AUUGN

AUUG Inc. Newsletter

Volume 15, Number 3

June 1994

The AUUG Incorporated Newsletter

Volume 15 Number 3

June 1994

CONTENTS

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
Freephone:	1-800 625 655
Email:	auug@munnari.oz.au

AUUG Business Manager

Catrina Dwyer,
P.O. Box 366,
Kensington, N.S.W. 2033.
AUSTRALIA

Phone:	(02) 959 3656
Fax:	(02) 957 6706
Email:	catrina@sw.oz.au

AUUG Executive

President

Phil McCrea
pmc@atom.ansto.gov.au

Watch this space ...

Vice-President

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

Secretary

Peter Wishart
peter.wishart@csis.dit.csiro.au
CSIRO Div. of Information Technology
GPO Box 664
Canberra ACT 2601

Treasurer

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

Committee Members

Stephen Boucher
stephen@mtiame.mtia.oz.au
MTIA
509 St. Kilda Rd.
Melbourne VIC 3004

Chris Maltby
chris@softway.sw.oz.au
Softway Pty. Ltd.
P.O. Box 305
Strawberry Hills NSW 2021

Rick Stevenson
rick@stallion.oz.au
Stallion Technologies Pty. Ltd.
56 Sylvan Rd.
Toowong, QLD 4066

Lucy Chubb
lucyc@softway.sw.oz.au
Softway Pty. Ltd.
P.O. Box 305
Strawberry Hills NSW 2021

Michael Paddon
mwp@munnari.oz.au
Kodak
173 Elizabeth St
Coburg, Vic 3058

AUUG General Information

Next AUUG Meeting

The AUUG'94 Conference and Exhibition will be held from the 7th to 9th September, 1994, at the World Congress Centre, Melbourne.

Advertising

Advertisements to be included in AUUGN are welcome. They should conform to the standards of other contributions (see page 5). Advertising rates are $120 for a quarter page, $180 for half a page, $300 for the first A4 page, $250 for a second page, $500 for the inside cover 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 business manager 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 and AUUG'93 are still available, at $50 for members and $60 for non-members. 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. A copy of FrameMaker for use in the production of the newsletter has been provided by Platform Technologies[*].

Disclaimer

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

---

[*] Platform Technologies are no longer distributors of FrameMaker, Information Technology Consultants, in Paddington NSW are now distributing FrameMaker.

AUUG Newsletter

Editorial

Welcome to AUUGN Volume 15 Number 3. We are now well into the year, with final plans being made for the conference. You should start seeing announcements, *etc*, for AUUG94 in papers and magazines.

We also have a "new" management committee. The Returning Officers report is published elsewhere in this issue. I'd like to take this opportunity to congratulate those elected, and to those who weren't elected, keep trying, AUUG needs your continuing support.

In addition to the regular features, such as the Chapter reports and the AUUG related announcements, we have a report on UniForum NZ'94, some papers from the various Summer Conferences, and some reprints from a new publication "The Australian System Administrator", the newsletter of SAGE-AU.

We also have reprinted a number of the AUUG columns from the Australian. These are as they were submitted, so you can see what the author originally wrote. You will also see the area covered by AUUG, from PC's to the Internet, from System Administration to Object-Oriented Programming, from highly commercial to home computing.

Hope you enjoy it, and if you have anything of interest, be it technical or otherwise, please consider submitting it for publication.

Jagoda Crawford

AUUGN Correspondence

All correspondence regarding the AUUGN should be addressed to:-

AUUGN Editor,   Phone: +61 2 717 3885
P.O. Box 366,   Fax: +61 2 717 9273
Kensington, N.S.W. 2033.   Email: auugn@munnari.oz.au
AUSTRALIA

AUUGN Book Reviews

The AUUGN book review editor is Frank Crawford. Anyone interested in reviewing books or with book reviews to submit for publishing in AUUGN please contact Frank. His address can be found on page two of this issue. Remember, that any books you review, you keep.

Contributions

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

Volume 15 No 4   Friday 29th July
Volume 15 No 5   Friday 23th September
Volume 15 No 6   Friday 25th 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 margins, and 30 mm left at the bottom so that the AUUGN footers can be pasted on to the page. Small page numbers printed in the footer area would help.

AUUG President's Page

"Let's not confuse the User Interface with the Operating System..."

I have been a UNIX user since 1975, as a result of its arrival at UNSW direct from Bell Labs. I lay no claim to being a UNIX expert, but, apart from a brief dalliance with VMS when I worked for AAP during the years 1986 - 1988, I have been 'using' UNIX for almost 20 years (which used to be called a 'generation' when I was at school...).

I've progressed through a variety of UNIX terminals and interfaces, from VT100s, Ampec terminals, X terminals and PCs; from ed, vi, rn, nn etc. My most recent configuration involves a PC, running telnet when I want to access my friendly UNIX machine - not an ideal arrangement, but it is a good compromise.

I have been flirting with Windows for almost a year now, and must confess I really like it. I still don't like having to use the mouse quite as much as I have to, but by and large, it is a good User Interface. And forgetting whether it is good or not, it is effectively a defacto standard - I forget how many copies of Windows have been shipped, but there could not be too many PCs out there without Windows.

Microsoft have been keen to call their new operating system "Windows NT", which annoys me, because Windows is a User Interface, and NT is an operating system - they are a long way apart on the layered software model: an operating system is close to the hardware, and the User Interface is close to you and me, the users. In between there are all sorts of software layers such as databases, communications software, transaction monitors, applications programs, screen protocols, etc.

So why does Microsoft try to tie the operating system and the User interface together? It's easy to answer: they are leveraging off the success of Windows, and want the new operating system to be seen as a new version of Windows... A nice approach, and it appears to be working!

But Microsoft have changed their approach of late - the name NT has been dropped, or at least down-played, in favour of city names, such as Chicago, Cairo, or Daytona! I'm not sure what the obsession with cities is - will we ever see Microsoft come up with an operating system called 'Sydney', or 'Hobart', or even 'Wagga Wagga'!! No, cities for me work well for rock groups - anyone remember Chicago, or Kansas? - but not for operating systems.

The real point of the above is that the association with Windows has been dropped. Why is this? Easy - Microsoft has designs on the server market, not just the desktop. The Windows association does not help the server operating system market.

The server market, of course, is the domain of UNIX at the present point in time: MIS Research have reported in their MIS3000 survey that UNIX is used in over 47% of the top 3000 IT sites in the country, and rising fast. It is unlikely that UNIX can be toppled from the server market, but Microsoft is making a strong effort.

So what are the 'cities' all about? Well, despite their marketing might, Microsoft have managed to confuse the marketplace because people are unsure whether NT was (is?) a desktop or a server operating system - it was being targeted at the desktop at the same time as companies like Sequent were planning to make it available as a server.

The cities is an effort to clear things up. Chicago is the next version of NT especially for the desktop. It's running late, which is not unusual... And Cairo is the next version for the server market. Confused? Join the club!
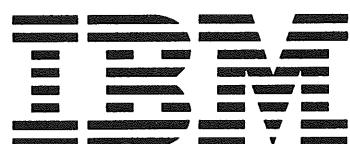
In an ideal world, we would have UNIX running everywhere!

Phil McCrea

AUUG Inc.
Corporate Sponsors

AUUG Inc. is pleased to acknowledge the generous support given by the following corporate sponsors:

A. Goninan & Co. Limited
A.N.U.
AAII
Aberfoyle Resource Limited
ACAY Network Computing Pty.Ltd.
ACT Government Computing Service
Actrol Parts
Adept Software
Advanced Software Engineering
Alcatel Australia
Amalgamated Television Services
Amdahl Australia Pty Ltd
Amdahl Pacific Services
Andersen Consulting
ANI Manufacturing Group
Animal Logic Research Pty. Ltd.
ANSTO
Anti-Cancer Council of Victoria
ANZ McCaughan
Atlas Computer Systems
Attorney-General's Dept.
AUSOM Inc.
Australian Airlines Limited
Australian Archives
Australian Bureau of Statistics
Australian Defence Industries Ltd.
Australian Electoral Commission
Australian Film Television and Radio School
Australian Information Processing Centre Pty. Ltd.
Australian Museum
Australian National Audit Office
Australian Submarine Corporation
Australian Taxation Office
Australian Technology Resources (ACT) Pty. Ltd.
Australian Technology Resources (WA) Pty. Ltd.
Australian Tourist Commission
Australian Wool Research & Promotion Organisation
B & D Australia
Bain & Company
Bay Technologies Pty Ltd
BHA Computer Pty. Limited
BHP Information Technology
BHP Petroleum
BHP Research - Melbourne Laboratories
BHP Research - Newcastle Laboratories
Bond University
Burdett, Buckeridge & Young Ltd.
Bureau of Meteorology
Butterworths
Bytecraft Pty. Ltd.
C.I.S.R.A.
Cadcom Solutions Pty. Ltd.
Cape Grim B.A.P.S
Capricorn Coal Management Pty. Ltd.
CelsiusTech Australia
Chief Secretary's Dept.
CITEC
Clarity International
Classified Computers Pty. Ltd.
Clegg Driscoll Consultants Pty. Ltd.
Co-Cam Computer Group
Coal & Allied Operations

Cognos Pty. Ltd.
Com Net Solutions
Com Tech Communications
Commercial Dynamics
Communica Software Consultants
Composite Buyers Ltd.
Computechnics Pty. Ltd.
Computer De Tokyo Corporation
Computer Law Corporation
Computer Sciences of Australia Pty. Ltd.
Computer Software Packages
Computer Systems (Australia) Pty. Ltd.
Comsys International Pty. Ltd.
Concord Repatriation General Hospital
Copper Refineries Pty. Ltd.
Corinthian Engineering Pty. Ltd.
Corporate Systems Planning
Corporate Workgroup Resources
CSIRO Division of Information Technology
CSIRO Division of Manufacturing Technology
CSIRO Division of Wool Technology
Curtin University of Technology
Customised Software Solutions Centre
Cyberdyne Systems Corporation Pty. Ltd.
Cyberscience Corporation Pty. Ltd.
Cybersource Pty. Ltd.
Data General Australia
Datacraft Limited
Datacraft Technologies
Dawn Technologies
Deakin University
Defence Housing Authority
Defence Service Homes
Dep. of Environment & Natural Resources
Department of Defence (TC Section)
Department of Education QLD
Dept. of Business & Employment
Dept. of Defence
Dept. of Family Services & Aboriginal & Islander Affairs
Dept. of Industrial Relations, Employment, Training & Further Education
Dept. of Planning & Development
Dept. of State Services
Dept. of the Treasury
Dept. of Transport
Department of Planning & Development
DEVETIR
Dialix
Digital Equipment Corp. (Australia) Pty. Ltd.
DSTO, Lab 73
Earindi Holdings Pty. Ltd.
EASAMS (Australia) Limited
Edith Cowan University
Electricity Trust of South Australia
Electro Optics Pty. Ltd.
Engineering Computer Services Pty. Ltd.
Equity Systems Pty. Limited
ERIN Unit, Australian National Parks & Wildlife Service
ESRI Australia Pty. Ltd.
Executive Computing

AUUG Institutional Members as at 14/06/1994

FGH Decision Support Systems Pty. Ltd.
Financial Network Services
Fire Fighting Enterprises
First State Computing
Flinders University
Fremantle Port Authority
Fujitsu Australia Ltd.
G.James Australia Pty. Ltd.
GEC Alsthom Australia
GEC Alsthom Information Technology
GEC Marconi Systems Ltd.
Geelong & District Water Board
Genasys II Pty. Ltd.
General Automation Pty. Ltd.
GIO Australia
Golden Casket Office
Golden Circle Australia
Great Barrier Reef Marine Park Authority
Gribbles Pathology
Haltek Pty. Ltd.
Hamersley Iron
Hamersley Iron
Heath Insuramce
Hermes Precisa Australia Pty. Ltd.
Hollywood Private Hospital
Honeywell Australia Ltd.
Honeywell Ltd.
Hong Kong Jockey Club Systems (Australia)
        Pty. Ltd.
I.P.S Radio & Space Services
IBA Healthcare Pty. Ltd.
IBM Australia Ltd.
Iconix Pty. Ltd.
Ideas International Pty. Ltd.
Independent Systems Integrators
Information Technology Consultants
Information Technology Services Branch,
        Dept of Public Administration
Informed Technology
Insession Pty. Ltd.
Insurance & Superannuation Commission
Integration Design Pty. Ltd.
International Imaging Systems
Internode Systems Pty. Ltd.
ISR Group Ltd.
James Cook University of North Queensland
JTEC Pty. Ltd.
Knowledge Engineering Pty. Ltd.
Laboratory Systems Pty. Ltd.
Labtam Australia Pty. Ltd.
Land Information Centre
Land Titles Office
Leeds & Northrup Australia Pty. Limited
Legent Australia Pty. Ltd.
Logica Pty. Ltd.
Lotus Development
Lyons Computer Pty. Ltd.
Macquarie University
Main Roads Western Australia
Matcom Technologies
Mayne Nickless Courier Systems
Mayne Nickless Information Technology Services

