the language independent specification and a C language binding (POSIX.16). The idea is that these two should ballot together, so that balloters can compare the original standard with the new pairing.

It's planned now that the two standards will go to ballot on July 7th. This has been made possible because:

> the documents are close to being ready, have been mock balloted and finally preened by the working group
>
> the Steering Committee on Conformance Testing (SCCT) has agreed that the documents do not need a completely new set of test methods written for them. They can use the already existing test methods for POSIX.1, contained in POSIX.3.1, which has nearly completed balloting.

Not needing new test methods is a great concession because it avoids the rule that insists on test methods being available for all new standards before they go to ballot. In my opinion, someone will need to find some funding to get the new test methods written. There is no enthusiasm for doing this in the working group. This is also the consensus of the group, when asked just that question.

What are test methods? That's a little hard to explain. Basically, they are terse English statements that assert facts about the standard. The idea is that these are easier to convert into programs that actually test the interfaces. Each assertion is classified as "testable" or "not testable," and whether or not it applies to optional behavior. It's a little more complex than this. Look at POSIX.3 (IEEE Std. 1003.3-1991), the standard for test methodologies for POSIX, for more information.

The current document drafts are based on the ordering in 9945-1. This is good because sections in all the documents refer to the same material. If you are looking at Section 3.2.1 in 9945-1:1990, then the same material will be found in the same numbered section in POSIX.1LIS and POSIX.16.

A small group of people who are close to the document – the editor (Hal Jesperson), the person

really running the LIS project (Paul Rabin from the OSF), and the chair of the POSIX.1 Working Group (Donn Terry) – have realised that this is POSITIVELY THE LAST CHANCE to change the ordering of the document. *[Ed. — the close() and open() functions are in different chapters of the standard, as an example.]*

Donn has come up with a potential re-ordering and this will be applied to the new documents. I was concerned that this would make balloting difficult, because we lose the ability to easily cross reference. The idea is to print a re-ordered version of 9945-1 (without rationale) to act as a balloter's aid.

The two new documents will also contain "other editorial changes." The adoption of the LIS has meant that the original text has been inspected very closely indeed, and has been found wanting in many places. It's often ambiguous with unclear wording. The text has been tightened up in these places. One of the tasks of the working group this week has been to examine a list of lines containing "may," "can," "cannot," "system defined," and some other words to ensure that they are all used consistently throughout the documents. Where ambiguities exist the wording has been repaired.

Now, you may argue that this will change the sense of the document, and it might. It will be up to the balloting group to worry about that. There are NO conscious changes.

New functionality and real bug fixes have been held over in POSIX.1a. There was no discussion on this during the week, because the person driving that, Roy McKean from X/Open, was unable to be in Dallas.

Report on POSIX.2: Shell and Utilities

David Rowley <david@mks.com> reports on the April 6-10 meeting in Dallas, TX:

Summary

Well, it looks like it's all over but the final formalities. New drafts of POSIX.2 and POSIX.2a incorporating minor editorial changes have been approved at the New Zealand meeting of ISO WG15 as Draft International Standards. They are Draft 12 of POSIX.2 and Draft 8.05 of POSIX.2a. Both POSIX.2 and POSIX.2a should go before the Standards Board in September for approval as full-use IEEE standards.

NIST is currently working on a new FIPS (Federal Information Processing Standard) for POSIX.2, expected in draft form by early Fall 1992.

POSIX.2b work progresses, incorporating symbolic link support within a number of utilities, and a new PAX archive format.

Test assertion work continues, with the POSIX.2 work adapting to an underwhelming mock ballot. POSIX.2a test assertion work is well under way, and appears to be easier than previously thought.

Background

A brief POSIX.2 project description:

POSIX.2 is the base standard dealing with the basic shell programming language and a set of utilities required for the portability of shell scripts. It excludes most features that might be considered interactive. POSIX.2 also standardizes command-line and function interfaces related to certain POSIX.2 utilities (e.g., popen(), regular expressions, etc.). This part of POSIX.2 , which was developed first, is sometimes known as "Dot 2 Classic."

POSIX.2a , the User Portability Extension or UPE , is a supplement to the base standard. It standardizes commands, such as vi, that might not appear in shell scripts, but are important enough that users must learn them on any real system. It is essentially an interactive standard, and will eventually be an optional chapter to a future draft of the base document. This approach allows the adoption of the UPE to trail Dot 2 Classic without delaying it.

Some utilities have both interactive and non-interactive features. In such cases, the UPE defines extensions from the base POSIX.2 utility. Features used both interactively and in scripts tend to be defined in the base standard.

POSIX.2b is a newly approved project which will cover extensions and new requests from other groups, such as a new file format for PAX and extensions for symbolic links.

Together, Dot 2 Classic and the UPE will make up the International Standards Organization's ISO 9945-2 – the second volume of the proposed ISO three-volume POSIX standard.

POSIX.2 Status

Draft 12 of POSIX.2 has been prepared, a minor revision of Draft 11.3 to take care of some editorial concerns fir ISO WG15. This new draft will form the final POSIX.2 standard, expected to be approved at the September meeting of the IEEE

Standards Board. Draft 12 has also been ap-proved by ISO WG15 as a Draft International Standard. It is certainly a help to implementors to have both the IEEE and ISO versions of the Shell and Utility standard coordinated in this manner.

POSIX.2a Status

In a similar fashion to POSIX.2 Classic, a minor revision of POSIX.2a has been prepared to address some minor ISO editorial concerns. Draft 8.05 (so named to reflect the extent of the changes) will form the final POSIX.2a standard, and should also be approved at the September meeting of the IEEE Standards Board. This draft has also been approved by ISO as a Draft International Standard.

FIPS and Certification

Now that NIST is preparing a new FIPS for POSIX.2 and POSIX.2a, the issue of conformance testing and certification is rearing its contentious head once again. The problem is one of timing and organization. NIST of course wishes the certification suite to be based on the POSIX.3.2 test methods work. However, it has only just gone to mock ballot, and is still quite a distance from completion. The POSIX.2a test methods work has only recently started. In spite of this, NIST wishes to put forth a FIPS now in order to encourage the use of the standard within the US Government. Unfortunately no standard metric for gauging conformance will exist for some time. NIST's lack of money for test suite efforts is causing a number of vendors concern and frustration, causing other solutions to be investigated. If you would like up to date information on the current status of POSIX.2 conformance testing, please feel free to drop me a note.

PAX File Format

The new file format for PAX is progressing, but the group is still not completely convinced that the ISO 1001 tape format is the best technology to base the format upon. No alternatives have been put forth, so the group will likely continue along the current path until someone makes a counter-proposal.

One issue decided at the Dallas meeting was the codeset to be used within the archive to represent filenames. The 16-bit plane of Unicode/ ISO 10646 (UCS2) has been selected as a good reference set of glyphs which should suit the needs of the vast majority of users. A step up to UCS4, the 32-bit version, will be planned for in the format. Gary Miller (IBM), POSIX internationalization and codeset guru, has given his blessing to the approach.

Test Methods

The POSIX.3.2 Test Methods for POSIX.2 mock ballot did not go well. Hardly any comments were received, so the group spent the Dallas meeting in small groups, one group working on creating ballot objections, and another on ballot resolutions. This isn't how it's supposed to work, folks. It is critical that the test methods work has the same level of broad-based input that the POSIX.2 standard enjoyed. Although the skill set required to effectively ballot the document is specialized and rare, the effort needs as much input as possible.

The document will go out of mock ballot for a while until a plan to get reasonable feedback has been formulated.

Work on the POSIX.2a test methods also progressed. The earlier fears of the difficulty of creating assertions for the interactive commands (*vi*, *talk*, etc.) have proven to be largely unfounded. However, turning the assertions into a test suite may still be a challenge.

Report on POSIX.3: POSIX Test Methods and Conformance

Andrew Twigger <att@root.co.uk> reports on the April 6-10, 1992 meeting in Dallas, TX:

SCCT Matters

Once again, the Steering Committee for Conformance Testing (SCCT) met for three sessions during the week. During the first session, Roger Martin (Chair) announced that three new members had been invited to join the SCCT. These are Jerry Powell (IBM), Stephe Walli (MKS) and Dan Hegerty (US Navy). The four remaining members of the SCCT (Roger Martin, Lowell Johnson, Andrew Twigger, and Bruce Weiner) were appointed for a further two year period.

The SCCT realised that unless it became more pro-active in encouraging the POSIX working groups to meet their test method development plans, the groups would not complete this work item. There had been a marked drop in the requests to the SCCT for test method consultancy from the working groups. It was believed that, in several cases, test method development was being sidelined while other issues were advanced. The SCCT decided that it needed to monitor progress more regularly and to advise the Project Management Committee in cases where slippage became evident.