Medical Benefits Funds of Australia Ltd.
Memtec Limited
Mentor Technologies Pty. Ltd.
Mercedes-Benz (Australia)
Metal Trades Industry Association
Mincom Pty. Ltd.
Minenco Pty. Ltd.
Mitsubishi Motors Australia Ltd.
Mitsui Computer Limited
Moldflow Pty. Ltd.
Motorola Computer Systems
Motorolla Communications Australia
MPA International Pty. Ltd.
MUA Pty. Ltd.
Multibase Pty. Ltd.
Multiline BBS
Multiuser Solutions Pty. Ltd.
National Library of Australia
National Resource Information Centre
NCR Australia
NEC Australia Pty. Ltd.
Northern Territory Library Service
Northern Territory University
Novell Pty. Ltd.
NSW Agriculture
NSW Teachers Federation Health Society
Object Design Pty. Ltd.
Object Oriented Pty. Ltd.
Object Technology International Pty. Ltd.
Ochre Development
Office of National Assessments
Office of State Revenue
Office of the Director of Public Prosecutions
Open Software Associates Ltd.
Open Technology Pty Ltd
Opentec Pty Ltd
OPSM
OSIX Pty. Ltd.
OzWare Developments Pty. Ltd.
Pacific Semiconductor Pty. Ltd.
Pacific Star Communications
Petrosys Pty. Ltd.
Philips PTS
Platform Technologies Pty. Ltd.
Port of Melbourne Authority
Powerhouse Museum
PPIT Pty. Ltd.
Process Software Solutions Pty. Ltd.
Prospect Electricity
pTizan Computer Services Pty. Ltd.
Public Works Department, Information Services
Pulse Club Computers Pty. Ltd.
Qantek
QLD Electricity Commission
Quality By Design Pty. Ltd.
Redland Shire Council
Rehabilitation Tasmania
Renison Golfields Consolidated Ltd.
RGC Minerals Sands, Divisional Office
Rinbina Pty. Ltd.
Royal Melbourne Institute of Technology
SCEGGS Redlands Ltd

AUUG Institutional Members as at 14/06/1994

Sculptor 4GL+SQL
SEQEB Business Systems
SEQEB Control Centre
Siemens Nixdorf Information Systems Pty. Ltd.
Smorgon ARC
Snowy Mountains Authority
SoftGen Pacific Pty. Ltd.
Software Development International Pty. Ltd.
Softway Pty. Ltd.
South Australian Co-operative Bulk Handling
St. Catherine's School
St. Gregory's Armenian School
St. John God Hospital
St. John of God Health Care System
St. Vincent's Private Hospital
Standards Australia
State Revenue Office
State Super (SSIMC)
Steelmark Eagle & Globe
Sterling Software
Storage Technology of Australia
Strategic Information Technologies Pty. Ltd.
Sunburst Regency Foods
Sydney Electricity
Sydney Ports Authority
System Builder Development Pty. Ltd.
Systems and Management Pty Ltd
Systems Development Telecom Australia
TAB of Queensland
TAFE NSW, Information Systems Division
Tandem Computers
Tattersall Sweep Consultation
Technical Software Services
TechNIX Consulting Group International
Telecom (Applied Research & Development)
Telecom Australia
Telecom Australia Corporate Customer
Telecom Network Engineering Computer
        Support Services
Telecom Payphone Services
The Far North QLD Electricity Board
The Fulcrum Consulting Group
The Knowledge Group Pty Ltd
The Preston Group
The Roads & Traffic Authority
The Southport School
The University of Western Australia
Thiess Contractors Pty. Ltd.
Thomas Cook Ltd.
TNT Australia Information Technology
Toshiba International Corporation Pty. Ltd.
Tower Technology Pty. Ltd.
Tradelink Plumbing Supplies Centres
Turbosoft Pty. Ltd.
TUSC Computer Systems
UCCQ
UCR
Unidata Australia
Uninet Consulting Pty. Ltd.
Unisys Australia Limited
University of Adelaide
University of Melbourne

University of New England, Dept. of Maths,
        Stats & Computer Science
University of New South Wales
University of Queensland
University of South Australia
University of Sydney
University of Tasmania
University of Technology, Sydney
Unixpac Pty. Ltd.
Vanguard Computer Services Pty. Ltd.
Vanoco Pty. Ltd.
Vicomp
Victoria University of Technology
VME Systems Pty. Ltd.
Walter & Eliza Hall Institute
Wang Australia Pty. Ltd.
Water Board
Western Mining Corporation
Woodside Offshore Petroleum
Work Health Authority
Workstations Plus
XEDOC Software Development Pty. Ltd.
Zircon Systems Pty. Ltd.
Zurich Australian Insurance

Letters to the Editor

Michael,

Having just received our copy of AUUGN (Vol 15 No 2) I noticed your response to the response to Stuart McCormack's article which appeared in the Australian.

I appreciate the article's humour as intended and support the pro-Unix/anti-DOS sentiment expressed, I would like to point out however that contrary to Stuart's statement in paragraph 17, Norton's utilities _are_ available for Unix.

ITC is the distributor of Almond Utilities[†] (formerly Norton's Utilities for Unix) which is a SCO Unix port of the P.C. Norton Utilities. Ports to other platforms, including Sun, are expected soon.

I would appreciate it if you would forward this information to Stuart McCormack and to anyone else who you feel is interested in this kind of facility in the Unix environment.

Also, I would be grateful if AUUGN could publish a correction to this statement from Stuart's article in it's next issue.

Best regards,

Brendan Hills
<brendan@itc.oz.au>

† See page 12.

Returning Officer's Report on the 1994 Election

A total of 224 votes were received by the due date. 4 were set aside as the outer envelope had not been signed. 1 was set aside because it was not in the outer envelope. 2 were set aside as the signatures could not be found on the membership lists. This left a total of 217 valid votes.

PRESIDENT

McCrea	120	* elected
Paddon	90	
Informal	7	

VICE PRESIDENT

Huxtable	132	* elected
Maltby	75	
Informal	9	

SECRETARY

Purdue	63	
Wishart	145	* elected
Informal	9	

TREASURER

Boucher	63	
Crawford	141	* elected
Informal	13	

COMMITTEE MEMBERS
Elected (in order):   Paddon
Maltby
Chubb
Boucher
Stevenson

Unsuccessful candidates (in reverse order of elimination):
Purdue
Goodheart
Hughes
Koltai
Jelfs

RETURNING OFFICER
David Purdue (elected as the only candidate left in the ballot)

ASSISTANT RETURNING OFFICER
vacant

Request for Information on Crackers/Hackers

Ian Hayward is a PhD candidate at the Victoria University of Technology, doing a thesis on the cracker/hacker profile. The intention at this stage is to delineate common characteristics of these unauthorised users of computer systems with a view to building an expert system to aid in computer security. Hopefully the expert system will be able to give an indication of user's security risk and possible security measures that can be taken in order to minimise the risk associated with a computer system.

The type of data Ian is trying to collect on "Crackers/Hackers" for his research is the following :-

— handle

— past handle(s)

— gender : male female homosexual lesbian etc.

— age at start of hacking

— age finished hacking (if finished)

— date of birth

— height

— weight

— eye colour

— hair colour

— computers used

— interests

— convictions if any

— other hackers/crackers known

— country of origin

— ethnic background : Anglo-Saxon, Italian, Greek, etc.

— marital status : married, single, divorced, widowed

— parental status : parent, non-parent, step-parent

— occupation : clerk, call-person, accountant, student etc.

— position : managing director, unemployed, politician etc.

— industry : construction, education, manufacturing etc.

— hacking accomplished : mainframe unix, vax, IBM, etc.

— country computers hacked into : US, Australia, etc.

— etc.

Obviously Ian does not want any names, but some indication of any hacking/cracking which has been undertaken would be very useful. Ian's contact details are as follows:

Ian J. Hayward
Lecturer in Computing Department of Business Computing
(Footscray Campus)
Victoria University of Technology
P.O. BOX 14428,
MELBOURNE MAIL CENTRE,
VICTORIA, AUSTRALIA 3000

E-mail : ianh@matilda.vut.edu.au

phone : +61 3 6884130 (Bus.)
+61 3 7430941 (Home)

fax : +61 3 6885024

Call for Articles for the Australian

The Australian newspaper runs an AUUG column every Tuesday, in its computer section. The aim of these articles is to inform the public and raise the profile of open systems within this country. Having one's views published in a respected national paper also carries kudos and recognition for authors.

AUUG would like to ensure that all members of the open system community have access to this voice, and we are actively seeking a diverse spectrum of people and opinions.

If you are interested in being part of this process, please provide me with the following information:

* your name
* contact details
* a copy of your article

Your article should be between 600 and 800 words in length, and can address any issue that may be of interest within the open systems community. If you can't decide on an appropriate topic, please provide me with some professional details and I'll try and give you some ideas tailored to your expertise. Some typical subjects are listed at the end of this article.

If you have access to email, this is the preferred form of submission. Please format your article as a plain text file, with lines no longer than 79 characters, and with a blank line separating paragraphs. If you don't have email, please provide a hardcopy in a similar format (there's not much point doing any fancy typesetting).

All submissions are accepted on the understanding that they may or may not be used and that the material may be edited. AUUG will only submit your work to the Australian newspaper, although unless you advise us otherwise we will reserve the right to add your articles to a public FTP archive at some time in the future. The copyright on the material remains yours, your act of submitting material only gives us licence for the abovementioned purposes.

In practice, I submit your work to the Australian unedited and leave the decision of what to print up to them (I'm not in the business of being a thought police!). Usually a period of 2 to 4 weeks will then pass before you'll see your article in print; I maintain a pipeline of material to buffer me against the inevitable fluctuations in supply.

Please email or phone me if you have any further queries.

Michael Paddon
mwp@mtiame.mtia.oz.au
(03) 353 2382

Some topical areas to get you started :-

Standards: POSIX, X/Open, System V.
The sudden demise of COSE;
        just another consortium?
The history of Unix.
The future of Unix.
If NT is so popular,
        how come no one is using it?
Competition for the desktop:
        Unix, Windows/NT and OS/2.
Managing security.
Computer viruses.
System administration.
Network administration.
Networking technologies.
Distributed computing.
What happened to OSI?

Managing multiple network protocols.
Living with the Internet.
Unix on PC's.
Linux - the people's Unix?
Would you run Unix at home?
The graphics revolution.
Virtual reality.
CASE tools.
Is Unix really that hard to use?
Now that Unix has grown up,
        where have the hackers gone?
The costs of open systems.
Analyse a market trend.
Run a straw poll on a topical subject.
New technologies.

AUUG '94 UPDATE

Sydney, May 1994 ..... With AUUG '94 only four months away, the conference and exhibition is well underway to repeating the success of AUUG '93.

Conference Programme -- Following the Call for Papers, the programme committee was very pleased with both the quantity and the quality of the submissions. Selecting papers for the conference was a difficult job. In keeping with the theme of AUUG '94, *"Open Systems. Looking into the Future"*, the conference programme will address the vital components of open systems - standards, networking and security - as well as providing world class case histories from Australian organisations who have sucessfully implemented open systems architectures. Programme Chair, Ian Hoyle said "We have put together a well balanced programme with a number of quality Australian and international speakers. The concentration on successful case studies brings AUUG even closer to the user community".

Tutorials -- Tuesday, September 6th promises to cover a wide variety of topics from the ever popular PERL, C++, Tcl and Internet connection to linux, security, cryptography and DCE. AUUG will be offering a choice of three full day tutorials and six half days.

New Feature -- This year AUUG has introduced a new feature called "The guru is IN". Delegates will be given the opportunity to quiz a "guru" or panel of "gurus" on a popular technical topic. Similar in style to the popular Birds of a Feather (BoF) sessions, the TGII sessions will be scheduled closer to the conference. Current topics include PERL and X/Motif/Tcl. All gurus will be suitably identifiable, so watch out!

Exhibition -- The concurrent AUUG '94 exhibition is now one of the largest (non PC) shows in the region. With only limited space still available, exhibition organiser, Wael Foda of ACMS said "To those companies thinking of exhibiting at AUUG '94, I would urge them to contact me as soon as possible as there are only a small number of spaces still available". AUUG acknowledges the generous support of their major sponsors; AT&T Global Information Solutions, Hewlett Packard Australia Limited and IBM Australia Ltd.

Registration -- AUUG members should have received the Early Bird Registration form by now. This early bird registration offers members the opportunity to save over $50 on the regular conference registration fee. Remember ..... the closing date for the early bird is 22 June.

SAGE-AU

The System Administrators Guild of Australia

It ought to go without saying that every organisation with shared computers or networks needs at least one professional systems administrator.

But if you're a system administrator, you're probably sick of trying to explain -- to your management, your coworkers and almost everyone you meet -- just what it is you do for your organisation, and why it's important. You may feel you're battling on alone, with no-one to turn to for advice.

In response to these needs, systems administrators formed SAGE-AU, the Systems Administrators Guild of Australia, in early 1993. Its aims are to share knowledge and advance the standing of the profession.

SAGE-AU welcomes people who run all kinds of computers and networks -- such as LANs, WANs, Unix, VMS, MVS, Macintosh, Novell -- because they share many concerns and can learn a lot from each other.

SAGE-AU's second annual conference is being held in Perth, from 11 to 13 July. Steve simmons, a noted US expert, as well as local and interstate systems administrators, will be speaking on both technical and general topics. And there will be plenty of opportunity to get together with your fellow professionals to share ideas and stories.

For more information on SAGE-AU and the conference, call Glenn Huxtable on 380 2878, fax 380 1089 or email glenn@cs.uwa.edu.au.

From an editorial by Janet Jackson, to appear in the *Business News* on June 23.

Updated AUUG Regional Contacts
1994 -1995

Location	Contact	Tel/Fax
Adelaide	Michael Wagner mhw@syserv.com.au	tel: (08) 212-2800 fax: (08) 231-0321
Brisbane	Greg Birnie greg@lna.oz.au	tel: (07) 340-2111 fax: (07) 340-2100
Canberra	John Barlow john.barlow@anu.edu.au	tel: (06) 249-2930 fax: (06) 249-0747
Darwin	Phil Maker pjm@cs.ntu.edu.au	tel: (089) 466-666 fax: (089) 270-612
Hobart	Steven Bittinger steven.bittinger@its.utas.edu.au	tel: (002) 207-406 fax: (002) 207-488
Melbourne	David Taylor davet@vaxc.cc.monash.edu.au	tel: (03) 857 5660
Perth	Glenn Huxtable glenn@cs.uwa.edu.au	tel: (09) 380-2878 fax: (09) 380-1089
Sydney	Julian Dryden julian@dwt.csiro.au	tel: (02) 809-9345 fax: (02) 809-9476

AUUG QUEENSLAND CHAPTER UPDATE

QAUUG May Meeting Report

(or "The Zen of Shuffleboard")

About 20 hardy souls braved an unseasonally cold and wet evening last Monday, May 31st, to attend a QAUUG meeting held at a new venue - Digital's plush new offices in Eagle Street, Brisbane. This attendance figure is somewhat low compared to what we've come to accept - perhaps it was just the weather, perhaps the venue change (QAUUG meetings are historically held at the Regatta Hotel), perhaps someone's fax machine simply broke under the weight of broadcasting reminder notices.....

Anyway, the meeting (kindly sponsored by Digital, who supplied drinks and munchies - thanks Warren!) centered around a presentation of DEC's Linkworks product - a piece of software that I'm told is nothing like Lotus notes, only better! Linkworks is basically a customisable GUI front end to just about any application, and works on a bunch of different windowing platforms, giving the user an opportunity to remain blissfully ignorant of the underlying filesystem architecture. Linkworks also facilitates efficient work flow of documents, messages and files throughout an organisation. Yes it does Windows 3.1, yes it does UNIX - but the question remains: will it change the world?

The QAUUG members (a self-confessed drinking group with a serious computing problem :-) pondered these and other rhetorical concepts before, after, and during the presentation, and an enjoyable time was had by all. Again, thanks are due to Warren Verity of Digital, and also the team from Applied Micro Systems, who performed the Linkworks demonstration and presentation.

Following the meeting, an even hardier subset of the aforementioned meeting attendees went on a thirst-quenching quest around inner-city Brisbane. Upon finding and entering a suitable establishment they had no sooner ordered the first round when challenged by a similar sized group of non-industry types to a "friendly" game of shuffleboard! Needless to say, an engaging battle followed, and after much blood, sweat, and beer the QAUUG team emerged triumphant. Of course they were immediately challenged to a rematch - an invitation that was politely declined!

QAUUG Meetings are held on the last Tuesday of each month, at the Regatta Hotel, Coronation Drive, Toowong. All Open Systems users, managers and developers are welcome to attend.

Mark White
Pacific Star Communications
<mwhite@pacstar.com.au>

AUUG Canberra Chapter

The following is the schedule for the monthly meetings of the AUUG Canberra chapter for the remainder of the year. All meetings are held at the Open Solutions Center in Barry Drive, TURNER ACT.

Time and Date:	7:30pm 12/7/94	
Topic:	NextStep	(tentative)
Presenter:	Peter Milne	

Time and Date:	7:30pm 9/8/94	
Topic:	The Magic Garden Explained	(tentative)
Presenter:	Berny Goodheart	

Time and Date:	7:30pm 13/9/94	
Topic:	Linux	(tentative)
Presenter:	Linus Torvalds	

Time and Date:	7:30pm 11/10/94
	Annual General Meeting

Time and Date:	7:30pm 8/11/94
Topic:	???
Presenter:	???

Time and Date:	5:30pm ??/12/94
	Christmas drinks at ANU (details to be confirmed)

Peter Davie
Meeting Co-ordinator
AUUG Canberra Chapter

AUUG NSW Chapter

1994 Meeting Schedule

July	19	Tuesday
August	9	Tuesday
September		++ No Meeting AUUG Winter Conference ++
October	11	Tuesday
November	8	Tuesday
December	13	Tuesday

Venue will be the Novotel Hotel Darling Harbour, 19:00 for 19:30.

We are (always) on the lookout for topics and speakers, if you have something to talk about, drop us a line. For more information feel free to contact me:

Julian Dryden	or	Brenda Parsons
julian@dwt.csiro.au		bdp@sydney.dialix.oz.au
(02) 809 9345 (bh)		(018) 647 259
(02) 809 9476 (fax)		(02) 808 2797 (fax)

AUUG Inc. - Victorian Chapter
(formerly SESSPOOLE)

AUUG-Vic is the official Victorian chapter of AUUG Inc. It was the first Chapter of the AUUG to be formed, then known as SESSPOOLE, and its members have been involved in the staging of the Victorian AUUG Summer Technical Meetings every year since 1990. AUUG-Vic currently meets approximately every six weeks to hold alternate social and technical meetings. It is open to all members of AUUG Inc., and visitors who are interested in promoting further knowledge and understanding of UNIX and Open Systems within Victoria.

The purpose of the social meetings is to discuss UNIX and open systems, drinking wines and ales (or fruit juices if alcohol is not their thing), and generally relaxing and socialising over dinner. Whilst the technical meetings provide one or two "stand-up" talks relating to technical or commercial issues, or works in progress of open systems.

The programme committee invites interested parties wishing to present their work, to submit informal proposals, ideas, or suggestions on any topics relating to Open Systems. We are interested in talks from both the commercial and research communities.

Social meetings are currently held in the Bistro of the *Oakleigh Hotel, 1555 Dandenong Road, Oakleigh,* starting at about 6:30pm. Venues for the technical meetings are varied and are announced prior to the event. The dates for the next few meetings are:

Wed, 6 July '94	Social
Thu, 18 August '94	Technical
Tue, 27 September '94	Social
Wed, 9 November '94	Technical
Thu, 22 December '94	Social

Hope to see you there!

To find out more about AUUG-Vic and its activities, contact the committee or look for announcements in the newsgroup **aus.org.auug**, or on the mailing list **auugvic@clcs.com.au**. The committee may be reached more directly on **auugvic-exec@clcs.co.au**.

AUUG-Vic committee:

President:	Enno Davids	Metva	(03) 882 2333	enno@metva.technix.oz.au
Secretary:	David Taylor	Monash University	(03) 857 5660	davet@vaxc.cc.monash.edu.au
Treasurer:	Neil Murray	Webster Computer	(03) 561 9999	neil@wcc.oz.au
Programme Chair	Michael Paddon	Kodak	(03) 353 2382	mwp@munnari.oz.au
Committee:	Arnold Pears	La Trobe University	(03) 479 1144	pears@latcs1.lat.oz.au
Committee:	Peter Lazarus	Legent Australia	(03) 286 5200	plazarus@auspacific.legent.com

AUUG-Vic Email addresses:

General Membership	auugvic@clcs.com.au
Committee:	auugvic-exec@clcs.com.au
Mailing list administration:	auugvic-request@clcs.com.au

AUUG-Vic chapter activities

by Enno Davids
President AUUG-Vic
enno@metva.technix.oz.au

A new committee

Well these are my first notes as president of AUUG-Vic and there are a few things which are worth reporting on. First and foremost, as Steve reported in his last column, we now officially have a new committee. The change over has been somewhat more drawn out than I for one expected, but it seems to be all over bar some details now. The details of the new committee are in the usual place at the bottom of the meeting banner page.

As is usual for incoming presidents I would like to thank the outgoing committee, most of whom had been there since our very earliest days, before we even became a chapter. They have all done sterling service from which we the Victorian members of AUUG and in quite a few cases much of the national membership have benefitted. The new committee has a tough act to follow.

Postal Address

As part of the rigmarole of changing our banking details for the new committee we have arranged to get a 'real' postal address. This is in the form of a post office box in central Melbourne. This address will now be a point of contact for AUUG-Vic members and non members who need to send us something by non-electronic means. The address is:

> AUUG-Vic
> P.O. Box 14507
> Melbourne City Mail Centre
> Melbourne 3000

By the way, the old electronic mail addresses are not going away and in fact are still likely to be faster for general queries and talking to the committee. In fact, we still prefer they be used wherever possible. But just try sending a cheque that way when you book for the next Summer AUUG!

Social meetings.

For some time now there has been discussion in the group of moving the venue for the social meetings from the current location in Melbourne's south eastern suburbs to a more central location. The only trouble is, we have no real idea of where to move to. So, this is your chance to make a suggestion. Our current venue for the social meetings is one where we have had low (i.e. no) overheads. We have been able to book a reasonably quiet area for a variable number of people at no cost. The food has been good, the drinks are reasonably priced and the band ... is mostly not there. If anyone knows a similar place in or near the CBD we'd love to hear about it.

Internet access

Not a lot of progress has been made on this front since Steve reported on it last time. One or two members have made suggestions which we are looking at and we have a couple of offers on the table. Beyond this, things are still moving, just slowly.

Book discounts

Some discussion took place at the last committee meeting on the subject of book discounts with the major Melbourne based computer book vendors. Most of these appear to have lapsed and we are seeing into renewing the ones we had and where possible expanding these arrangements to new vendors. In particular, I would like to see us sign up a few hardware and software vendors in addition to the bookstores. Once again if anyone has any ideas here, feel free to drop us a line.

Well, that's about all for the moment. Remember, if you have something to say or a question to ask you can drop us a line at any time either electronically or via OzPost or even just come along and corner one of the committee at one of the social or technical evenings.

See you there,

Enno.

From the Western Front

WAUG held its Annual General Meeting on 18 May. Glenn Huxtable stood down after several years in the Chair, and nominated Adrian Booth to replace him. Adrian was elected unopposed.

We welcomed two new committee members, Tom Hallam and Russell Hampton. The new chapter committee is as follows.

Chair: Adrian Booth. Treasurer: Patrick Ko. Secretary: Major. Ordinary committee members: Mark Baker (meeting organiser), Luigi Cantoni, Tom Hallam, Russell Hampton, Glenn Huxtable, Don Griffiths, Janet Jackson (newsletter editor).

After the business part of the AGM we held an unusual meeting. I had heard something "different" was planned, but the details were kept secret. The advertised topic was "Unix Networking", with "experienced speakers". This had led me to expect something like a panel discussion on NFS, NIS, TCP, UDP, DCE, OSI, and the like.

However, our meeting organiser (Mark Baker) had in mind a different meaning of "networking". Each person was required to stand up and in 30 seconds describe themselves and their interest in Unix (it seems that the U in WAUG stands quite openly for Unix). Although some people were rather put on the spot, it was good to be able to put names to faces and vice versa.

Our April meeting was also unusual, with Berny Goodheart describing his book (see Adrian's review for more details). Although Berny's talk was a blatant sales pitch, it was a very entertaining one, and therefore unusually effective. It even broke through my own anti-sales defences.

Some of us in Perth are lucky enough to be looking forward to travelling to Melbourne for AUUG '94, and catching up with people we otherwise see only on the Net or in the pages of AUUGN. And we're all looking forward to hearing Gene Spafford — and with any luck Linus Torvalds too — speak here as well as at the Winter Conference. To all the people who help bring these events to AUUG's members, keep up the good work!

Janet Jackson (WA Chapter Sub-editor) <jackson@cwr.uwa.edu.au>, (09) 380 2408
From WAUG, the WA Chapter of AUUG

WAUG Meeting Review

April
The Magic Garden Explained
Berny Goodheart, Tandem Computers

Berny was flown over to Perth by Tandem so he could tell us about his new book† (and I guess demonstrate that Tandem have some very technical Unix support staff). The talk didn't cover SVR4 internals (which are the subject of the book), instead it covered the history, development and politics of the book itself.

Berny's description was entertaining and interesting, but I won't repeat it here since it's all summarised in a white paper that was handed out at the meeting (contact me if you haven't seen a copy). Berny may supply a copy to be reproduced in AUUGN.

Berny of course has high hopes for the book, believing that it will appeal to a large audience. He seemed genuinely surprised, though, when I mentioned I was reading it from cover to cover; it seems to have been developed as a very detailed reference rather than an armchair book.

Berny welcomes any comments anyone may have on the book, and everyone who reports technical errors will be acknowledged in future print runs. I was quite proud of the two or three errors I had found, but Berny suggested that comments had been "flooding in", and that a new print run (including corrections) was already on the presses. So my copy is already obsolete...

Adrian Booth, Adrian Booth Computing Consultants <abcc@dialix.oz.au>, (09) 354 4936
From WAUG, the WA Chapter of AUUG

PS: About half of the people who attended the meeting bought or ordered a copy on the spot!

CD-ROM Review

Info Magic CD-ROM Standards January 1994
InfoMagic Inc, P.O. Box 708, Rocky Hill, NJ 08553-0708, USA
<info@infomagic.com>, (800) 800-6613, (609) 683-5503

Updated every six months, or so they claim on the back of the disc.
$29.95 from Hedlams Computers Perth; probably similarly priced elsewhere

"A comprehensive collection of domestic & international communications standards and documentation...

Includes the following: ITU 'Bluebook' 1988 and 1992 versions which cover networking, telecommunications and data communications standards. Internet materials include the complete IENs & RFCs as of the publication date, Netinfo directories and templates, the Network Resource Guide, IETF and IESG minutes and reports and network maps. Major directories include: CCITT (both 1988 & 1992 'Bluebook', INET (assorted Internet related materials including RFCs), ISO (other ISO standards & drafts), WINSOCK (Windows Sockets specs and example programs) HYPERRFC (RFC's in HyperText including files & software)..."

Support for accessing and using the material on the CD via a Unix or MSDOS system is provided. A directory of MSDOS versions of Unix commands is provided.

Most material is text-based. Many Postscript documents are included as well.

Copies of *Zen and the Art of the Internet* (version 1.0), and the Internet Resource Guide are also included.

I have not had time to evaluate it properly, but thought it was worth mentioning. For those without full Internet access or even for those with, this is a useful reference. Please feel free to contact me, preferably via email, if you want any more information.

Gordon Turner <gordont@dialix.oz.au>
From WAUG, the WA Chapter of AUUG

---

† *The Magic Garden Explained*, Goodheart & Cox, Prentice-Hall 1994
ISBN 013 098138 9

CONFERENCE COMMENT: UniForum NZ'94

Brenda Lobb
<brenda@seabrook.nacjack.gen.nz>

The eleventh annual conference of UniForum NZ, held in Rotorua, New Zealand, 18 -21 May 1994, was yet another beaut., with a large and wonderful mix of techos, managers, boffins, VIPs, smoothies, systems experts, telecomms experts, network groupies, all sorts. Most gratifyingly, more than half of the 187 delegates were users (51%) as opposed to vendors (35%) and consultants (10%) - and taking into account that many of the representatives of vendor organisations and nearly all the consultants and press people present were in fact themselves computer users, a more realistic estimate would be 85%. And they were all talking to each other.