The SCCT also became involved in discussions about the production of test methods for language independent specifications (LIS). *[Ed –*

Don't groan people. This stuff has real value.] This was discussed in the context of the POSIX.1 LIS. The thinking goes as follows:

> Language Independent Specifications are useful. They provide the functional specification upon which programming language syntax is layered in the language's most natural form. The intent is to allow different languages to bind as easily as possible.
>
> Real implementations support the functionality described in the language independent functional specification through language bindings. Real implementations are only valid through real languages, and can only be tested using real languages.
>
> In the same way that the functionality behind each language binding is the same, the test assertions are for the most part functional test assertions. There are additional syntax related assertions for each language, but a large percentage are functional assertions.
>
> By expressing the assertions as functional assertions written to the LIS standard, real test cases in different languages can be written. *[Ed — think about the problem of verifying a POSIX.5 (Ada) run-time implementation.]*

The initial target for LI test assertions was the revised POSIX.1 LIS, which is expected to enter ballot in the next quarter. The SCCT decided that they would accept POSIX.1LIS entering ballot with a reference to the current POSIX.3.1 C language binding assertion set, but that LI test assertions would be needed before ballot could complete.

At the moment there seems to be little interest in producing the LI test assertions — the task was described as a further layer of boredom on top of an already boring task! However, the SCCT believe that there is considerable value to those working groups who are amending POSIX.1 to develop a set of LI test assertions and this really needs a base set of assertions from POSIX.1 LIS.

Test Methods for POSIX.1

POSIX.3.1 is the test methods document for POSIX.1, the base operating system service interface. During the meeting the technical reviewers worked to resolve the remaining objections against draft 13 of this standard. It is believed that all of the outstanding objections have now been dealt with, and that the document is ready for a final recirculation ballot. It is hoped that this will be completed by the end of June and the document forwarded to the standards board shortly afterwards.

Test Methods for POSIX.2

The POSIX.3 and POSIX.2 working groups met jointly for most of the week with the available members from each of the groups starting to review the current draft document. This exercise caused many of the members of the group to realise how many areas still needed to be addressed, and at the end of the week a plan was put together to provide enough input to the technical editor to allow a much more complete draft to be produced.

Concurrent with this task, a few members of the POSIX.2 working group continued with the specification of test methods for POSIX.2a (UPE). Most of the work on the simpler utilities was completed, but the larger utilities still need to be tackled.

Report on POSIX.5: Ada Binding to POSIX.1

Del Swanson <dswanson@email.sp.unisys.com> reports on the April 8-12, 1992 meeting in Dallas, TX:

The POSIX.5 group has been working to produce Ada language bindings to POSIX standards. So far, we have been concentrating on the POSIX.1 standard and the Real-time Extensions standards being developed by POSIX.4. There are informal plans to prepare a project request (PAR) to develop an Ada binding to POSIX.2 as well.

The big excitement at the Dallas meeting was that Draft 8 had been produced in a short time, fixing minor problems in Draft 7, and was sent out for a fast recirculation. This draft was overwhelmingly approved, and Draft 9, encompassing a few editorial changes, is being submitted to the Standard Review Board for its final approval as an IEEE standard. *[Ed. – Del informs me that POSIX.5 has been approved as an IEEE Standard by the Standards Board on June 18. Congratulations to all who worked on and balloted the document!]*

Meanwhile, the group proceeded blithely along with its new task, to develop an Ada binding to the Real-time extensions being balloted from POSIX.4. Three position papers had been prepared, and were presented to the group, on the relationship of Ada runtime library functionality and the Real-time extensions. The issues were outlined in the report of the last meeting.

The group was fortunate to be presented with a draft thin binding to POSIX.4, which had been prepared at Florida State University under con-

tract to the U.S. Army. The group divided up the document, and individuals presented analyses to the group. The task for the POSIX.4 Ada binding group appears to be a cooperative effort with FSU, which should speed the process significantly.

Everyone agreed that the binding to POSIX.4 will be relatively straightforward. The POSIX.4a (Threads) binding, however, will have more significant problems.

Currently we are proceeding with the Ada bindings to the Real-time extensions in the same manner used for the POSIX.1 binding, i.e., working from the C language interface. By TCOS fiat, the binding will ultimately be to the Language Independent Specification of the POSIX.4 documents. The hitch is that the Real-time extensions group is just recently moving beyond its initial experiments with LIS, done in early 1990. The pieces are all finally in place, with an LIS of POSIX.1 and its C-binding (POSIX.16), to start the work seriously. At the April session, there was significant interaction between the two groups, to try to make the transition smoother.

Two issues in particular were addressed. First, the POSIX.5 working group composed a list of elements of the C binding which we thought particularly needed to be made language neutral, and discussed them with the Real-time group. Second, since it was agreed that ideally the names of the LIS should be reflected in all the language bindings, we provided to POSIX.4 a list of identifiers which seemed appropriate for the functions.

We have also supplied to POSIX.4 a draft of the thin Ada binding to what we have projected as an LIS. The hope is that seeing the results of a binding to an LIS will provide some guidance for the development of one.

We are expecting that within a couple of more drafts the current thin binding to POSIX.4 will be in good condition. We are meanwhile dividing up the responsibilities to start on sections of POSIX.4a and POSIX.4b. It is still a bit early to project a realistic date for beginning balloting.

Report on POSIX.17- Directory Services API

Mark Hazzard <markh@rsvl.unisys.com> reports on the April 6 - 10, 1992 meeting in Dallas, TX:

Summary

Draft 3.0 of POSIX.17 began IEEE ballot on April 7th and finished the first round of balloting May 19th with 84% of the ballot group responding. We completed sending responses to all who participated in the Mock Ballot of Draft 2.0.

The group formed a ballot resolution team, and dealt with the "Which track to ISO?" issue. Splitting/re-casting our Project Authorization Request (PAR) was a hot topic. We're following a PMC recommendation to separate the Directory Services API work (which is in ballot) from the POSIX name space issue which hasn't received much attention.

Introduction

The POSIX.17 group has defined and is balloting a user to directory API (e.g., API to an X.500 DUA - Directory User Agent). We used APIA — X/Open's XDS specification as a basis for work. XDS is an object oriented interface and requires a companion specification (XOM) for object management.

XOM is a stand-alone specification with general applicability beyond the API to directory services. It is used by IEEE P1224.1 (X.400 API) and is being standardized by the P1224 working group.

The current POSIX.17 PAR has a two part scope. The first authorizes the group to work on an API to directory services. The second (and more contentious) part addresses the POSIX name space issue. The working group has discussed name space, but decided to focus on the API to directory.

POSIX.17 is one of five "networking" groups under TCOS, and comes under the purview of the Distributed Services Steering Committee (DSSC).

Status

The group finally completed all the written responses to the comments received from the Mock Ballot of Draft 2.0 of our document. If you responded, you should have a reply by now.

Draft 3.0 of POSIX.17 was distributed for IEEE ballot just prior to the Dallas meeting, and included all test methods and the language independent specification (LIS). The document grew from 234 pages in Draft 2.3 to 478 pgs in Draft 3.0 with the inclusion of all remaining test methods.

As of this writing, the 1st ballot is now officially closed, with 84% of the Ballot Group returning ballots.

Once again, we met with POSIX.12 (Protocol Independent Interfaces) in joint session and discussed their requirements on directory services. The white paper produced by POSIX.17 was used as a basis for moving ahead on requirements. (The white paper was the result of an action taken

in Irvine to document agreements, assumptions, issues, options and proposed actions.)

The meeting was quite productive and resulted in an understanding on how to progress the work. POSIX.17 took an action to assist the POSIX.12 group with writing an annex mapping a simplified, more focused interface to the POSIX.17 API.

Some POSIX.17 members met with P1224 to process the comments/objections raised during the initial round of balloting of the object management specification.
The PMC recommended in January that the POSIX.17 project request (PAR) be split into two separate projects, one for the Directory Services API work (which is in ballot) and the other for the POSIX name space issue which hasn't received much attention.

Name space conjures up many different things for different audiences. Some folks see the issue as a language issue, dealing with function prefixes and the like. The working group sees the issue as one in which objects are uniquely named in a global context, i.e. beyond a single kernel. If we use the process id as an example, we find that the 5-digit positive integer used as the name for a process within most kernels doesn't scale too well globally. If I want to have a utility that determines the status of all my processes, even those on other kernels, I have to somehow extend the name space.

There was a spirited debate as to whether or not a second PAR was needed for name space work, in that the issues could be resolved by some other mechanism in the TCOS realm. Neither POSIX.17 nor the System Interface Coordination Committee (SICC) believe that POSIX.17 owns the "C" name space issue. A white paper will be produced summarizing the name space issue and the work to date. Stay tuned ...

The road to ISO

The group spent much time debating how to progress the POSIX.17 API work for ISO standardization. The central point of contention was a proposal to remove the POSIX.17 API from ISO 9945-1 to join P1224/P1224.1 in a to-be-determined track in ISO. [*Ed. — ISO/IEC 9945-1 is the ISO name for IEEE 1003.1, or POSIX.1. All other system interfaces, such as POSIX.4 real-time and POSIX.6 security, are supposed to be integrated to 9945-1 in future ammendments.*]

The rationale given was that since the POSIX.17 work is dependent on P1224 and all three documents share the same style of interface and roots, they should all be progressed to ISO within the same Working Group. Since P1224 and P1224.1 aren't part of (and won't be part of) 9945-1, POSIX.17 should be pulled out of 9945-1 and progressed with the other two documents.