The conference theme was 'The Open Advantage' and the papers offered a hard look at the real-life costs and benefits, the technical and management issues of open information systems today. In a programme designed to cater for both technical and managerial interests, prominent local and international experts presented speeches on different aspects of open systems while users from several New Zealand organisations shared their experiences, good and bad, of migrating to open systems. Discussion from the floor was vigorous in most sessions and debate continued into the wee small hours after the strenuous social programmes. Sumptuous meals and wine are all included in the registration price at UniForum NZ conferences and most delegates live-in, giving a unique, on-going forum atmosphere to the event.

Whether because of the mix of people, the strong programme, the high standard of presentations, the very interesting exhibition, the good food, wine and entertainment, the excellent venue or just the mood of the delegates, this conference worked: the atmosphere was great.

There was a lot going on, this year, including some new features - the enhanced tutorial programme, which proved very popular, and the vendor presentations, which added some glitz and fun but aroused mixed reactions amongst delegates; it must be admitted that the anti-vendor devices (water-pistols) were not totally effective, but on occasion, delegates spoke up and halted vendor pitches. The bookshop enjoyed steady business and the free software and books were snapped up very quickly. The SIG-Bofs were well-attended even though the timing was a bit tight. The costumes at the Arabian evening were just amazing, the entertainment, food and wine all most enjoyable.

Some twenty local and multinational companies sponsored the conference. UniForum NZ'94 once again featured a multi-vendor interoperability display, coordinated by Victoria University of Wellington's Computer Science Department. Nine companies connected a variety of products to the exhibition network, and demonstrated that it could work together. The exhibition network had a 128Kb/second frame relay link to the Internet. A number of X Windows based GUI applications were demonstrated running across the network.

The star of the show had to be "Mosaic", an application that lets the user browse through a vast collection of interconnected multimedia documents scattered throughout the Internet, known as the "World Wide Web". Mosaic makes it very easy to follow references from one document to another, even when they are on different Internet hosts (perhaps even in different countries!). Even better is the fact that Mosaic is available for free from the National Centre for Supercomputing Applications in the U.S.

Another interesting demonstration was the use of the Internet to transmit real-time audio and video. For a short while, we were listening to an audiocast of a Usenix LISA (Large Installation Systems Administration) meeting taking place in San Franscisco. And for quite probably the first time in New Zealand, video from a session of a conference was sent out over the Internet so that people with Internet access who were unable to attend could watch it.

Unfortunately both these demonstrations revealed one thing. You definitely need more than a 128Kb/second Internet connection to get decent quality audio and video! Perhaps next year...

The conference questionnaire revealed that the most popular papers were John Hine's "Distributed Computing: What the Vendors Don't Tell You" and Brad Jelfs' "Novell's UNIX Strategy". Also popular were Bill Melody's keynote "Will Intelligent Networks Lead to More Intelligent Decision-making?", Robert Biddle's "GUI Display of Data with TCL/Tk", Anand Raman's workshop "Limited Internalisation - the Key to Unix Expertise" and Roger Hicks' "Software Quality Management". As usual, some presentations were classified in both 'best paper' and 'worst paper' categories. The Victoria University of Wellington stand was a clear leader in the popularity stakes for the exhibition. Also receiving honourable mention were the BCL stand and the Novell stand.

At the Annual General Meeting which took place during the conference, Dr Brenda Lobb was elected as President again this year. Dr Ray Brownrigg, Systems Administrator at the Institute of Statistics and Operations Research at Victoria University of Wellington continues as Vice President and Rob Pascoe of Koppens Pascoe Partnership continues as Secretary/Treasurer.

Members of the Board are: Kaye Batchelor of GCS Ltd, Noel Cheer of IBM NZ Ltd, John O'Gorman of O'Gorman Computer Consultants and Martin Lennon of Health Technology Lt, Julie Jones of Integral Technologies (immediate Past-President), Professor John Hine of Victoria University of Wellington, and Bruce Miller of Excalibur Consulting.

Board members can be contacted by email at:

Firstname.lastname@uniforum.org.nz

Two new SIGs, one on database interoperability (headed by John O'Gorman) and one on Internet applications for New Zealand businesses (headed by John Hine), kicked off at the conference; strong interest was also expressed at an object technology Bof session (headed by John Fisher) and a SIG may be formed later this year.

AUUG members are welcome to join UniForum NZ and any of these SIGs. Contact Brenda.Lobb@uniforum.org.nz or the appropriate John. Contrary to appearances from the above, you don't have to be called John to take part and it doesn't cost anything to join a SIG once you are a member of UniForum NZ.

Additional News from UniForum NZ[†]

UniForum NZ Offers Easy Email For Members

As a new membership benefit, from next month UniForum New Zealand is offering a very easy, minimal-cost, basic email set-up package for members nationwide.

Support for DOS, Windows, Unix and Mac systems will be provided by experts, who, if forced, are not too bad at speaking plain English to technophobes, either!

All members have to do is fill out a form telling what system they have, and they are mailed a diskette,

---

† Included for Information.

a page of notes and a phone number. They then ring that number to be talked through their initial set up. Further help is available from the same phone number, again at minimal cost.

Email Directory

UniForum New Zealand is also offering from next month a directory of member's email addresses so that members are always easy to contact, even if they change their job or move house.

If their email address changes, they only need to send one message, to the UniForum NZ administrator to change their directory listing. Anyone can then obtain the new address using the directory.

Listing in the directory is free and there is no charge to use it except for normal email costs. The directory is available via email to the general public, but the software cunningly protects members from commercial exploitation.

Background Information

UniForum New Zealand is one of New Zealand's leading computer systems user groups, with 450 members from all the major sectors of commerce, government and industry. Established in 1984 and originally centred on technical Unix issues, UniForum NZ's focus has broadened in recent years to the wider marketplace of open systems.

Membership of UniForum NZ, the only vendor-independent open systems user group in the country, has increased at 15% per annum for the last four years and the range of activities and services provided by the group has expanded markedly. Interaction with UniForum internationally has increased. Regular regional meetings, an extensive and up-to-date library, a bookshop and a monthly newsletter are provided, as well as the annual conference. New Zealand's UniForum conferences, which provide a unique forum for both novice and experienced information technology users and vendors, have become world-renowned over recent years for their big attendances, high standards and enthusiastic delegates.

AARNet

MAIL SERVICE

Dear Site Administrator,

As you may be aware, the arrangements for mailing to addresses outside Australia (and also to AARNet sites) changed in May 1991. Since then, the University of Melbourne are no longer managing the administrative details associated with maintaining this service. The AARNet (Australian Academic and Research Network) management has taken over administering the service, and are requiring all ACSnet and similar sites to register with AARNet and pay a fee for continued access to Internet mail services. AARNet have set this fee as $1000 per annum for most sites, with larger sites paying more (you know who you are).

The fee is intended to cover use of AARNet bandwidth for your network traffic. Registration with AARNet, however, provides ONLY the registration of your address in worldwide address tables - your site will be unreachable without this registration. The fee does NOT cover the costs involved in obtaining a connection to AARNet or ACSnet NOR does it include a guarantee that you can be connected or even to help you find a connection point. See Note B for some information about connection services.

AUUG as a service to its members has negotiated with AARNet to achieve a lower price for this basic address registration service. The lower price is based on the reduction in paperwork for the AARNet management authorities. The AUUG/AARNet fee is dependent on the membership status of the owner of the machine(s)/domain involved, and is currently $250 for members and $600 for non-members. As such it is a substantial discount on the AARNet fee, but only applies to sites in the AARNet $1000 category. Larger sites will need to negotiate directly with AARNet.

The address registration is for one AUUG membership year. Membership years start on the 1st January or July, whichever is nearest to receipt of your application. Sites which do not renew their AUUG/AARNet registration annually with their AUUG membership each year will be removed from the Internet tables and will no longer be able to communicate with international and AARNet hosts. Reminders/invoices will be sent along with your membership renewal.

The required initial registration form is attached below. It should be completed and forwarded to AUUG's (postal) mailing address at the bottom of the form or faxed to (02) 332 4066. If you have any queries on the AUUG/AARNet arrangements please direct them to the AUUG office on (02) 361 5994.

Regards,
Chris Maltby
AUUG-AARNET Administrator
AUUG Inc.

AARNet

MAIL SERVICE
APPLICATION

On behalf of the organisation listed below I wish to apply to be a Mail Service Affiliate Member of AARNet, and accordingly request that AUUG Incorporated arrange for the Australian Vice-Chancellors' Committee (AVCC) to maintain on my behalf an electronic mail delivery record in the Australian Academic and Research Network (AARNet) to allow my organisation to send and receive electronic mail carried across AARNet.
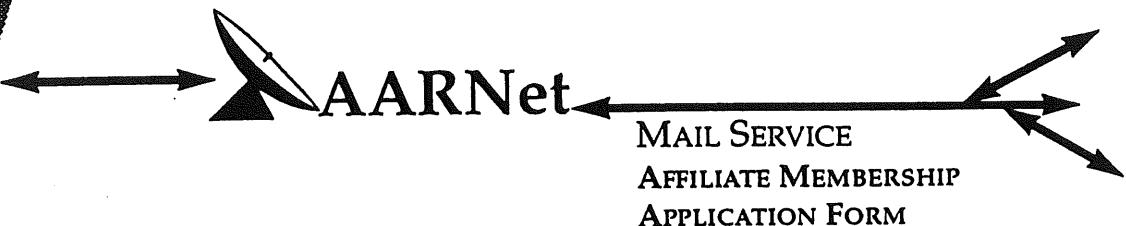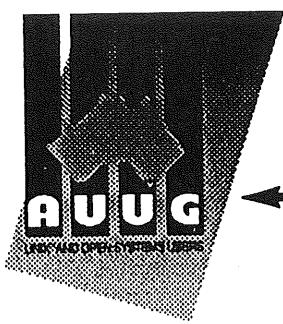
I understand that the AVCC may consult the recorded logs of my organisation's usage of AARNet facilities for 1990, and determine that I am ineligible for registration under the terms of the agreement between AVCC and AUUG Inc. I understand that AUUG Inc will invoice my organisation for this service for the calendar year 1991 and for subsequent years unless it receives my organisation's written advice to terminate the Affiliate Membership of AARNet.

I understand that the AVCC and AUUG Inc maintain the right to vary the Mail Service Affiliate Membership charges from year to year, and maintains the right to cease offering this service to my organisation at the start of any year, at their discretion. I understand that in the event of any variation of the Mail Service Affiliate Membership of AARNet, my organisation will be advised in writing by the AVCC or AUUG Inc to the address below.

I understand that in consideration of the AARNet Mail Service Affiliate Membership charge, AARNet will undertake to maintain a mail directory entry which will direct incoming electronic mail to the AARNet gateway system(s) which I have nominated below. Furthermore I accept that there is no other undertaking made by AARNet in terms of reliability of mail delivery or any other form of undertaking by AARNet or the AVCC in consideration of the payment to AARNet for the maintenance of the mail directory entry on AARNet.

I undertake that my organisation's use of the mail delivery services over AARNet will not be used as a common commercial carrier service between my organisation and other organisations receiving similar services from AARNet, nor will it be used as a commercial carrier service between branches of my organisation. Furthermore my organisation undertakes to use AARNet facilities within the terms and conditions stated in the AARNet Acceptable Use Policy. I accept the right of the AVCC or AUUG Inc to immediately terminate this service at their discretion if these undertakings are abused by my organisation (where the AVCC retains the right to determine what constitutes such abuse).

I understand that a fee is payable with this application: of $250 if the host/hosts covered are owned by a member of AUUG Incorporated, or $600 if the host/hosts covered are not owned by an AUUG member. Corporation host owners may only claim the member price if the corporation is an Institutional member of AUUG Inc. My cheque payment of either $250 or $600 as appropriate is enclosed with this application.

AARNet

MAIL SERVICE
AFFILIATE MEMBERSHIP
APPLICATION FORM

PLEASE PRINT CLEARLY!

Date:_____

Name of Organisation/Owner:_____

_____

Signed:_____AUUG Membership No (if known):_____

Name:_____Position:_____

on behalf of the organisation named above.

Address:_____

_____

_____

_____Postcode:_____

Administrative Contact:_____ Title:_____

E-Mail:_____ Phone: (____)_____

Fax: (____)_____

Technical Contact:_____ Title:_____

E-Mail:_____ Phone: (____)_____

Fax: (____)_____

Mail Delivery Information to be entered in AARNet (see Note A next page)

Domain Names Requested: _____

_____

Gateway Addresses:_____

_____

_____
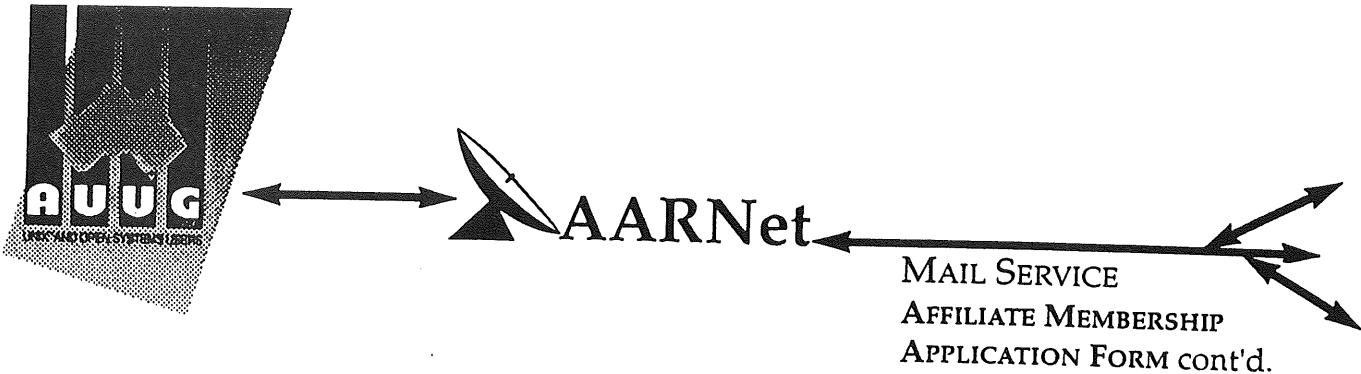
Expected Link Protocol: UUCP   SL/IP   MHSnet   Other:_____

* * * *

Send this page only to:

AUUG Incorporated
PO Box 366
Kensington NSW 2033

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

Note A. Mail Delivery Information

Two items of information are required: firstly the preferred name of your mail host (or the domain name(s) of a group of hosts) in Internet domain name system format, and secondly the name (or names) or AARNet gateway systems who will accept electronic mail over AARNet (and connected overseas networks) on your behalf and forward it to you. The primary requirement for an AARNet gateway is its ability to recognise your host/domain addresses and perform the necessary mail header rewriting reliably.