There is a risk that ISO SC22/WG15 (the ISO POSIX SubCommittee 22 Working Group) will not accept a work item for an API to directory services outside of 9945-1. The implication is that a new SC22 working group (or one from SC21 or SC18) may be required for this work, with all the associated start-up overhead. All this could delay the work and subsequently jeopardize its completion as an ISO standard.

Taking the work from 9945-1 also breaks the link requiring a distributed POSIX system to include an API to directory services. At least one other distributed services working group (POSIX.12) was concerned about this as well.

Arguments against the non-9945-1 track to ISO resulted in a compromise that will (hopefully) allow us to retain the reference to the POSIX.17 work in a work item for 9945-1. The work item could revert to a pointer to the work being done outside of 9945-1 (if that comes about) and also serve as a place holder for our work within SC22 WG15 if another track couldn't be found.

A resolution was prepared for the SEC, proposing that the SEC authorize POSIX.17 to take several actions relating to the mechanics of progressing our document through the IEEE ballot process and on to ISO. After some initial tough sledding late Thursday night, (my Minnesota roots showing), the SEC accepted all the time critical aspects of the resolution, deferring the rest until Chicago.

In Closing ...

Once again, there are quite a few homework assignments between meetings. The ballot resolution process begins. Look for a white paper rationalizing the directory services API work with the name space issue. We also need to submit a New Project proposal for progressing the POSIX.17 to ISO within SC22.

The group will meet next time in Chicago, concentrating on Ballot resolution and name space issues. We plan to meet in Utrecht and possibly for a few days in Reading, UK, to complete the work for our first (and hopefully final) ballot recirculation.

Report on P1224: X.400 API

Steve Trus <trus@duke.ncsl.nist.gov> reports on the April 8-12, 1992 meeting in Dallas, TX:

Summary

P1224 is the Object Management API, based on X/Open's Object Management specification (XOM). It is used by POSIX.17 (Directory Services API) and the P1224.1 document. P1224.1 is the X.400 API.

P1224 spent a productive meeting in Dallas, and we are very near the completion of the standardization of the P1224 and P1224.1 documents.

At the Dallas meeting we:

1. discussed our goals for the International Standardization of the IEEE Networking APIs,
2. planned future work for the P1224 group,
3. presented the status of the IEEE balloting of P1224,
4. presented the status of the IEEE balloting of P1224.1,
5. planned the recirculation of the P1224 document, and
6. resolved ballot objections and reviewed ballot comments for the P1224 document.

International Standardization of the networking APIs

We discussed options for the International Standardization of the networking APIs. The goals of the P1224 group are to have our work standardized with minimal changes in JTC1, and to have the X.400 and the POSIX.17 Directory Services APIs standardized in the same JTC1 Subcommittee.

P1224 Working Group Future Plans

Plans for standardizing future X.400 related APIs were discussed. The X.400 API Association and X/Open will have stable base documents for a P7 and an EDI API by the end of 1992. Tentatively, we would like to begin converting these documents into IEEE standards at the January 1993 meeting.

P1224 Status

Balloting of the P1224 document began January 1, 1992, and ended January 31. The ballot group consists of 73 members. The P1224 ballot closed with 87% of the ballots returned, and 75% of the eligible voters approved the document. The test methods for P1224 will be included in the first recirculation of the document. (Balloting cannot complete until the test methods are balloted.)

The group spent two days resolving ballot objections and reviewing ballot comments for the P1224 document. The technical editor will incorporate the changes and the test methods into the document.

We agreed to limit the recirculation objections and comments to changes to draft 4 of the P1224 document and test methods. Recirculation begins May 17, 1992 and it will end June 19.

P1224.1 Status

The P1224.1 balloting period will begin May 6, 1992 and will end June 5. There are 49 people in this balloting group. The test methods will be included in the initial ballot of the P1224.1 document.

Iain Devine, the P1224 technical editor will be the ballot resolution reviewer, assisted in technical matters by members of the X.400 API Association and X/Open.

In Closing ...

The progress of the P1224 working group is very good. We hope to have the P1224 and P1224.1 standards complete by the end of 1992. The primary function of the July and October meetings will be P1224 and P1224.1 ballot resolution.

Report on The IEEE Standards Board

An Anonymous Friend of USENIX reports on the March 1992 meeting.

[Ed — Anyone wishing to send comments to the report writer may do so through me.]

The March 92 meeting of the IEEE Standards Board contained some very interesting action on the GUI project authorization requests (PARs), more forward (or backward) movement on other TCOS (POSIX) PARs, and broad developments in the IT (Information Technology) field in general.

X3/JTC1 U.S. TAG merger

One of the big discussion items on last year's Board agendas was the proposed merger of X3 with the ISO/IEC JTC1 U.S. TAG (U.S. Technical Advisory Group for ISO/IEC Joint Technical Committee 1). Considered to be an administrative advantage for both organizations, and a means to speed up the possible internationalization of U.S. IT standards, there was concern within the IEEE as to how its standards groups would be represented in the international arena.

At the March 92 meeting, it was reported that this merger in its current form did not achieve approval through a consensus vote. It is expected that work will be done on the proposed merger and that it will reappear in a future letter ballot of the JTC1 U.S. TAG (the IEEE is a member of this TAG).

IT Standards Funding

Another motion that the Board discussed is a proposal from ANSI to charge "participants" from the U.S. in international standardization efforts a fee to cover the administrative costs of handling the international IT standards activities. Remember, ANSI is the member body representing the U.S. in ISO. (For the IEC, its a group called the U.S. National Committee, or U.S.N.C). The Board created an ad-hoc committee to address this. This committee held its first meeting during Board week and explored guidelines and processes to come up with a response to this request. Gary Robinson and John Rankine are the Computer Society representatives on this committee.

Cray Users Group

Cray Users Group requested Organizational Representative status at this Board meeting and, with the recommendation of the TCOS chair, was approved as an OR by the Board.

TCOS Inside Track on RevCom

One of the TCOS vice-chairs, Lorraine Kevra, is now on the IEEE Standards Board Review Committee (RevCom), which gives recommendations for final approval of standards to the Board. Lorraine will be able to bring first-hand experience with this back to TCOS and, hopefully, be able to explain the convoluted existence of POSIX to RevCom!!

The next IEEE Standards Board meeting will be June 16-18 in San Juan, Puerto Rico. The following meeting will be September 15-17 in New York City. The deadline for submission of PARs and standards for the September meeting is August 7.

NesCom (New Standards Committee Activity)

NesCom set a new record for work, with over 75 PARs on their agenda. The meeting went for over six hours. If you could ever imagine a completely exhausted committee, NesCom was it at the end of their day!

Approved New TCOS Projects

P1003.7.1 (OS) Standard for Information Technology--Portable Operating System Interface (POSIX)--Part 3: System Administration--Amendment: Print Administration

P1003.7.2 (OS) Standard for Information Technology--Portable Operating System Interface (POSIX)--Part 3: System Administration--Amendment: Software Administration

P1003.16a (OS) Standard for Information Technology--POSIX C Language Interfaces --Part 1: Binding for System Application Program Interface (API)-- Amendment 1: System API Extensions

Approved TCOS PARs to Revise Existing Standards:

P1003.2b (OS) Standard for Information Technology--Portable Operating System Interface (POSIX)--Part 2: Shell and Utilities

Approved TCOS Revised PARs:

P1003.1 (OS) Standard for Information Technology--Portable Operating System Interface (POSIX)--Part 1: System Application Program Interface (API) [Language Independent]

P1003.1a (OS) Standard for Information Technology--Portable Operating System Interface (POSIX)--Part 1: System Application Program Interface (API) [Language Independent]--Amendment 1: System API Extensions

P1003.7 (OS) Standard for Information Technology—Portable Operating System Interface (POSIX)--Part 3: System Administration Interface

P1003.16 (OS) Standard for Information Technology—POSIX C Language Interfaces--Part 1: Binding for System Application Program Interface (API)

P1201.1 (OS) Standard for Information Technology--Uniform Application Program Interface--Graphical User Interfaces

TCOS PARS for Which Approval Was Withheld

There was one unapproved TCOS PAR:

P1003.19 (OS) Standard for Information Technology--POSIX Fortran 90 Language Interfaces--Part 1: Binding for System Application Program Interface (API)

Ths project was not approved because the scope did not clearly imply that this standard would not change the existing language standard produced in X3. The amended PAR was not filed in time for the June Board meeting; let's hope for September!

PARs Removed From the NesCom Agenda:

P1295.1 (SCC) Standard for Information Technology--X Window System--Modular Toolkit

P1295.2 (SCC) Standard for Information Technol -

ogy--X Window System--Open Toolkit Environment

These PARs (the GUI PARs) were removed from the NesCom agenda per NesCom member John Horch because the Sponsor-approved wording changes were not available in time for the NesCom meeting. They will be reintroduced at the June Board meeting.

[Ed. — The Standards Advisory Board has apparently withdrawn the offer of hosting the sponsorship of the GUI PARs from TCOS. The supporters of the Open Toolkit Environment and Modular Toolkit PARs (Motif and Open Look by different names), have convinced the SAB their destiny lies elsewhere.

This is despite the fact that they fall within TCOS's scope statement, and that the P1201 windowing PARs lie within TCOS.]

Report on ANSI X3J11 and ISO/IEC SC22/WG14: C Language

Michael Meissner <meissner@osf.org> reports on the May 10-15, 1992 meeting in Salt Lake City, UT:

On May 10-12 of 1992, I attended the ANSI X3J11.1 meeting, and on May 13-15 of 1992, I attended the combined ANSI X3J11 and ISO WG14 meetings.

For those people who aren't aware of how the various committees interact, and what their charter is, here is a thumbnail sketch. In the beginning was the ANSI X3J11 committee, which is the American committee chartered to produce a C standard. The first C standard was approved in December of 1989, and is available as X3.159-1989. The X3J11 committee is now doing interpretations, where they have to answer queries about the standard, but cannot change it.

Around 1988, the ISO WG14 committee was formed to lead the American C standard through as an international standard. In ISO, each country gets one vote, and the USA votes through ANSI. After reformatting the standard and moving some sections around to meet ISO guidelines, the C standard was approved as an international standard, which is available as ISO/IEC:9899-1989(E).

At the time the standard was approved, there were three open issues raised by Japan, Denmark, and England, and there was approval to work on a normative addenda to address the problems. (These issues are covered later.)

Around 1989, some people started meeting to discuss numerical issues and the C language. The committee, originally called NCEG (Numerical C Extension Group), has since become X3J11.1, a subcommittee of X3J11. Their charter is to produce a technical report, which does not have the weight of a ANSI or ISO standard. I suspect many of the X3J11.1 features will be items to be considered for the next ANSI/ISO C standard. This committee is made up of various interested parties who care about floating point calculations.

X3J11.1

X3J11.1 met for the first three days, from May 10 through May 12.

I went to the floating point extensions subgroup on Sunday night. For the most part, this meeting was uncontroversial. The floating point extensions group had submitted their draft to a letter ballot which passed, and the meeting was used to address minor editorial changes and comments from the ballot. The draft contains the following items:

> New syntax for floating point constants, so that you can specify the exponent and mantissa in hexadecimal, rather than decimal.
>
> *Printf/scanf* `%a`/`%A` format specifiers to print the floating number in the new hexadecimal format.
>
> More math functions.
>
> Overloaded math functions — these functions are a step towards C++ style overloading: if the arguments are single precision, the calculation is done in single precision. Unlike C++, these are only required for the system functions and not the user functions.
>
> Requirements on exactly when Nan/Infinity/-0 is produced from the various match functions if the system uses IEEE 754/854 floating point. (Most systems these days use IEEE 754 format).
>
> Adding IEEE unordered comparisons (!>, etc.) which return true if either value is a Nan, instead of false.
>
> Adding floating point classification functions.
>
> Ways to get/set exception flags.
>
> Two new include files are added.

On Monday and Tuesday, I went to the normal X3J11.1 meetings. The following items were discussed:

The restricted type qualifier proposal had a successful letter ballot, and will go outside of X3J11.1 for review. This proposal is halfway between the current situation where the compiler can't fully vectorize, and noalias, which got shot down before the standard went out. It adds a new qualifier, restricted, which says that you promise that the given pointer is the only way a particular item is referenced. This will allow a function to take two restricted pointers, and to fully vectorize the accesses, because the compiler doesn't have to worry about overlap cases.

Automatic variables with variable dimensions were discussed, but no conclusion was reached. There are two proposals on the floor, one from Cray and the other from USL. The Cray proposal would require people to pass the bounds explicitly for arrays, and has problems in scoping if the bound is passed after the array. The USL proposal which is authored by Dennis Ritchie, would pass a "fat" pointer, which is a descriptor that contains the bounds as well as the pointer. The debate went on as to which was more in the spirit of C. I personally tend to favor the USL proposal.

Designated initializers will go out for a review. These allow a programmer to initialize a structure or array out of order. For example:

```
struct foo {
 int a, b;
} st = {
   .b = 1, .a = 2
};
int foo2[10] = { 1, [5] = 2, 3 };
```

(In the array example, element 6 is initialized to '3'). Gcc 2.0 has a similar feature, though the syntax is slightly different.

Compound literals will go out for a review. These allow a programmer to create an automatic (or static if at file scope) aggregate without having to give it a name. Gcc has this feature. For example:

```
foo (&(struct bar){ 1, 2 });
```

The floating point extensions draft mentioned earlier was approved to go out for a review. One item that will go in a cover letter is to warn people that the #pragmas specified may be changed into macros, since pragmas are not allowed inside macro expansions.

The complex arithmetic draft was not ready to be sent out for review at this time. The draft needs to be more fully specified for IEEE floating point with respect to Nans and Infinities. Also, there was concern that the complex functions be folded in with the overloaded functions (ie, having just sin instead of csin). Finally, some people feel that in addition to real, and complex types, there needs to be an imaginary type that has no real component, particularly in the case with Nans and Infinities.

There was some spirited discussion about extended integers and 64 bit machines. The 64-bit consortium (vendors who will be producing 64 bit CPUs) want the ANSI group to exactly specify what sizes short, int, long, etc. are in 64 bit environments. Given that ANSI committees typically take years to come down from the mountain, and the 64-bit consortium needs to deliver products soon, it was hopeless. Also, there are good reasons why the standard only gives minimums. The crux of the problem is that when you move to 64 bits, programs will break (just like they did in moving 16 bits to 32 bits, but there is more extant code in C now). No matter what you choose, you break somebody's cherished notations. One camp wants int, pointer size, and long to all be 64 bits, and there is no explicit 32 bit type. Another camp wants int to be 32 bits, and pointers/long to be 64 bits. Finally at least one person wanted int to be 64 bits and long to be 32 bits. The C committee roundly reviled any rule that broke the rule that sizeof (int) <= sizeof (long), but otherwise had no comments to send back to the 64-bit consortium. The array syntax subgroup met on Monday night. This group is charged with doing things to arrays, so that fast code can be generated on the vectorizers and/or massively parallel machines (essentially Cray vs. Thinking Machines).

The meeting quickly broke down into shouting matches and such. I felt that it made negative progress, to the point that the only positive vote was a "motherhood" vote on the group's charter. There was another array syntax subcommittee meeting on Tuesday night (and possibly Wednesday night also), but I declined to attend

NSI X3J11/ISO WG14

On Wednesday through Friday (May 13 - 15), the ANSI X3J11 and ISO WG14 met together. At times the meeting was run in ANSI X3J11 mode, and at other times it was in ISO WG14 mode. The primary objective for the ANSI part of the meeting was to answer questions about the standard. The primary objective of the ISO part of the meeting was to deal with the three proposed normative addendum.

The U.K. addendum is designed to tighten up the wording of the standard, but not to make any substantive changes. The goal of the Japanese addendum is to add additional wide character

functions and a new header in which to declare them. The Danish addendum provides alternatives to the ANSI trigraphs, while not using any of the national replacement characters from ISO 646.

The big news is that the ANSI C standard will soon be withdrawn and replaced with the ISO C standard, so that the standards remain synchronized. This means that chapter and verse quotations will soon change, due to paragraph renumbering required by ISO. Also, when the normative addenda come out, they will become part of the ANSI C standard, in addition to the ISO C standard.

Some of the decisions reached in talking about the Japanese addenda include:

> Wide character I/O functions can return errors if they can't translate multibyte <-> wide characters. Errno is set to `CEILSEQ` upon such an error.
>
> If a wide character value is >= 0 and <= UCHAR_MAX, then the single byte character classification functions (*isprint(), isspace()*, etc.) if true, implies that the wide version (*iswprint(), iswspace()*, etc.) is also true. If the single byte version is false, it does not imply that the wide version also returns false. This is to allow wide characters to fill up positions in the encoding that aren't valid single byte values.
>
> We voted against adding more support for mixing multibyte and wide character strings in the **printf()/*scanf()* family of functions. The proposal was for `%hs` to always mean multibyte characters in both *printf()* and *wprintf()*, `%ls` would always mean wide characters, and `%s` would mean either multibyte or wide characters, depending on whether the function was *printf()* or *wprintf()*.
>
> The new function *wcswcs()* (wide version of *strstr()*), got renamed to *wcsstr()*, since most people felt that the second 'str' represented substring.
>
> We voted not to reserve the wide stdio functions for a future standard to put in stdio.h (ie, you always have to include wchar.h to properly declare those functions).
>
> We voted that no illegal multibyte sequence will be emitted by the wide character output routines (including through `%S` or `%C` in *printf()*).
>
> We voted that only a single byte space terminates `scanf('' %S'')`, ie. not *iswspace()*, to allow for logically ungeting just a single byte.

The Danish digraph proposal was shot down (again). I suspect it may be for the last time, because more countries are concerned about delaying the rest of the addenda for this one small issue. Japan and the Netherlands both voiced this opinion for the first time at this meeting.

There will be letter ballots sent out on the various responses to interpretation requests. One letter ballot will cover all decisions in which there were no "no" votes at the committee, and one letter ballot will be sent out for each decision that had at least one "no" vote. It is hoped that the draft for the document of interpretation requests will be passed in the letter ballot, so it can be sent out for the next meeting (6 months from now).