Please check with the postmaster at your preferred AARNet gateway host site before citing them as a gateway for AARNet mail delivery. For ACSnet addresses (*.oz.au), the host "munnari.oz.au" (Melbourne University) is a recommended gateway. Other possible sites include "metro.ucc.su.oz.au" (Sydney University), sirius.ucs.adelaide.edu.au (University of Adelaide), uniwa.uwa.oz.au (University of WA) and bunyip.cc.uq.oz.au (University of Qld). Note that all gateway addresses must be fully domain qualified.

Example Mail Directory Information request:

Mail addresses required:	acme.oz.au, *.acme.oz.au
Mail Gateways (primary)	gw.somewhere.edu.au
(secondary)	munnari.oz.au
(secondary)	unnet.uu.net

The addressability of your site and the willingness of your nominated gateways to act in that capacity will be determined before registration proceeds. Processing will be made faster if you contact the postmaster at your nominated gateways in advance to inform them of your intentions. Your nominated technical contact will be notified by email when registration is complete.

Note B. Getting Connected

New sites will need to find an existing AARNet or ACSnet site who will accept their site as a connection, and also select a protocol for transferring data over their mutual link. Although the UUCP package is a standard inclusion with UNIX, it is little used in Australia due to its relatively poor performance. Other possible choices for your link protocol include SLIP (TCP/IP) and MHSnet.

Among a number of organisations who provide connection services, Message Handling Systems Pty Ltd have announced a special offer on both their link software and connect time for AUUG members. For more details on this offer, contact Message Handling Systems on (02) 550 4448 or elaine.mhs.oz.au.

Book Reviews

Welcome to the latest set of book reviews. This is a bit shorter than usual, however, it is still pack with useful information. Thanks must go to David Hughes from Bond University for sending in a couple of reviews he has done for other publications. These reviews are for two Addison-Wesley publications, which we normally don't have access to. Thanks to the efforts of the AUUG Business Manager, we have almost concluded an agreement for AUUG to review Addison-Wesley publications, and AUUG members to receive discounts on their books.

This means that we will have lots of books for review. The current practice is to post a note to the newsgroup *aus.org.auug* when we have new books available. Unfortunately, this disadvantages members without network connections, or on the end of a low speed link. For people in such a position, either mail, via the AUUG PO Box, or fax me on (02) 717 9429, with your contact details and preferences.

However, volunteering to review books also imposes some obligations. Normally, I expect reviews back in about six weeks, i.e. in time for the following edition of AUUG. Lately, I've been have some trouble getting reviews back in time to fit with the AUUGN publication schedule. If you are planning to review books, remember this, and if you currently have a book for review please try and comply with these dates.

Frank Crawford

Migrating to Fortran 90

by James F. Kerrigan
O'Reilly & Associates
1993, 361 pp, $49.95
ISBN 1-56592-049-X

Reviewed by
Michael Werner
Department of Physics, University of Queensland
<werner@physics.uq.oz.au>

Migrating to Fortran 90 is another in the Nutshell Handbook series. At over 360 pages it isn't a small nut but it is just as edible as many of the other not-so-small nuts. A Canadian goose graces the front cover which is quite apt although I wonder whether a (lame) duck may be more appropriate. As the title suggests this book describes F90 and the differences between F77 and F90. Over the years many extensions to F77 have become expected features from commercial compilers. These typical extensions have in many instances become accepted in the F90 standard. In this way, the upward compatibility is in fact more generous than one may have expected.

The first chapter outlines the differences between F77 and F90 and gives a quick tour of the new features. To facilitate the use of the new features are 160 intrinsics where 75 are genuinely new F90 intrinsics. These are detailed throughout the book in its numerous examples and in the last chapter where a summary of the new F90 intrinsics is given in table form. The second chapter describes array operations which is one of the most important additions to F77 and one that has been most readily adopted in non-F90 compilers, in particular, for use with SIMD machines.

Chapters 3 and 4 discuss some new features in F90 which have existed in K&R C in part and in ANSI C for some time -- derived types and type checking of passed parameters. The latter is more verbose than ANSI C and is implemented with an interface block. The interface also serves to allow arguments of different types to be passed in the same position. However, the book doesn't point out that C++ has a much better facility called templates.

Chapter 5 discusses how F90 allows operators to be defined specifically for particular operands. This is implemented via the interface block. The author gives a table of which standard operators can be extended which is very helpful. Chapter 6 discusses modules which can be used to replace all uses of include statements and common blocks. This is an important addition to support data and code encapsulation with public and private access. Chapter 7 discusses dynamic memory management and in particular allocatable arrays and pointers which are not the same as C pointers.

My main criticism of the book is that it doesn't put F90 in perspective with the available languages today such as C++ or Sather. Also, the pedestrian style is a little over done. However, the book was intended to be a practical guide to Fortran programmers in getting up to speed in F90. Although I have no access to a F90 compiler at present, I feel that this book would be invaluable in porting in a sensible way an F77 program to F90. On the other hand, since this was my first look at F90 I plan to take a closer look at Sather and await Sather 1.0 with anticipation :-)

SCO Open Desktop/SCO Open Server System Administrator's Guide

by SCO
SCO Press/Prentice-Hall
1994, 1037 pp
ISBN 0-13-106808-3

Reviewed by
Craig Macbride
<craig@rmit.edu.au>

When I first saw this book listed I thought "Great. A book on SCO Unix system administration. I hope it's better than the one that SCO supplies with the system." I was to be disappointed. It _is_ the one that SCO supplies! As far as I could see there are only three differences. The original SCO one doesn't have the pretty colour cover, but uses thicker, whiter paper and a bolder font, making for better contrast and a book which is much easier to read in dim lighting.

This book covers how to administer the operating system, MSDOS services, and networking, including performance management and troubleshooting. It is a book which covers most aspects of system administration to some depth, but that depth varies greatly from subject to subject.

In the section on shutting down and rebooting the system, it tells us exactly what files in the security database are necessary when the system boots and what action the system will take if they are missing or corrupt at boot time.

Unfortunately, there are many other sections in which the information given is nowhere near as detailed. Some are so shallow as to be practically useless.

The network tuning and troubleshooting section, for example, is hopeless. Its only method for configuration of STREAMS parameters is to start with the defaults and, buffers of a particular size are running out, reconfigure the kernel to increase them by 50% and reboot. No attempt is made to give any way of estimating what a sensible number should be in the first place. Given the defaults which are set up by the system, the given method could mean rebooting the system a large number of times over a period of weeks once the system is running live. This is totally unsatisfactory in a commercial environment, which is what SCO are supposed to be aiming at. A sensible means of estimating resource usage would enable a system to be set up properly before the system goes live.

The very thin section on network tuning and troubleshooting goes on to describe the fields which are reported by the simple options of the "netstat" command. It ends up by giving an example of the output from the statistics option, which lists about 60 categories of statistics for IP, ICMP, TCP and UDP packets. Not a single word appears describing what any of them mean. Neither is there any description of which ones matter nor what to do if any of the error categories are non-zero!

In the modem section, required settings are suggested for any modem that you configure for dial-in/dial-out using a dialer program other than those supplied with the system. One of the recommended settings is "ATS0=1", so that the modem answers on the first ring. They don't tell you that many of the dialers that SCO provides, such as for a standard 2400 bps Hayes compatible, don't conform to these recommendations! In this particular case, the SCO-supplied "dialHA24" program to dial out disables auto-answer altogether on the modem it uses by sending it an "ATs0=0", which is, of course, a real pain if you are expecting to ever get incoming calls again!

In summary, this manual contains lots of useful information on administering a SCO system. It does that job fairly well overall. It is let down badly by glaring gaps in the troubleshooting and tuning areas, especially where networking is involved. If you set up a SCO system and the only problems you ever have are with printers, tape units and the like, this manual will be fine.

If you have any difficulties with running more complex portions of the system, such as TCP/IP, or using such programs as "ct", you will probably find that this manual stops short of giving the answers.

Given the greater readability of the SCO original, which can be ordered as part of the operating system, use the SCO original rather than this one if you can. And try to find a book which covers the finer details that are skipped.

Advanced Programming in the UNIX Environment

by W. Richard Stevens
Addison Wesley
744 Pages Hardback
ISBN 0-201-56317-7

Reviewed by
David J. Hughes
Bond University
<bambi@Bond.edu.au>

The name W. Richard Stevens is one that is familiar to most people who are involved with UNIX Network Programming. In fact, his book "UNIX Network Programming" is viewed by some as the bible for the topic. Well, he's been at it again and this one is no less biblical than the last.

"Advanced Programming in a UNIX Environment" is to systems programming what his last book was to network programming. It is a well structured and informative work covering topics from basic file I/O, through process control, to pseudo terminals (and everything in between). It is jam-packed with example code and explanatory text to guide the reader through the intricacies of UNIX.

In my opinion, one of the key features of the book is its constant reference to the relevant standards - both ratified and de facto. As a library function or system call is discussed, references are made to the relevant standards from ANSI, X/Open, and IEEE's POSIX, as well as 4.3+BSD and System V Release 4. The reader is left in no doubt as to the possible portability problems associated with a particular feature of the operating system.

The book starts at an introductory level discussing the basic file I/O facilities provided

by the UNIX kernel. It slowly but surely delves deeper into the areas associated with systems programming including asynchronous I/O, memory mapped files, terminal I/O, signal handling and inter-process communication. During the journey, enough detail is given to the inner workings of the kernel to aid the understanding of the material at hand. It does not provide information relating to kernel algorithms or data structures although the primary focus of the book does not warrant that level of detail.

"Advanced Programming in the UNIX Environment" provides a well structured course in UNIX systems programming for somebody wishing to teach themselves the trade. It's informative style and copious example code would also make it an ideal text for a tertiary course on the subject. That said, it is still one of the best instructional references I've seen for UNIX sys-prog's and I'm sure that it will become a well thumbed addition to many bookshelves around the country.

Operating System Concepts
Fourth Edition

by Silberschatz & Galvin
Addison Wesley
780 Pages Hardback
ISBN 0-201-50480-4

Reviewed by
David J. Hughes
Bond University
<bambi@Bond.edu.au>

"Operating System Concepts" has long been known as an informative book for the teaching of operating systems. The Fourth Edition builds on the success of the earlier editions and expands on the material presented. It achieves this by covering in great detail the techniques employed by operating systems that are in common use today.

The book is aimed at a serious tertiary course in operating systems design and implementation. The topics covered include process scheduling and synchronisation, memory management (including demand paging), file system implementation, security schemes, and distributed systems including distributed file

systems. Its coverage and detail of the topics is quite complete including algorithms and pseudo code for components of an operating system kernel.

Throughout the book, constant references are made to current-day operating systems, including Solaris 2, MS-DOS, OS/2, and the Macintosh OS. Special detail is given to both 4.3BSD and Mach with an entire chapter devoted to a case study on each.

An interesting facet of "Operating Systems Concepts" is the final chapter in which a case study of The Nachos System is given. Nachos is an "operating system" written by the authors for use in their operating systems classes at the University of Texus at Austin and at Brown University. It is a semi-complete kernel offering a thread manager, a file system, program execution, and a simple network mailbox scheme for inter process communication.

The thing that sets Nachos aside from other operating systems is that it is a high level simulation in which the kernel runs as a user process on a UNIX system. Nachos sits over a simulation of a typical computer system and offers a software emulator for the MIPS R4000 chipset. Therefore, Nachos can execute R4000 binaries, controlling the threads of execution, without having to monopolise a particular machine. From an educational point of view, this is wonderful as a class full of students can each be running and developing their own kernel on a standard time-sharing UNIX host. The authors freely provide the source code for Nachos and a set of predefined assignments to anyone who wishes to use them. In fact, the source for the current snapshot of Nachos is available via anonymous FTP.

The Fourth Edition of "Operating Systems Concepts" differs from previous editions due to its earlier coverage of light-weight processes and threads, its expanded coverage of memory management, a more detailed coverage of real-time and multiprocessor systems, and updated references to the operating systems that are in use today.

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

AUUG BOOK CLUB
&
PRENTICE HALL AUSTRALIA

20% DISCOUNT TO AUUG MEMBERS

Please send me a copy/copies of the following book —

☐ ***SCO Open Desktop / System Administrator's Guide***
ISBN: 0131068083, Paper, 1994 , RRP $92.95

*Deduct 20% from listed retail price

Name: _____ Position: _____

Organisation:_____

Address: _____

_____ Telephone: _____

☐ Please send my book/s on 30-day approval (tick box)
Enclosed cheque for $ _____ (Payable to 'Prentice Hall Australia')
☐ Charge to me **OR** ☐ Company purchase Order No. _____
Please charge my: ☐ Bankcard ☐ Visa ☐ MasterCard
Credit Card No: ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐

Expiry Date: _____ Signature: _____

Mail or fax completed order form to Prentice Hall Australia, PO Box 151,
Brookvale NSW 2100
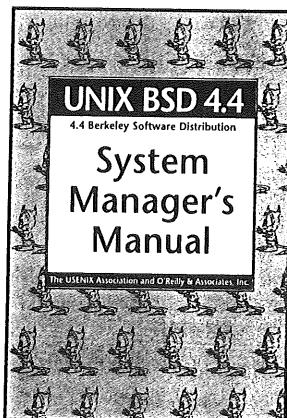
OR ✆ Use our FAST PHONE SERVICE by calling Sandra Bendall SYDNEY (02) 939 1333


A Report on LISA 93. †

David Jones
Department of Maths & Computing
University of Central Queensland
Rockhampton
d.jones@ucq.edu.au

Do you ever get lonely being a sys admin? Are you the only sys admin (or one of the few) at a site, suburb or even region? Tired of fighting a loosing battle against people who don't understand the hassles and pitfalls of your job?