ISO Monitor Report

ISO Monitor Report on the May 1992 ISO POSIX Meeting

by Stephen Walli
<stephe@mks.com>

Overview

The International Standards Organisation (ISO) and the International Electrotechnical Commission (IEC) jointly develop international standards for information technology. The family of IEEE standards known as POSIX are being brought forward as international standards.

The ISO view of this process is that the standards are being developed by a national body (U.S.) instead of the more traditional model of ISO working group development. (Similar national body development is going on for C++ in JTC1/SC22/WG21 which meets jointly with ANSI sponsored X3J16.) The IEEE forwards work through an ANSI sponsored Technical Advisory Group (TAG), to ISO/IEC JTC1/SC22/WG15. This frightfully long agglomeration of acronyms stands for ISO/IEC Joint Technical Committee 1 (JTC1), Subcommittee 22 (SC22) on Programming Languages, Working Group 15 (WG15) on POSIX.

WG15 (as we shall refer to it) helps guide the IEEE documents as they come forward as ISO standards. Direct development of the documents does not happen in WG15, but rather it acts as a focal point for international comment and much of the liaison work that is required to ensure that the IEEE documents will be able to stand as ISO standards.

The point of the process is to develop a single standard which does not diverge from the IEEE counterpart. The groups have succeeded to date, with the base operating system API embodied by IEEE Std 1003.1-1990 being identical to ISO/IEC 9945-1:1990 with the minor exception of the plain white ISO book cover. The IEEE Standards Press even produces the ISO book, and they do so on A4 paper no less!

The WG15 projects are organised into three standards: 9945-1 represents all of the operating system APIs, 9945-2 represents the shell and utilities, and 9945-3 will be the system administration functionality.

Currently, the IEEE POSIX.4 (Real-time), POSIX.6 (Security), and POSIX.8 (Transparent File Access) documents are all somewhere in the WG15 review-and-comment process. These documents will all be rolled (as programming language independent functional specifications) into 9945-1. POSIX.2 and POSIX.2a will become 9945-2 in the (relatively) near future. POSIX.7.1 (Printer Administration) is making its debut on the ISO WG15 scene this meeting in a very informal way, as the WG15 members were encouraged to join the initial mock ballot. This book will eventually become part of 9945-3.

The last thing worth mentioning before getting into the report of this meeting is the group itself. There were 21 attendees. (The IEEE typically has around 350 attendees.) This number is a little low, as we were meeting on the other side of the globe in New Zealand. These 21 people represented 9 countries (one country gets one vote.) Size of delegation is always fun to note. (Please see the table.)

Country	Count	IEEE
U.S.	4	4
Canada	4	2
England	2	2
Germany	1	1
France	1	-
Italy	1	-
Japan	1	-
Denmark	1	-
New Zealand	4	-
Officers	2	2
9	21	11

The officers are the convener (Jim Isaak, U.S.) and the project technical editor (Hal Jespersen, U.S.). The overlap is also interesting. Jim Isaak is both chair of the IEEE Technical Committee on Operating Systems – Standards Subcommittee (TCOS-SS), the group responsible for building the POSIX documents, as well as ISO WG15 convenor. Hal Jespersen is also TCOS-SS Vice Chair of Technical Editing, and chair of IEEE POSIX.2 (Shell and Utilities).

The other American delegates are all voting members of the TCOS-SS Sponsor Executive Committee as well, representing the Chair of IEEE POSIX.1, the Chair of the Steering Committee for Conformance Testing, the Uniforum Institutional Representative, and Vice-Chair of Logistics. One of the English delegates is Chair of POSIX.7 (System Administration). The German

delegate is Vice Chair of POSIX.6 (Security). One of the Canadians (the author) is the EurOpen Institutional Representative.

This overlap proves useful since the size of IEEE POSIX (ap 350 members) makes it almost impossible to completely overlap the WG15 and IEEE TCOS-SS meetings, as the C++ people do. There just aren't enough hours in a day for all the coordination meetings. The best that can be currently done is to run one WG15 meeting a year right beside an IEEE meeting. WG15 meets twice a year. TCOS-SS meets four times a year.

The next WG15 meeting will be in Reading, U.K., October 27-30, 1992, following the IEEE meeting in Utrecht, NL, October 19-23.

Enough of this didactic rambling. On to the report!

The Meeting

This meeting was held in Hamilton, New Zealand, as WG15 travelled to the far side of the globe in the hopes of encouraging future participation from New Zealand. Before everyone starts the "exotic locations" routine, let me point out it is 19 hours by plane for someone from the east coast of North America, with a brief (2 hour stop) in a transit lounge. Our accommodations were undergraduate (!) dormitories at the University of Waikato, who hosted the meeting. You remember undergrad dorms, a bed, a desk, a narrow aisle between them in which to dress, and the W.C. down the hall. The cafeteria (!!) food wasn't all that bad, but....

POSIX.2

One of the primary accomplishments of the week was the acceptance of POSIX.2 (Shell and Utilities) and the POSIX.2a (User Portability Extension) as a Draft International Standard (DIS). Through the hard work of Hal Jespersen, as chair of POSIX.2 and the project technical editor of both the ISO and IEEE working groups, WG15 was able to settle on a draft of the documents which met with everyone's approval.

The POSIX.2a User Portability Extension (UPE) is an amendment of the base POSIX.2 document. The two will be rolled together now.

With a little luck and optimism, the schedule should work something like this:

Summer, 1992 — Final recirculation of the two documents in the IEEE balloting group. This will be similar to the final editorial circulation of POSIX.1a as a reformatted IEEE Std. 1003.1-1988, just prior to becoming IEEE Std. 1003.1-1990 and ISO/IEC 9945-1:1990.

September, 1992 — the two documents come forward to the IEEE Standards Board for final approval as IEEE standards (IEEE Std. 1003.2-1992).

Fall, 1992 — The combined book (ap 1400 pages!) will be recirculated for one last ballot at the international level. This ballot changes 9945-2 from a DIS to a full International Standard (IS).

Because of its sheer size (volume?), there will still be ballot objections. There is just too much being covered to have people who are happy with all of it. There are still areas which have demonstrable problems. These can and will be fixed in future amendments. We are finally down to the wire for a document that because of the breadth of its coverage has been in ballot for four years. The community is finally going to get the companion standard to 9945-1 (POSIX.1) that it wants and needs.

LIS

One of the requirements placed on the IEEE working groups forwarding API documents as standards to ISO, was that they be forwarded as programming language independent functional specifications (LIS), with at least one language binding. The intent of this method is to allow other languages to bind to the functional specification in a manner most natural to the language, and not merely re-cast the original standard's programming language syntax into something in a new language. (No one wants to propagate the GKS API that demonstrated that one could write Fortran in any language.)

There is currently an LIS version of POSIX.1, with a C binding. This was built from the original C-based 1003.1-1990. (These documents are referred to as POSIX.1/LIS and POSIX.16.) They are about to go to IEEE ballot this Summer.

Originally, these two new documents were to be an exact mapping to 1003.1-1990. The organization of the original left a little to be desired. The open() function and the close() function are in different chapters. At the New Zealand meeting, WG15 voted to allow the POSIX.1/LIS and POSIX.16 technical editor to re-organize the work based upon a new organization agreed to by all.

Additionally, it was agreed that small bug fixes should be allowed to the documents. The timing of ballots is such that it could be a long time before another round of changes comes along to "fix" the POSIX.1 book.

A concern was raised that we are opening a nasty hole into which many things will find their way. Bug fixes and wording changes (based on interpretations) are small. New functionality is not. This is something that the balloting groups will have to watch out for. As help for the balloter, two things will be added to the balloting package.

A mini 1003.1-1990, without the rationale and annexes, and reorganised to the new sections, will be sent out to allow balloters to see how the LIS and C binding align with the C-based original.

A list of all changes for bug fixes will be sent to allow balloters to quickly locate material that has actually changed in content from the C-based original.

A request has been made by ISO SC22/WG11 (Language Bindings) to bring the IEEE TCOS-SS Guidelines document that describing how to build LIS and language bindings, forward as an ISO Technical Report. The new work item request will be brought forward in the Fall meeting.

Profiling Activities

POSIX profiling work is continuing to gain acceptance in the WG15 arena. Profiles are seen by some to be the way that all the open systems standards will be put together to form coherent working environments.

WG15 has created a Rapporteur Group for the Coordination of Profiling Activities (RGCPA) to handle activities relating to POSIX profiles within ISO. (Rapporteur groups are a essentially a formal special interest group within an ISO Working Group, which acts as an official point of coordination.) RGCPA has met twice now, once last Fall and again in January.

The terms of reference for the group were established at this meeting. The RGCPA's most important role will be as a liaison point for other profiling activities within the open systems world.

The European Workshop on Open Systems (EWOS) has done some good work in determining just how to build useful profiles. Luigi Bertuzzi, representing Italy at this WG15 meeting, has been involved in this work and presented it to WG15. The EWOS work involves a number of steps to help shape a functional profile from user requirements, applying standards only as the last step. It does not try to cram user requirements onto standards, nor make the mistake of assuming the standards represent user requirements.

The IEEE POSIX.0 (Guide to Open Systems Environments) also contains profile related work. This document is about to be balloted at the IEEE level. POSIX.0 is to be brought forward as an ISO technical report as well. This WG15 meeting was the beginning of that process.

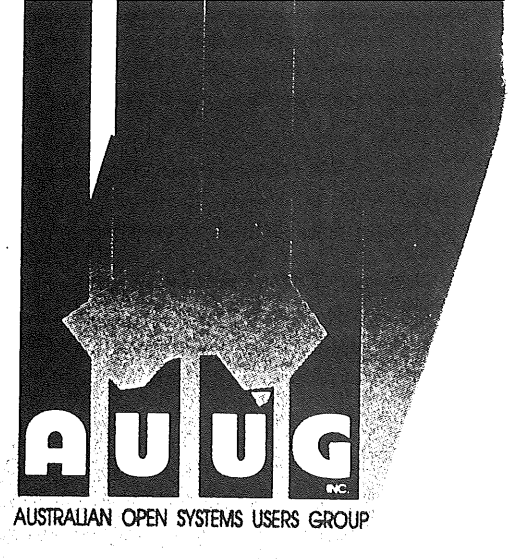
Internationalisation (i18n)

Internationalisation (i18n) is an obvious interest to an ISO standards body. WG15 created a rapporteur group on i18n for POSIX early on in its existence. WG20 is another SC22 (Programming Languages) working group which concerns itself with i18n issues with respect to programming languages in general. Keld Simenson (DK), as a member of both groups, acts as the liaison in both directions between the groups.

[One member quietly suggested we should really be concerned with intergalacticalisation. The two of us quickly coined the term "i20n". When we make first contact, remember, you heard it here first.]

WG15 forwarded a liaison statement to WG20 (Internationalisation). One of the important points of the statement was the recognition of the fact that while internationalising an application is a good thing to do, and a common portable method of doing so is a good thing to have, internationalising an application probably reduces its portability. One can very quickly add a lot of requirements to the portability of an application by internationalising it.

Frequently Asked Questions



??
??

This is a list of frequently asked questions about AUUG.
It is updated regularly and posted monthly to aus.auug.

INDEX TO QUESTIONS

(1) What is aus.auug?

aus.auug is an electronic newsgroup for discussions about AUUG and its activities.

(2) What is AUUG?

AUUG Inc. is the Australian UNIX(*) and Open System user group. It is a national body offering its members access to information on current and future UNIX and open systems technologies.

AUUG's aims as stated in its constitution are:

> To promote knowledge and understanding of Open Systems including but not restricted to the UNIX system, networking, graphics, user interfaces and programming and development environments, and related standards.

(3) How do I join AUUG?

There are membership forms at the end of this FAQ. Please print one, fill it in and return it to the AUUG Secretariat at:

AUUG Secretariat
PO Box 366
Kensington NSW 2021
Phone: (02) 332 4622
FAX: (02) 332 4066

(4) How many members does AUUG have?

AUUG currently has over 700 members, around 500 individuals and 250 organisations. AUUG membership has more than doubled in size in the past two years.

(5) Who are the members of AUUG?

A survey conducted in July 1992 revealed that the average AUUG member had a 4 year university degree and 5 years experience with UNIX. AUUG members are employed in Software Development, Government/Military or Education/Consulting. They are predominantly Executives/Senior Managers/MIS Directors and Systems Administrator/Programmers, with about an equal number in each category.

(6) What are the classes of membership?

AUUG has three classes of membership: ordinary, student and institutional. All members are entitled to one vote in ballots.

For organisations who do not wish to be a member but want to receive a copy of AUUGN, there is a newsletter subscription service. Contact the AUUG Secretariat for details.

(7) How much does membership cost?

Ordinary:	$78/year
Student:	$45/year
Institutional:	$325/year

(8) What is AUUGN?

AUUGN is AUUG's bi-monthly technical newsletter. It publishes papers and information of general interest to members. It contains details of local chapter activities and is how AUUG members stay up-to-date with AUUG events. Submissions to AUUGN can be made through the AUUGN Editor:

AUUGN Editor
PO Box 366
Kensington NSW 2033
Email: auugn@munnari.oz.au

(9) How do I get in touch with AUUG?

AUUG can be contacted through the Secretariat at:

AUUG Secretariat
PO Box 366
Kensington NSW 2021
Phone: (02) 332 4622
FAX: (02) 332 4066

There are also a number of e-mail aliases to facilitate contact with AUUG:

auug@munnari.oz.au	general inquiries
auugn@munnari.oz.au	newsletter correspondence (the AUUGN editor)
auugexec@munnari.oz.au	AUUG Management Committee

(10) Who runs AUUG?

AUUG is run by an elected management committee. Elections are held in May/June each year and all AUUG members are eligible to stand for election. The Management Committee consists of the President, Vice-President, Secretary, Treasurer and 5 general committee members. The term of office runs from 1 July to 30 June.

The current members of the Management Committee are:

President:	Phil McCrea	<phil@softway.oz.au>
Vice-President:	Glenn Huxtable	<glenn@cs.uwa.oz.au>
Secretary:	Peter Wishart	<pjw@lobo.canberra.edu.au>
Treasurer:	Frank Crawford	<frank@atom.ansto.gov.au>

Committee Members:

Rolf Jester	<rolf.jester@sno.mts.dec.com>
Chris Maltby	<chris@softway.oz.au>
John O'Brien	<john@wsa.oz.au>
Michael Paddon	<mwp@iconix.oz.au>
Greg Rose	<ggr@acci.com.au>

Elections are run by the Returning Officer and Assistant Returning Officer, who are elected each year along with other officers. The Returning Officer and Assistant Returning Officer are not members of the Management Committee.

Returning Officer:	Michael Tuke	<mjt@anl.oz.au>
Assistant:	vacant	

AUUG employs a Business Manager to run the day to day business of AUUG. The Business Manager reports to the AUUG Management Committee.

Business Manager:	Liz Fraumann	<eaf@softway.oz.au>

AUUG employs ACMS (Australian Convention Management Services) to provide Secretariat services for AUUG. ACMS provide membership services and support for Management Committee activities.

AUUG Secretariat:	Wael Foda	See contact details above.

AUUGN (AUUG's newsletter) is put together by a volunteer Editor who publishes AUUGN every two months. The AUUGN Editor is appointed by the Management Committee.

AUUGN Editor:	Jagoda Crawford	<jc@atom.ansto.gov.au>

AUUG is formally incorporated in the state of Victoria. The incorporation rules require that a resident of Victoria be the public officer of the association.

Public Officer: Robert Elz <kre@munnari.oz.au>

(11) What do I get for my membership?

AUUG membership provides you with:

AUUGN - AUUG's technical bimonthly newsletter. See details above.

Discounted right to use AARNet for mail - see details below.

Discounted access to Email and News services - TMX, Dial-IX and connect.com.au provide electronic mail and news access to all the major networks. They all provide substantial discounts to AUUG members. Contact the AUUG Secretariat for details.

Reciprocal Membership Rights with International Affiliates (e.g. Usenix, UniForum) - see details below.

Discounts on products and services - AUUG has negotiated discounts at selected book stores and service suppliers (e.g. Prentice Hall, NetComm). Details are available from the AUUG Secretariat or the AUUG Members Handbook.

Discounted education - AUUG members have access to technical training organisations like Softway at discounted rates. Contact the AUUG Secretariat for details.

AUUG Annual Conference and Exhibition - AUUG members get discounted registration fees to Australia's premiere Open Systems Conference and the largest Exhibition of UNIX and Open Systems equipment in Australia. See details below.

(12) How do I get a discount on AARNet through AUUG?

AARNet accepts Mail Affiliates from the general public for a minimum of $1000 per annum. AUUG offers its machine-owning members (thus usually corporate members) the same service for $250 per annum. Non-members can still profit from the AUUG deal but pay $600 per annum. As AUUG corporate membership costs $325 per annum, you can see why we have only one or two non-members involved.

The Mail Affiliate service provides low volume users the right to use AARNet to carry mail traffic. This service gives you registration of

your host/domain addresses in the Internet nameserver tables. It does NOT provide or cover the cost of any kind of connection to the Internet or any other host. In other words, you approach AUUG after you are connected to get the rights to transport your mail traffic across AARNet. This will provide you with an MX record which gets your name "switched on" in the Internet world. If you would like a registration form, please contact the AUUG Secretariat (see contact details above).

To get help you get connected AUUG conducts a survey which assists people requiring network connections to find people who are willing to accept new connections. Contact the AUUG Secretariat for details.

(13) What do the reciprocal membership rights give me?

AUUG is affiliated with with Usenix, UniForum, X/Open, and EurOpen. These organisations provide AUUG members with access to some of their member benefits. In general this provides AUUG members with member prices on publications and member discounts at events. The exact details will vary with the organisation, consult the AUUG member handbook for details.

(14) What is the AUUG National Conference and Exhibition?

AUUG runs a national annual conference and exhibition referred to as AUUGxx (where xx is the year) in September each year. The venue rotates between Sydney and Melbourne. AUUG92, held at the World Congress Centre in Melbourne had over 500 delegates to the conference and over 2000 visitors to the exhibition.