One solution to these woes is to attend a specialist systems administration conference. At a sys admin conference everyone there suffers from the same problems, the same users and the same vendors you do. It is an excellent chance to whinge about your situation with someone who can understand and sympathise. It also provides an opportunity to discover unique solutions that have been used to fix problems you've faced (or are lurking just around the corner).

Last year I had the chance to attend two specialist systems administration conferences. SAGE-AU '93 in Melbourne and in November I had the good fortune to attend the Usenix Association's LISA conference in Monterey, California. LISA (Large Installation Systems Administration) has developed into the major Systems Administration Conference in the United States (if not the world). At this conference there were about 1000 attendees from numerous countries including the USA, Germany, the UK and three from Australia.

The conference is no longer restricted to large sites and encompasses all shapes and sizes of systems administration. Attendees ranged from part time systems administrators at small sites to the full time guru looking after hundreds of machines. There was something to be gained by the whole variety of attendees.

LISA 93 was a five day conference including two days of tutorials and three days of technical sessions.

Tutorials included
* an introduction to Perl and Network Programming in Perl by Tom Christiansen
* an overview of internal Unix security by Rob Kolstad Fairly standard security material delivered with what appears to be the usual Kolstad panache.
* Topics in Systems Administration.
  Covering topics like integrating non-Unix Systems, managing help desks, SLIP/PPP, network performance, Usenet News, routing, introduction to Cisco IP routing and administering DOS-based PCs
* a session on how to administer heterogeneous systems

* Introduction to Sendmail configuration and administration taken by Eric Allman the author of Sendmail.

The technical sessions included talks based on referred papers, various invited talks and some panel sessions. Some of the issues covered were
* Using MUDs (you heard right Multi-User Dungeons the scourge of AARNET) as a communications tool for systems administrators. It provides the sys admins a nice central virtual environment in which they can communicate even when they are in physically separate locations.
* A talk on the evolution of Intel's corporate polices on the use of Internet by staff.
* A presentation from one of HP's research centre's where all their users have the root password and in some cases the sys admins don't!
* Various software installation, customization and distribution schemes.
* Panel sessions on single person administration and on managing very large sites.

As with most conferences there were some very good sessions and some not quite up to expectation. Overall however at any one time there was usually something worthwhile.

On a more light hearted note there was also an element of "star gazing" involved in attending a LISA conference. The opportunity presents itself to put faces to names you've seen in books or on the net. One example was Eric Allman. The O'Reilly Sendmail book was launched at the conference and all the conference attendees received a t-shirt with the book's cover design emblazoned on the front. Through out the conference "star-struck" sys admins could be seen nervously approaching legendary guru and author of Sendmail (and co-author of the book) Eric Allman asking him to sign their t-shirt. (Another "highlight" of my visit to the US was the opportunity to see Luke Perry filming a scene from Beverly Hills 90210 on Santa Monica beach. Impresses some of my younger students no end.)

Another advantage to attending a conference is the chance to hear some of the gossip making the rounds. For example one of the other "celebrities" sighted was Evi Nemeth co-author of the Unix sys admin bible *Unix Systems Administration Handbook*. It was mentioned at the conference party that Evi was putting the final touches on a new edition of the book.

Overall LISA '93 was a very useful and enjoyable conference. If you are a Unix sys admin I think you would find it well worth your while attending. Of course there is no need to make the trek to the United States. You can always attend SAGE-AU's conference (information on this year's conference is included in this newsletter). It offers the same advantages as LISA but in a local setting.

† This is reprinted from The Australian Systems Administrator (SAGE-AU's Newsletter), Volume 1 Number 1

The Free Software Top Ten †

David Jones
d.jones@cqu.edu.au
Department Maths & Computing
Central Queensland University

Free software can be a lifeline for a Systems Administrator. There are thousands of highly intelligent people through out the world employed as Systems Administrators. If you've just been lumbered with a new problem then chances are that it has already been solved. In most cases you will find that the software needed to solve the problem is available off of the Internet free of charge.

Managerial types can sometimes have difficulties grasping the concept of free software. They can be full of worries like "What about viruses?" and "Who is going to support the software if it is free?". For one Sys Admin I know from Melbourne the only way management would allow the use of free software was for him to purchase a CD-ROM containing the software. Since money was spent (even though the money was for distribution and not for the software) then it had to be alright.

Like many questions "What are the top ten free pieces software?" is subjective and there is no correct solution. The answer depends on your background, environment and needs. The following includes Top Ten Lists from four people (including myself). Thanks go to Janet Jackson, Nick Frisinia, Lyn Kempter for contributing their personal lists..

My Top "Ten"

Leading off my list is Linux the free Unix operating system that was developed on the Internet. A distribution of Linux is one of the biggest collections of free software around (it usually includes 90%, or more, of the products mentioned in this article). This concentration of quality software is what allows Linux to compete against the commerical versions. Apart from doing the occasional flirt with the Department's Suns or the Uni's Ultrix box most of my Unix work, including that of my project students, is done on PCs running Linux. It is simply the cheapest (legal) way I know of getting a Unix box.

Second on my list is the software that makes World Wide Web (WWW) possible. WWW is the latest greatest navigation tool for the Internet that can be used for much more. All the software involved with WWW, clients, servers and other miscelanous programs to support WWW are all freely available. The information available on WWW, WWW's capabilities as a standard information gathering and dissemination tool and the free software all combine to make this number two on my list. (Keep an eye out for a SAGE-AU WWW server sometime in the next year.)

Number three is gcc the Free Software Foundation's (FSF) C, C++, Objective C and even Ada compiler. gcc is the second thing I install on any Unix box. The first thing I install on any Unix box is gzip. The compression program from the FSF that gives better compression than compress and is used to compress all of FSF's software. gzip is first because I can't install gcc or bash until I do.

Coming in number ten on my list is bash, the Bourne again shell. The ability to edit previous commands using the cursor keys is to much to ignore for a lazy person like myself. For csh afficionados I suppose tcsh would be the equivalent product.

As the saying goes "All work and no play make.....". I couldn't come up with a list of my favourite free software without including the demo version of Doom. A game designed to appeal to my adolscent tendencies and fulfill those violent urges caused by students. It almost impressed me enough to go out and buy the game.

1) Linux
2) World Wide Web software
3) gcc
4) top
5) sudo
6) expect
7) perl
8) elm
9) gzip
10) bash
11) the demo version of doom

Janet's Top Ten
jackson@cwr.uwa.edu.au

Having moved from an environment that ran on free software, to one that had almost none before I arrived, I have a pretty good idea of the things I, personally, simply cannot manage without: perl and MH (Rand's Message Handling system, God's own mailer). I get pretty mad if I don't have less, gzip, and MIT X11. My most used X clients are xcalendar

and xpostit. I also like to have TeX (for policy documents, user manuals, what-have-you), ghostscript and groff.

For Unix systems administration: perl (for when shell isn't up to it). gcc (for the few things perl can't handle). sudo (so you can avoid giving out root access without having to resign). CAP (for connectivity with Macs). Amd for automounting. OSU Backup (which I am in process of installing) also looks like it's going to come in handy.

If you want an actual top ten -- here's a list, roughly in order of must-have-ness:

1) perl (I could probably live without the rest, but perl is indispensable)
2) sudo   (If it wasn't available I would have had to write it)
3) MH
4) less
5) X11
6) gzip
7) TeX
8) CAP
9) xcalendar
10) ghostscript

Nick's Top Ten
nick@pyramid.com.au

1) top
2) COPS
3) CRACK
4) TCP Wrappers
5) some sort of "in-house" monitoring script that does exception reporting
6) monthly and annual trimmimg and archiving of logfiles by some in house script.
7) something like sudo and/or priv to delegate duties to other staff.
8) elm or the like as a mailer. (even ida sendmail as the configurer)
9) tapescan to give details about tapes.
10) ask steve landers to put into the public domain his terminfo configurer. doing it by hand is always a pain.

Lyn's Top Ten
l.kempter@ucq.edu.au

1) top (an all-time fave!)
2) sudo
3) elm or pine
4) expect
5) TCP Wrappers
6) perl
7) dig

8) cops
9) inn
10) tin

THE FINAL TOP TEN

Using a deeply scientific accounting method that is to complex to explain. The following Top Eight list has been produced. All the other software received votes from one person only and thus are deemed by the judge to be ineligible.

1) top
2) perl
2) sudo
4) elm
5) TCP wrappers
6) cops
7) expect
8) gzip

Where to get them

All of the top eight and the majority of the other applications mentioned in this article are all available off of the Internet. Using Archie is the best method to locate an exact location.

Included below is the list of locations from Australian ftp sites for the top eight.

1) ftp://ftp.jcu.edu.au/pub/ultrix/top-3.2.tar.gz
2) ftp://archie.au/gnu/perl-4.036.tar.gz
3) ftp://ftp.connect.com.au/pub/security/ sudo.v1.2.tar.Z
4) ftp://archie.au/usenet/
   · comp.source.unix/volume6/elm/*
5) ftp://ftp.jcu.edu.au/pub/ultrix/ tcp_wrappers_6.1.shar.Z
6) ftp://ftp.connect.com.au/pub/ security/cops_104.tar.Z
7) ftp://ftp.connect.com.au/pub/ misc/expect.tar.Z (needs tcl)
8) ftp://archie.au/pub/gnu/gzip-1.2.4.tar.gz

!AUUGN

TELEPHONE
345 1844

TELEGRAMS
UNIMELB PARKVILLE

University of Melbourne

DEPARTMENT OF COMPUTER SCIENCE

Parkville, Victoria 3052

2nd November, 1978.

Mr. Ian Johnstone,
AGSM,
University of New South Wales,
Kensington,
NSW.

Dear Ian,

Here is the tape that I promised you in our telephone conversation on the 21st.
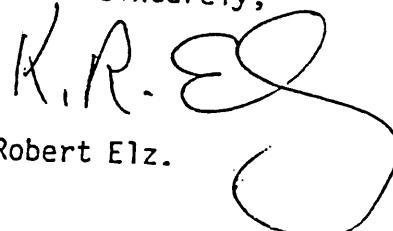
It contains 5 files:

1) <u>dtp</u> format - of which a directory listing is also enclosed.

2) <u>tp</u> format, in case our <u>dtp</u> format isn't the same as yours.

3) & 4) copies of 1) and 2) in case of parity errors etc.

5) dtp.c in dd output format:
            dd if=dtp.c of=mt0
   - in case our <u>tp</u> format isn't the same as yours
     (a possibility I have some evidence to support).

You will note that most programs included here are simply modifications of programs from other places. I have tried to select programs of which I believe our modifications to be useful (which is not to say anything about the program itself).

Further to my letter of the 15th, I can now add that the Gandalf PACX is now installed and operational, and looks like being a useful addition to UNIX.

Yours sincerely,

Robert Elz.

I couldn't survive without my Internet connection!

Charles Cave
Customer Services Manager
cmc@sydney.unidata.oz.au

Unidata Australasia
Level 2, 280 Pacific Highway
Lindfield, NSW, 2074
Phone (02) 416 6877
Fax (02) 416 2086

Abstract

This paper is a case study in the technical and financial benefits gained by a software company by implementing an Internet connection for electronic mail. Following the connection of the Sydney subsidiary and the US head office to the Internet, many benefits have been gained in the areas of technical support, dissemination of information, distribution of software patches and documentation.

Facsimile charges and telephone usage has dropped giving financial and time saving benefits. Unidata Australasia has been able to take advantage of the abundant software available through Internet, in particular the GNU packages from the Free Software Foundation (FSF), and information and resources available through the USENET news system.

This paper is aimed at people who don't yet have an Internet connection, summarising the benefits to justify its installation. Experienced Internet users may also benefit by finding new uses for this powerful resource.

Contents

Background on Unidata the company and UniData the product

Unidata Inc was founded in 1984 in Denver, Colorado and first released its UniData Nested Relational Database product in 1988. The company has several offices in the USA, subsidiaries in the United Kingdom, Australia, France, and distributors in Germany, South Africa, South East Asia and South America.

The product is ported to most Unix platforms and Digital's VMS operating system, and there are over 2000 sites world-wide. Distribution is through a network of value-added resellers (VARs) who combine application packages and services with UniData providing a total solution.

The Australian subsidiary opened in early 1990, initially as a joint venture company, with one sales person and myself as technical and pre-sales support specialist. New staff have joined, including two computer science graduates, and the company now has a total of twelve employees based in the northern Sydney suburb of Lindfield.

How I used to do support

Customer support was a major component of my job, and problems were reported by telephone and facsimile. Problems which could not be solved locally were passed on to the US customer support department . In many cases sample programs were printed and program output was faxed to Denver to accompany the problem reports. Communications were then received overnight via facsimile.

Occasionally a customer would supply a cartridge tape of data files, and programs for examination usually with a yellow Post-It note attached requesting the return of the tape! Further delays and shipping expenses were incurred if the tape had to be sent to Denver to assist in the resolution of a problem.

Tapes were also received irregularly from the head office containing copies of the database of known problems. Production of this tape was complicated by well-meaning system administrators forbidding access to the tape drive because the computer was located in the computer room, and obtaining cartridge tapes (because of their expense) also delayed the process.

Production and transmission of a one page support fax involved the following steps:-

- Type the fax document and print on the laser printer
- Hope that the laser printer has enough paper and toner
- Place the printed page in fax machine, dial number and hope that the head office fax has not run out of paper during the night preventing receipt. This was a regular occurrence.
- Punch holes in the printed page and store in a folder
- Facsimile handshaking and transmission of a one page takes approximately 45 seconds to send, and costs around $1.20 a page. This cost does reduce when multiple pages are sent, but not below $1.00 a page.

The logistics of creating and sending faxes often discouraged me from sending questions and I would defer these questions to another time. At the end of each month, our company's telephone bill was high because of the facsimile charges in addition to the regular telephone calls. In addition, time and money had been spent sending the faxes, feeding and wearing out the laser printer and filing of incoming and outgoing faxes.

Consider the time taken when a three page C program was received by fax and then had to be re-keyed and debugged!

A major problem we encountered with faxes is the lack of control over their delivery to their intended recipient. Sometimes they are mixed up with someone else's fax and lost forever. On other occasions, someone else may pick up the fax thinking they are more suited to handle the issue or question. Re-sending faxes adds more unnecessary charges to the phone bill.

The connection was made

In late 1992, the Denver office installed a SL/IP connection (Serial Line Internet Protocol) to the Internet using the Colorado Super Net via dedicated modem. This event was not widely publicised and only found out by accident, despite my pushing the head office to use electronic mail for two years.

Unidata Australasia as a corporate member of AUUG took advantage of the special price of MHS software (for $495) from Message Handling Systems, and an MX (Message Exchange) record with AARNET for an annual fee of $250. This has proved to be a very cost effective service with an hourly connection charge of $8. Our monthly charges are around $100.

Our MHS software runs on a Silicon Graphics Indigo workstation with Irix 4.0, and a 9600 baud Interlink Fastbit Model II modem. The software has been configured to dial up the MHS gateway machine at 7 am, 9 am, 11 am 3 pm and 8 pm. This has proved to be of sufficient frequency as urgent mail can be sent by issuing the **netcall tmx** command at any time.

Following the installation of the software, test transmissions were made with Denver and after a few hiccups of address registrations and misunderstandings on my behalf, the two offices were talking.

Streamlining the internal systems

The internal problem reporting system used for Customer Services was modified to send the call submission reports with the MHS **netmail** command instead of producing hard copy for the fax. I estimate one page of text taking 5 seconds to transmit over the modem at a cost of around two cents.

Many benefits flowed on from this new system:

- Outgoing and incoming faxes no longer had to be filed in an ever growing collection of Lever-Arch files. Instead, the information was saved in electronic folders which can be searched quickly and re-transmitted if required.

- Sample programs and command output can be included in an email message with little extra cost. A twenty page fax costs around $20 to send, whereas a twenty page email message costs around 50 cents.

- Source programs received from head office can be used immediately without time taken to re-key.

Services provided by Internet

An abundance of information is now available in books and in *Unix World* magazine, so I won't go into too much detail. The four main services offered by Internet are:

- Electronic Mail
- USENET newsfeed
- File transfers (anonynous ftp)
- Telnet access

Electronic mail

Initially we hobbled along with the Unix **mail** and **mailx** commands (for reading mail) coupled with the **netmail** command of the MHS software for sending mail. The two major problems encountered were a runaway **sendmail** daemon and the difficult to use mail and mailx utilities.

In early 1993, a new graduate employee brought a copy of the **elm** mailer program which was ported and is an excellent piece of public domain software. A template **sendmail.cf** file was supplied by Silicon Graphics support and the same graduate given the task of localising this file. It wasn't long before we had a functional system with no runaway sendmail daemons and a user-friendly mailer program.

The main users of electronic email in the Denver office were the technical support staff and working relationships developed through the regular exchange of email messages. Despite the implementation of a voice mail system in head office, electronic mail is effective for asking questions. A quick question can be constructed with **elm**, assisted by the useful feature of **aliases**, and queued for transmission. Some mail has been lost, and some recipients are not regular email readers, but a brief voice mail message will ensure the message is received.

Personal computers running Microsoft Windows and Microsoft Mail in the head office have paved more inroads to corporate wide acceptance of electronic mail and eliminating the myth that electronic communications are the sole domain of Unix wizards.

Electronic mail is a mainstream corporate tool and here to stay.

In our customer service brochure, we publish an email address for support questions (**support@sydney.unidata.oz.au**). This address has a **.forward** file redirecting the message to a staff member. Nearly all hardware vendors have connections to Internet, as well as some of our business partners and customers. Our customer support staff enjoy supporting these people as they find responses are much faster and the game of telephone tag is eliminated. In many cases, conversations are easier with electronic mail than the telephone when only technical information has to be exchanged and less time taken.

A major benefit of electronic mail for support is the ability to send the results of **ps**, **iostat**, **vmstat** and **sar** commands for analysis. Searching with **grep**, **vi** or **emacs** is so much easier than poring over a dozen pages of curled up fax paper. Sample programs and program output can be mailed, and of course, forwarded on to head office.

Network news feed (USENET)

Network news is the Internet equivalent of a discussion group or a "bulletin board system" (BBS) like those on Compuserve or private dial up facilities. Discussions are organised into newsgroups, and articles (or postings) can be read with a newsreader program. The network news gives the user access to a large group of people around the world to answer questions and for participation in discussions.

The **cnews** software package. used for splitting up the incoming newsfeed was partially installed on the Silicon Graphics and a set of newsgroups selected for receipt. Lack of time and a certain amount of frustration porting the **cnews** package produced a system of home brewed programs which posts the news into directories and sends selected items to nominated users with the **mail** command. The bulk of the news items are read with **emacs** and a variation of the **dired** macro.

The newsfeed can of course fill disks very quickly and one has to be very selective about which groups to receive. My selection of newsgroups is listed on the next page with a brief description. News articles can be posted using an email gateway at MHS.

Newsgroup Name	Description	Benefits
aus.auug	AUUG Inc - Australian Unix User Group	Announcement of activities and general information not found in the newsletter
aus.jobs	Jobs available and wanted	If you need graduates for your company, this group is widely read in academic circles
aus.archives	Announcements of available software	Public domain software for downloading with anonymous ftp
comp.databases	Database and data management issues and theory	Our company is in the database market, and information can be gleaned on competitive product information. A spin-off group comp.databases.pick was formed in May 1993 covering our market sector
comp.databases.pick	Discussions on databases compatible with PICK`	Information both technical and marketing is available on our competitors and our end-users. More details later in this paper.
comp.lang.perl	Discussions on the perl language	Questions on perl are answered and questions can be posted
comp.sources.unix	Postings of complete Unix oriented software	Sometimes useful software appears in this newsgroup
comp.sys.mac.apps	Discussions on Macintosh applications	The author uses an Apple Macintosh for work-related-desktop publishing projects and useful information appears about new products, both commercial and shareware.
comp.sys.sgi	Silicon Graphic's workstations and software	We have a Silicon Graphics machine, so this information is useful
comp.text	Text processing issues and methods	Of interest
comp.text.desktop	Technology and techniques of desk-top publishing	Of interest
comp.unix.admin	Administering a Unix-based system	General knowledge
comp.unix.misc	General discussions regarding Unix	General knowledge
gnu.announce	Announcements of new gnu software (Free Software Foundation)	New releases are posted to various archive sites
news.answers	Repository for periodic USENET articles	An essential newsgroup for receiving Frequently Asked questions documents
rec.humor.funny	Moderated jokes	Computing professionals require a regular diet of humour and this group is low volume but generally high quality. The rec.humor group is high volume with occasionally funny postings but requires too much time to digest

Which newsgroups to get?

When selecting the newsgroups the major factors are as follows:

*   Communications costs (currently $8 per hour dialup)
*   Disk space requirements
*   Time taken to read the articles (the most important factor).

Attempting to read a large number of newsgroups is similar to receiving every newspaper and magazine. No-one has the time to digest all this stuff <u>and</u> get their job done.

The **comp.databases.pick** newsgroup has been a good method of posting information to our customers who read this group. I prepared a *Unidata Frequently Asked Questions* document (FAQ) and posted an announcement to the group on how to subscribe to this document. We now have over 150 recipients in Australia, USA, UK, Europe, NZ and Canada and the number of recipients is growing steadily.

I have found the USENET news to be of great value in getting questions answered and locating resources such as programs. A question can be posted to the newsgroup, and answers will appear unless an email response was requested.

It is wise to read a group for several weeks before posting anything, as *Frequently Asked Questions* documents will be posted which answer common questions. There is no formal management of the newsgroups, but inappropriate postings will be quickly flamed.

The next step for me is to set up an **email server** (or listserver) where people can send requests for documents, programs and subscription requests. Email servers are common around the Internet and will greatly eliminate my administrative workload. An email address is set up and a daemon process monitors the mail box looking for requests and taking necessary action, namely replying to the requests.

If someone can give me advice on public domain listserver I would be most grateful.

File transfer

Electronic mail is limited to sending files of printable characters and reasonable line lengths. The method of sending a non-text file is to use the **uuencode** and **uudecode** Unix utilities.

For example, to send the executable **myprog**, the following steps are required:-

```
uuencode myprog myprog > outfile
```

The file **outfile** is then sent with the **netmail** command or included in a message created with the mailer program (such as **elm**).

The recipient saves the message, deletes the mail headers and runs the command:

```
uudecode filename
```

Sending a directory structure requires the extra step of creating a **tar** file which is usually compressed.

```
tar cvf /tmp/tarfile directoryname
compress /tmp/tarfile
```

The tarfile.Z (compressed file) is then uuencoded and sent by email.

The recipient saves the message, deletes the mail headers, runs the commands:

```
uudecode filename
uncompress tarfile.Z
tar xvf tarfile
```

MHS software provides an Internet ftp gateway with the command **netfetch**. For those sites with a connection supporting anonymous ftp, this command is not required..

The **netfetch** command can be used for retrieving files as well as directory listings, and despite the batch style usage, it has proved to be a most useful command.

Sample commands to retrieve a directory listing and a file are:

```
netfetch -lv nodename pub
netfetch nodename pub/text/document.txt.Z
```

Email gateways exist for retrieving files through email such as **mail-server@pit-manager.mit.edu**. Just send it a message with the word **help** in the subject line.

Publicly available on-line archives of data and software are the gold mine of the Internet. The Catalog section of Ed Krol's book [3] gives a selection of resources. The primary resources I access are software from the Free Software Foundation, Macintosh utilities and TrueType fonts, and text files (Frequently Asked Questions documents). More details will be given in the presentation of this paper.

File compression methods

Once you start mining these gold fields, the subject of compression must be tackled as most archives have been compressed to save storage space and reduce network transmission time. The most common compression programs are shown in this table and should be studied.

Compression Program	Decompression Program	File Suffix	Typical File Name
compress	uncompress	.Z	rfc.txt.Z
gzip	gunzip	.z	testfile.z
Stuffit	Stuffit Expander	.sit	program.sit Macintosh format
PackIt	unpit	.pit	report.pit
PKZIP	unzip1	.ZIP	package.ZIP DOS format
zoo210	zoo210	.zoo	picture.zoo
gzip	gunzip	.z	FSF software
BinHex	BinHex BinHqx	.hqx	Myfont.sit.hqx Macintosh format

Free Software Foundation

The Free Software Foundation (FSF) is an organisation devoted to the creation and dissemination of software that is free from licensing fees or restrictions. Software is distributed under the terms of the *General Public License* which also provides a good summary of the Foundation's goals and principles.

The FSF has developed the GNU **emacs** editor, in addition to replacements for many Unix utilities and many other tools. For example, **flex** (replaces **lex**), **bison** (replaces **yacc**), **bash** (a superset of **ksh**, **sh** and **csh**), **gawk** (replaces **awk**), and the very popular **gcc** C compiler and **gcc++** compiler.

The **perl** "Swiss Army Chain Saw" tool is distributed under the GNU Public License, and rapidly becoming an essential tool at Unix sites.

UniData Documentation

Our office became involved in proof-reading documentation for the head office technical writers. I volunteered to review this material, and packages of photocopied pages soon appeared in our courier shipments.

I would review this material usually at home, and write up my comments at the office and email the review. Naturally, deadlines were tight and made difficult by the delays in photocopying the documentation, and sending the package by courier.

The Silicon Graphics machine has the Adobe Systems Display PostScript Document Previewer program, **xpsview** which takes PostScript files and displays them on the workstation. I made the suggestion to the technical writers to email me PostScript files created with Framekaker which I viewed with **xpsview**, and wrote my comments in another window running **elm**. Turnaround time was quick, and no paper was used. An alternative program to **xpsview** is the FSF's **ghostscript** and **ghostview** programs.

The ability to utilise the global resources of the company for this task was easily realized by utilizing electronic mail. The reverse procedure is also possible of sending text files to the technical writers for publishing, eliminating any re-keying.

Problem Database

The database of reported technical problems used by the head office customer support staff is replicated on our machine. Each week, a file of changes is received by email and used to update our copy. This procedure used to be a cartridge tape sent by courier every few months which was far from satisfactory. Now, our customer support staff have access to the latest details on the product and known problems.

Software distribution

Occasionally a product component (e.g. libraries or executables) has been corrupted. Software is distributed from Denver, so there are always delays in shipping. On a couple of occasions, a uuencoded file has been sent

Distribution of word-processing documents and spreadsheets

Documents created with Windows applications can be copied to a Unix machine, and uuencoded for transmission. This has been useful for sending documents which require some localization, e.g. price books and product availability schedules.

The future

Each morning I look forward to seeing what new email messages have arrived, particularly from people I do not know. Generally these are requests for my FAQ document and new employees in the head office who are starting out with email. Our support staff are regular readers of email with the **xbiff** program running on their X-terminals to alert them to new mail.

Our office is switched on the benefits of email, with our Internet addresses on our business cards.

The challenges ahead are:-

- Get the remaining subsidiaries connected. The United Kingdom office will be connected by the end of March 1994.

- Encouraging more of our business partners to get connected.

- Changing the corporate culture in head office to use electronic mail for distribution of memos, information and status reports across the company.

- Implementing an email server (called **infobot@sydney.unidata.oz.au** after the name used by **Wired** magazine) for information and program dissemination.

- Build the infrastructure to support a global 24-hour a day support network.