The conference is Australia's premier event on UNIX and Open Systems. It runs for 3 days and features many overseas invited speakers as well as speakers from around Australia. On the day before the conference there are tutorials given by recognised national and international experts. The tutorials cover a wide range of subjects and are targeted at both novice users and experienced users wishing to get detailed information in a specialist subject area.

The exhibition features the largest collection of UNIX and Open Systems hardware and software in Australia. All major vendors of UNIX equipment are present at the exhibition with their latest hardware and software.

(15) When is the next conference and exhibition?

The next conference and exhibition, AUUG93, is listed in AUUGN under General Information and will be held at the Darling Harbor Exhibition Centre in Sydney during September 1993.

(16) What are Summer Conferences?

To complement the National Conference a series of local summer conferences are held in Feb-Apr each year. These conferences feature local and national speakers on a range of subjects and are held in each local region. For details of the next summer conference in your region get in touch with your regional contact (see below).

(17) Who organises events in my region?

AUUG local region activities are organised by members in a local chapter. These activities are typically:

local technical meetings and seminars
an annual summer conference
local newsletter
technical library
access to electronic networks

AUUG regional contacts are:

Adelaide: Michael Wagner
Phone: (08) 212 2800
FAX: (08) 231 0321

Brisbane: Tim Butterfield
Phone: (07) 279-0149
FAX: (07) 279-0249

Canberra: John Barlow <john.barlow@anu.edu.au>
Phone: (06) 249 2930
FAX: (06) 249 0747

Darwin: Phil Maker <pjm@cs.ntu.edu.au>
Phone: (089) 46 6382 or 46 6666
FAX: (089) 27 0612

Hobart: Steven Bittinger <steven.bittinger@cc.utas.edu.au>
Phone: (002) 23 2811
FAX: (002) 23 1772

Melbourne: Stephen Prince <sp@cls.com.au>
Phone: (03) 608 9011
FAX: (03) 608 0505

Perth: Glenn Huxtable <glenn@cs.uwa.edu.au>
Phone: (09) 380 2878
FAX: (09) 380 1089

Sydney: Peter Chubb <peterc@softway.sw.oz.au>
Phone: (02) 698 2322
FAX: (02) 688 9174

If there is no regional contact listed for your region or you would like to organise events in your region please contact Peter Wishart <pjw@lobo.canberra.edu.au> or Glenn Huxtable <glenn@cs.uwa.oz.au>.

(18) What does AUUG stand for?

AUUG was originally formed as the "Australian Unix-systems Users Group", hence the abbreviation "AUUG". Officially AUUG is now known as "AUUG Inc.". The group is formally incorporated in the state of Victoria as "AUUG Inc." (registered number A0016636N).

(19) How did AUUG get started?

AUUG was founded in 1974 by Professor John Lions from the University of New South Wales. It began life as an informal gathering of computer people interested in the "new" operating system UNIX developed by AT&T Bell Labs. By 1984 the interest had gathered enough momentum that it reached across Australia.

AUUG adopted a constitution on 27th of August 1984 and was formally incorporated on the 26th of August 1988.

(*) UNIX is a registered trademark of UNIX System Laboratories Inc.

Information contained herewithin is valid until 31 May 1993.

AUUG
MANAGEMENT COMMITTEE
MINUTES OF MEETING 12 October 1992

Held at ACMS, Paddington

Present: Frank Crawford (afternoon), Glenn Huxtable, Rolf Jester(morning), Phil McCrea, Michael Paddon, Greg Rose, John O'Brien, Peter Wishart, Michael Tuke, and Liz Fraumann.

Meeting commenced at 10:06am

Wael Foda was present for most of the morning for AUUG '92 and future conference discussions. Lachie Hill presented a Public Relations proposal at 1:30 - 1:45p

1. MINUTES OF LAST MEETING (8 September 1992)

Were assumed to be correct and stand as presented

2. PRESIDENT'S REPORT

Phil McCrea stated he was extremely pleased with the AUUG '92 conference and exhibition. He thanked, once again, Liz Fraumann and Wael Foda for their efforts in helping to make it a success. He also noted that while several exhibitors were initially skeptical about the exhibition, in the end, they too were pleased with their results and participation.

3. SECRETARY'S REPORT

Peter Wishart reiterated that the chapter charter and rules are published in the forth coming issue of AUUGN and the AGM minutes have been posted to aus.auug. Finalisation of the chapter charter and rules are pending input from members who do not have access to electronic mail and may be sending comments in via post.

4. TREASURER'S REPORT

- (actually presented in the afternoon when FC arrived) Frank Crawford reported the current reporting mechanism will need to be modified to more closely provide the information required by the taxation department with details of subscriptions. He will work closely with the Secretariat to ensure the guidelines are met. He also issued copies of the auditor's report to the committee. The report indicated 3 discrepancies one of which was a double payment to Symmetry for $1,500.00. It was suggested by Glenn Huxtable to issue a letter to symmetry indicating the error and requesting the funds. In addition follow up phone calls may be necessary. Frank will follow-up.

ACTION: FC

Frank continued and informed the committee he was suggested to write a letter to the bank regarding the tax exempt status and it will likely be approved. If not, it will motivate the powers to be to inform AUUG as to the tax exempt status. Assuming the tax exempt status is obtained, it will mean a potential refund of withholding dollars.

5. BUSINESS CARRIED OVER

5.1 Assistant RO - MT - Not high priority - carried again

ACTION: MT

5.2 Legal advice chapters - PW Did not have time yet.

ACTION:PW

5.3 Financial Obligations (balance sheets) - PW Did not have time yet.

ACTION:PW

5.4 Summer Conf. call for papers - GH Glenn is working with organisers several dates are set. See update for summer conferences.

6. RECAP OF AUUG '92

Wael issued a detailed report of attendance, budgeted and actual figures. The percentage of non-members was less than anticipated and expenses were approximately $16,000.00 over anticipated amounts. Factors which play heavily into the over run figure are: speaker equipment, design/printing errors, and a larger number of speakers.

6.1 Discussion ensued: Wael suggested the current registration fees are not in line with what is being offered. It was suggested by PM to modify the fees to $350 for members and $450 for non-members. A single day registration of $150 was recommended. As of this date, exhibition space of 20% is booked for 1993. It was felt that a poor advertising and marketing campaign took place for 1992 and LF reminded the committee that what took place is a decidedly cut campaign from what was proposed. "We got what we asked for." Several committee members were unaware that both radio and television advertisements in conjunction with the exhibition had taken place.

It was suggested to limit numbers of attendees for tutorials and GR pointed out this could create a feeling of unrest and instead offered to that bookings made prior to a certain date will receive their first choice tutorial and after that date will be handled on a first come first serve basis. LF also suggested a system similar to USENIX where attendees must have a voucher for each specific tutorial to attend. This allows for planning of printed material and proper seating.

The group reiterated the feeling that papers should be submitted for inclusion in the proceedings and not simply hardcopy of slides to be presented. If slides are given for inclusion they must be between 3-4 slides per page. It has also been suggested the number of pages of papers be limited. These and other suggestions will be incorporated in the next release of the speakers' guidelines.

ACTION: LF

7. AUUG '93

7.1 Wael announced the dates of September 28-30 1993 at Darling Harbour for this conference. It was noted that it is a Tuesday - Thursday programme with the full day of tutorials being scheduled for Monday. Sept. 27th. It was also noted it is during school holidays. Debate ensued as to the pro/con of this time frame however, it is impossible to change the date maintaining the venue. It is felt with proper PR it may actually be an asset.

7.2 Suggestions for the theme included: UNIX vs. NT, Battle for the Desktop, UNIX for Solutions and Results through Open Systems. It was unamious in favour of Results through Open Systems. LF suggested a sub-theme for each day of Networking, Communications, and Security, respectively. The group was positive and felt this would attract day registrations.

7.3 Suggested names for the 1993 program committee included: Liz Fraumann, Ian Hoyle, Hugh Irvine, Rolf Jester, Piers Lauder, Phil McCrea, Greg Rose, and Ian Waters. Piers Lauder has volunteered as Program Chair. It was moved by JO and seconded by GR the committee will

consist of: Piers Lauder - Chair, Liz Fraumann, Rolf Jester, Hugh Irvine, and Greg Rose.

7.4 An exhibition Focus Group was established and will consist of Rolf Jester, Phil McCrea, Michael Paddon, Wael Foda, and Liz Fraumann. The purpose of the group is to work more closely with exhibitors on the theme and coordination of an "AUUG Village". This will be a section of the whole exhibit floor where "store fronts" will be arranged to show examples of open systems in operation. The space will be controlled by AUUG and leased to vendors or directly to the organisations using open systems.

7.5 It was suggested to pursue Clifford Stoll. LF had done initial research and noted he has associated fees with his presentations due to popularity. It was moved by GH/GR to pursue Clifford for a presentation at AUUG '93.

ACTION: LF

8. AUUG '94 AND FUTURE

8.1 Discussion as to the possibility of pursuing alternative venues took place. Initial thinking was AUUG should have AUUG '94 in Canberra. Wael stated the venue does not lend itself to the type of show we had even in Melbourne and highly recommended we consider having a static show in Sydney as most of the exhibitors are in that area. Wael suggested if this was unacceptable the committee could consider sponsoring one of the Summer Conferences to a greater extent and put on a small exhibition with it. Overall the committee agreed this was a fine suggestion. RJ/GR moved Perth be the conference of selection for Summer '93. Motion was passed. GH will work closely with conference organiser. Wael will reserve space in 94-Melbourne, '95 Sydney, and '96 Melbourne.

9. SUMMER CONFERENCES

9.1 GH updated the group on the progress of the Summer Conferences. He expects next month the call for papers till be issued in the majority of locations. Conference dates and speakers should be decided by December with conferences taking place between February and March.

9.2 Already scheduled are:
Canberra: 16 & 17 February (16th is a workshop)
Hobart:11 February will be charging this year $50 for members/$60 non

9.3 GH has informed organisers they should anticipate 1-2 speakers from interstate.

10. COLLATERAL

10.1 LF brought in a rough draft of a handbook for members. This is designed to explain participation and securing of benefits to members. JO along with the rest of the committee were very please with the output and will provide edits/comments as appropriate. It is anticipated to have the handbook ready for distribution in January '93. It was determined rather than the presented A4 format, it should be constructed in A5 for easier handling and carrying. Documents such as the constitution and chapter policies which are included in an appendix will be reformatted at 8pt. type. Initial suggestions included an "about this document" in the front of the handbook and AUUG mail aliases listing.

ACTION: LF/committee

10.2 LF brought in samples of an overview pamphlet for AUUG. She has found in negotiation for benefits and generally there is currently no literature about the organisation except the application for membership, AUUGN, and the old Open Forum. These do not serve the purpose of telling companies or potential members about the organisation. The committee selected the "Opening the World of Open Systems" version. RJ will work with LF to complete text for insertion. LF will provide quotations for printing to committee via e-mail.

ACTION:LF/RJ

10.3 Connections Brochure - one which explains many of the viable options available for network connections and the different networks available is being pursued. LF will work with Dave K. who has offered to assist and pick up the current status.

ACTION: LF/DK

11. LACHIE HILL PROPOSAL

11.1 Lachie Hill, upon the request of PM came to the meeting after previous meetings with PM and LF to ascertain where her services may be best used by the organisation. She presented a project by project proposal of which included:

* Promotion of the Membership survey - this will include round table discussions with journalists as to the demographics of the organisation and future direction with PM and LF.
* Promotion of and Public Relations for the summer conferences - this would include a direct mail campaign and assisting with media promotion of speakers, and notification of calendars.
* Promotion of and Public Relations for AUUG '93 - this involved direct mail, industry brief, and other mechanisms to attract delegates, speakers, delegates, and general interest.

GR/GH moved we accept the project of membership survey promotion then proceed as desired with the other projects. Motion passed.

ACTION: LF/LH

12. LAPEL PINS

12.1 LF brought in samples of lapel pins at the request of the committee at the September meeting. Choice was made and JO/MP moved we purchase pins for distribution at the summer conferences and local chapter meetings. Motion carried. Colours of pin will be gold and burgundy.

ACTION: LF

13. NEGOTIATIONS ONGOING

13.1 LF review several vendors she is negotiating for discounts with. A "press release was issued for inclusion in the forthcoming AUUGN regarding a 12.5% discount with X/Open. She asked for names of modem companies. JO will supply names.

ACTION: JO/LF

14. SAGE-SIG

14.1 GR updated the committee of current activities and an out-growth of the LISA group from USENIX. He reported currently 1/3 of the USENIX members are members of SAGE. SAGE is a systems administrators group wanting to share ideas etc. He provided the following two proposals, indicating they both were relatively informal:

I. AUUG could establish a sub-chapter of SAGE in Australia and affiliate with the group in the U.S.

II. SAGE could start a SAGE sub-chapter in Australia and have a stand at AUUG exhibition etc.

14.2 Greg stated Hall Miller, Robert Elz, and himself are all interested in pursuing the group. He also suggested possibly having a highly technical conference similar to a LISA conference.

14.3 It was moved, CM/FC that AUUG form a sub-chapter for SAGE. Support will be provided by a minimal member fee. Motion passed.

14.4 The initial committee will consist of Hall Miller, Glenn Huxtable, Greg Rose, and Frank Crawford. They will provide a posting for the next issue of AUUGN.

ACTION: GR/group

15. QUEENSLAND CHAPTER UPDATE

15.1 GH reported on the most recent trip he and LF took to attend the QUUG group meeting. The reminder meeting was instrumental in gathering approximately 8 of the ~40 attendees. Norm B., current chairman of the group with assistance from Tim B. have agreed to put a simple majority vote for a decision on whether to become a chapter or not.

15.2 PM will attend their next meeting on 27 October to field questions and be present for vote. TB had also scheduled a seminar but has yet to send fax with details for PM participation. A reminder will be distributed to the QLD AUUG members for venue of meeting and an opportunity to vote.

ACTION: LF/PM

16. AUUGN

16.1 JC and LF met with respect to input from the membership survey on incorporating additional items into AUUGN. The cover will be redone to match the letterhead. It will incorporate sections and have a bleed-edge tab to make finding sections easier. LF is negotiating with a vendor to secure software to make reformatting easier in exchange for some advertising.

16.2 PM also noted Jagoda is doing an outstanding job with AUUGN and while not present to hear this, she received many thanks for her efforts.

17. AFUU

17.1 Michael Tuke reiterated the details which had been posted with respect to our exchange. It has been there will be a 1 year trial where this exchange will offer, air, accommodation, and conference registration for the presenting of a paper and either 1 full day tutorial or 2 half day tutorials.

17.2 MT had placed the call for papers on the net and to date has received no responses. He took the initiative to draft 2 abstracts for submittal by himself. GH suggested we contact Peter Elford for permission and availability for participation. GR also offered two of his papers. It was decided given permission from PE all three papers will be submitted and allow the AFUU to select one.

ACTION: MT

18. OTHER BUSINESS

18.1 PM stated he has been contacted by UniForum India for advice on how to run the organisation and conference/exhibition. He will reply.

ACTION: PM

18.2 DataPro will host a round table discussion which PM will participate in. The committee cautioned in the past this had been done and DataPro had taken the information provided as their own. Caution was advised.

ACTION: PM

18.3 CD Rom exchange - USENIX is sourcing free software such as GNU. Discussion ensued whether AUUG should do a bulk purchase and distribute through the local chapters. It was suggested a library be maintained. CM pointed out the returning of the CD may prove to be a problem and suggested an order blank to be published in AUUGN. GH offered to actively pursue order taking and will produce an order form for release.

ACTION: GH

18.4 PW suggested a list of Frequently Asked Questions (FAQ) which will be posted on aus.auug on a regular basis. Questions will include: What is AUUG?, What is MX?, etc.

ACTION: PW

18.5 Michael Tuke presented a document of Election Procedures. With minor edits, the group felt the document was good. MT aims are to eventually modify the constitution to conform to the outline of the procedures presented.

ACTION: MT

19. NEXT MEETING(s)

4 December 1992 @ACMS

Future dates will be determined in December and will coordinated with summer conferences.

Meeting adjourned at 4:39pm

Respectfully Submitted,

Elizabeth A. Fraumann

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 an 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 Membership is not a membership you can 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.

It's also possible to subscribe to the newsletter without being an AUUG member. This saves you nothing financially, that is, the subscription price is greater than 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 your membership type, examine your membership card or the mailing label of this AUUGN. Both of these contain information about your current membership status. The first letter is your membership type code, M for regular members, S for students, and I for institutions, or R for newsletter subscription. Membership falls due in January or July, as appropriate. You will be invoiced prior to the expiry of your membership.

Check that your membership isn't about to expire and always keep your address up-to-date. 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 Visa or Mastercard by simply completing the authorisation on the application form.

AUUG Incorporated
Application for Institutional Membership
AUUG Inc.

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.

This form is valid only until 31st May, 1993

..... does hereby apply for

☐ New/Renewal* Institutional Membership of AUUG $325.00

☐ International Surface Mail $ 40.00

☐ 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 and becomes renewable on the following January or July, as appropriate.

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 usally 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.

- ☐ System V.3 source
- ☐ System V.2 source
- ☐ System V source
- ☐ System III source
- ☐ 4.2 or 4.3 BSD source
- ☐ 4.1 BSD source
- ☐ V7 source
- ☐ Other *(Indicate which)* .....

- ☐ System V.3 binary
- ☐ System V.2 binary
- ☐ System V binary
- ☐ System III binary

AUUG Incorporated
Application for Ordinary, or Student, Membership
AUUG Inc.

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.**

This form is valid only until 31st May, 1993

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 and becomes renewable on the following January or July, as appropriate.

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 Incorporated
Application for Newsletter Subscription
AUUG Inc.

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.

This form is valid only until 31st May, 1993

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:

☐ Subscription to AUUGN	$ 90.00
☐ International Surface Mail	$ 20.00
☐ 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

AUUG Inc.

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#* _____