References

[1] Glister, Paul. *The Internet Navigator: The Essential Guide to Network Exploration for the individual dial-up user*. John Wiley and Sons, Inc, 1993

[2] Kehoe, Brendan P. *Zen and the Art of the Internet: A Beginner's Guide to the Internet*. Chester, PA: Widener University, 1992.

[3] Krol, Ed. *The Whole Internet: User's Guide & Catalog*. O'Reilly & Associates, Inc. 1992

Firewalls and Proxy Servers

Access and Security on the Internet

Tony McGrath

Uniq Professional Services Pty. Ltd.

Abstract

One of the more prevalent activities on the Internet in recent times has been the creation of firewall systems to restrict access between systems. A number of different facilities have been created to help network administrators to create the level of security they need and also allow limited access to services for trusted users. One of these systems is SOCKS, which comprises an authentication daemon and a limited API that allows BSD Socket programs to be created that can access services through the SOCKS daemon. One of the major restrictions of the SOCKS package is the implicit assumption that only a single firewall system exists between the client system and the server system, an assumption that is no longer valid given the proliferation of firewall systems. This paper will address the issues that arise when multiple levels of firewalls are established on the network, but it is still required to permit client applications to access services that exist on systems that have multiple firewalls between them. A limited implementation of this facility, called PROXY, is also described.

1.0 Introduction

There has been considerable movement during the last twelve months in the development of security facilities to restrict access to systems on the Internet. These facilities, known as firewalls, generally work on the principle of providing a system that physically de-couples network traffic between network segments. Within this model, the only traffic that can pass between the two network segments must be copied by some agent executing on the firewall system.

These agents, generally called proxies, are responsible for implementing the security policy on the firewall system. Most proxies perform verification of users access rights as a minimum requirement, provide connections to remote resources and perform transfer of data between the two network interfaces, usually by copying the data packets as they are received.

This model is known to provide good security when the proxy services have been implemented properly. Unfortunately, these facilities usually require specially written client programs to access the proxy interfaces, and there is very little standardisation in this area at the current time.

The intention of this paper is to cover what techniques and standards currently exist on the Internet, and what restrictions they place on development of secure interfaces. It is also an intention of this paper to look at what facilities need to be provided to allow efficient client/server applications to be developed within the constraints of a secure environment.

2.0 History

The Internet is a large, amorphous collection of systems connected together using a variety of networking protocols. The most common protocol, the Internet Protocol (IP), was developed to meet the needs of military and educational users in the early 1970's. The design of this protocol was handled primarily as a co-operative venture between its participants and its evolution was based primarily on practice rather than theory. The long term effect of this evolution was the establishment of an extensive network that provides a remarkable level of homogeneity and interoperability.

Over the last 5 years there has been remarkable growth in the number of systems that have been connected to the Internet, so much so that some of the deficiencies of the design have started to create problems. For example, the original address space for IP was 32 bits, which was considered to be a large address space for the type of network that was envisaged at the time. The growth of the network has shown that this address space will shortly become significantly full and there is a real need for the address space to be expanded to 64 bits.

With the growth in size, there has been a parallel growth in access. Systems that had no significant network access are now being integrated at an alarming rate. Many of these systems are commercial and the users of these commercial systems have legitimate concerns about security and accessibility. One of the more recent mechanisms used to provide security has been a facility called a firewall.

2.1 Firewalls and Proxies

A firewall is basically a system configured with two or more network interfaces that denies any direct access for network traffic between those interfaces. The only mechanism available to allow traffic to be passed between the interfaces is a process running on the system that has access to both interfaces and can pass data between them. These processes are called proxies.
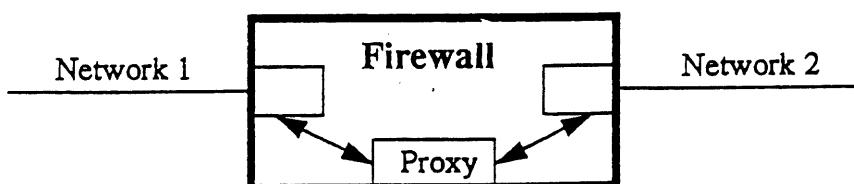
FIGURE 1. A Typical Firewall System

There are many different mechanisms used to create proxy processes. Some proxies configure themselves to emulate well-known services, such as FTP or TELNET, and attach themselves directly to well known TCP ports. When a remote system connects to the port, the proxy creates an equivalent connection to the target system and then forwards all traffic between the two networks.

The major problem with this form of proxy service is its inherent inflexibility. Generally, only a single target system can be accessed using this mechanism, but it does have the advantage that standard client applications can be used to make connections.

A more complex mechanism involves the use of a 'key' application, a security client that arranges for the proxy to prepare for a standard client application to access a service and redirect the connection to a given target host. This facility provides more security as the proxy only listens for a connection when the 'key' application requests access. Once again, the client application does not need to be modified to use the firewall.

2.2 SOCKS

One of the more recent adaptations for use with firewalls is a facility called SOCKS[1]. The concept behind SOCKS is to provide replacement networking interfaces for a number of the standard BSD Socket calls. Instead of using the standard routines, the SOCKS routines redirect the call to a firewall server process. This process, called *sockd*, emulates the routine on the firewall system, effectively making all the network connections on behalf of the SOCKS application.

2.2.1 Clients

To create a new client application, it needs to be *socksified*. This is the exercise of replacing a number of BSD socket calls with their SOCKS replacements. The socket routines that need to be replaced are *bind, connect, accept, listen,* and *getsockname.* Once the application has been re-compiled and linked with the SOCKS support library, all of these routines are redirected to a nominated firewall server. When the application uses any of these routines, the relevant information is passed to the firewall *sockd* server, which performs the requested action on behalf of the client.

The SOCKS library uses a number of environment variables to control the firewall system to be used, with a default value to be used if the environment is not configured. This allows a single binary version to be used on a number of systems without re-compiling.

---

1. This discussion focuses on SOCKS Version 4.0, which was current in December 1993. Later versions of SOCKS provide more functionality than the version described here. The author did not have enough time to investigate the newer versions before this paper was completed.
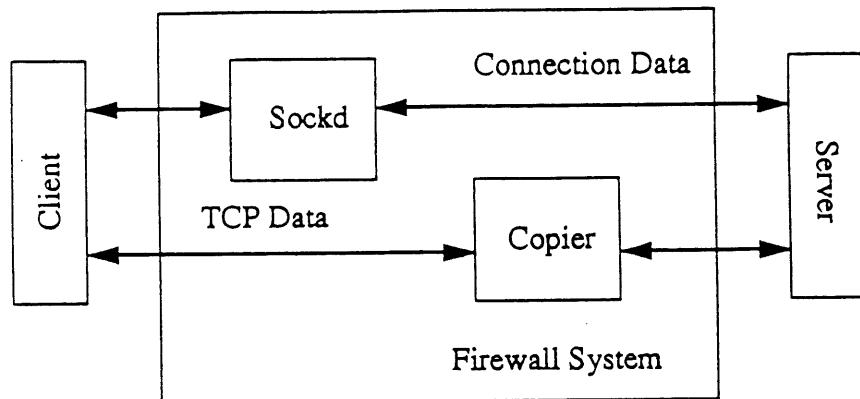
FIGURE 2. A typical SOCKS configuration.

2.2.2 Server

The *sockd* server provides security using a configuration system that allows access using user name, host, network and port permissions. Whenever a client requests any action, the daemon checks for access permission before completing the action. Because of the multitude of configuration options, the configuration file is strictly ordered and configuration options are searched in order until a matching set of criteria are found. This simplifies the configuration because general access policies can be expressed with liberal matching rules, and specific access policies can be placed earlier in the configuration with restrictive matching rules.

It is essential to understand how *sockd* handles most requests. If the request is for the creation of a network connection, *sockd* makes a connection to the remote service and then passes back enough status information for the client to configure its own connection resources. Once this is done, sockd generally converts itself into a packet copier, passing each data packet between the client and its remote server. This need to copy packets is forced by the basic firewall design that restricts the transfer of data between network interfaces.

This behavior has its side effects, since it is essential that there must be some process running on the firewall to perform this data transfer. Another side effect is the difficulty of handling UDP traffic using SOCKS, a difficulty that is avoided entirely by only supporting TCP connections.

2.2.3 Limitations

There are a number of limitations that cannot be avoided with SOCKS[2].

- The SOCKS model is based on BSD Sockets, which is widely available but not universal.

---

2. Some of these limitations have been removed in later versions.

- SOCKS only supports TCP connectivity.

- SOCKS does not fully support any DNS services, making it difficult to rely on name resolution.

- SOCKS only supports a single firewall between the client and its remote server. If more than one firewall exists between the two hosts, SOCKS cannot handle it.

3.0 Issues

Before we can venture further, there are a number of issues that should be addressed to give some perspective on the difficulties to be encountered while implementing a serious proxy facility. Some of these issues appear trivial, but there already exist a number of network facilities that rely heavily on the behavior of network services that must be preserved when using a proxy server.

3.1 Clients and Servers

The usual model used by networking applications is the client/server model. In this model, one system provides an application which performs some service for other applications on other systems.

To provide this access, the server must create a network access point that will accept connections from other systems. Using the BSD socket notation, the server uses *bind* to create an access point, and then uses *listen* and *accept* to handle remote connections. The client application uses *connect* to initiate a connection, which is handled by the *accept* routine in the server.

Despite the simplicity of this model, it is not uncommon to find applications that do not conform. A good example is FTP, which effectively supports both client and server modes at both ends. When FTP starts, it creates a command link between the client and the server. This link is used to handle simple requests and to configure larger requests. When the client wishes to perform a file transfer, it changes mode to become a server, creates an access point and then passes the connection details through the command link to the server, which then performs a client connection to the newly created access point.

This ability to mix modes is very important, if not critical, for the operation of some network services, so any serious attempt to provide proxy support must be capable of handling this type of application.

3.2 Multiple Firewalls

The simplest firewall model consists of a single firewall system between two networks. Most of the proxy services assume that this is the standard model and perform accordingly. As network topologies become more complex, and administrators become more aware of security issues, it will not be uncommon to find multiple firewalls exist between clients and servers.

The impact of this on proxy design is significant. How does a proxy initiate a connection to a service that may have an arbitrary number of firewalls between them? How can security be maintained consistently across each firewall? How can a client be designed to provide access regardless of the firewall topology?
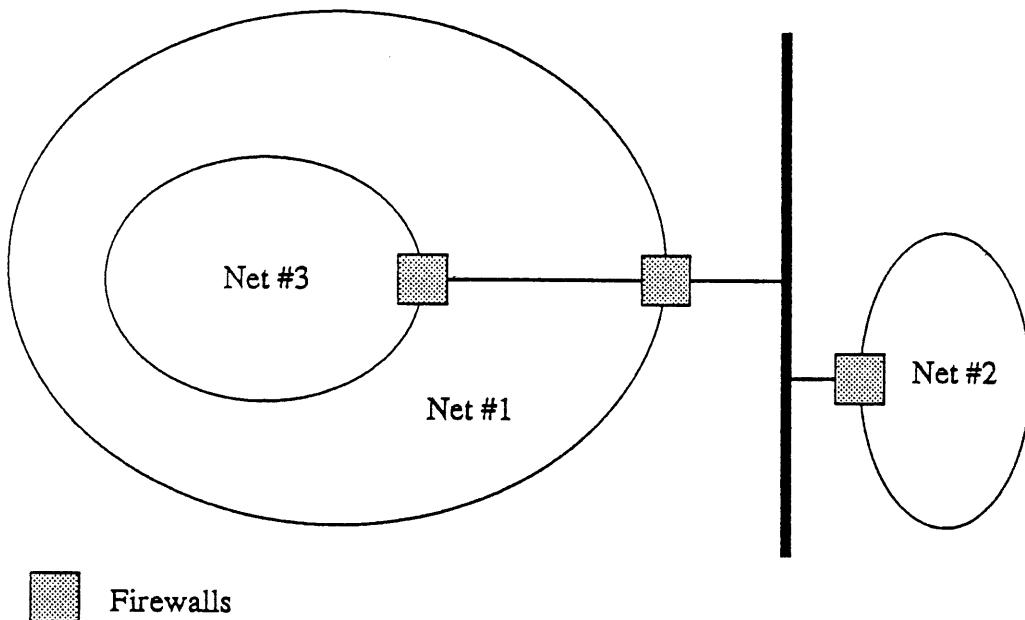
Firewalls

FIGURE 3. Multiple Firewall Topology

Take the topology shown in Figure 3, "Multiple Firewall Topology," above. Here we have three networks. The two major networks, Net #1 and Net #2, are connected together by some backbone network and protect themselves using a firewall. Net #1 contains a secure subnetwork, Net #3, which is protected by its own firewall. If a client in Net #3 wants to access a server in Net #1, a simple proxy server will be capable of providing that access. However, for a client in Net #3 to access a service in Net #2, three firewall systems must cooperate to enable the connection to be created. Current proxy systems cannot handle this topology very well.

The current answer to all the previous questions is basically "It cannot be done.", which is annoying for users and administrators who see a need for better security. It is not uncommon to find the solution is to provide a chain of TELNET sessions from firewall to firewall, with the final system providing the client application. Mind you, if that client is an X application, and you want the output to appear on your display, and the intermediate firewalls won't permit access to your display....

This is not a contrived example, it has occurred often enough in one organisation I am associated with that it required the implementation of specialist software to allow client applications to work properly. I am certain that as networking becomes more available there will be frequent occurrences of this type of firewall topology.

3.3 Name Services

One of the more interesting problems is that of name resolution. Most of the current name resolution relies on DNS, the Distributed Name Service. With the simple firewall model, the usual solution for name resolution is to resolve names on the firewall system, since it is more likely that the firewall will have access to major DNS links.

When multiple firewalls become involved, it is not so obvious how name resolution should be handled. Should names be resolved in the client system, on the first firewall system, at an intermediate firewall system, at the final firewall system, or everywhere in the chain? Remember that the idea of firewalls is to restrict the access to network segments, and it may not be possible for DNS services to penetrate a firewall. A name could be resolvable only within its limited secured network region, which means that the proxy system must be capable of resolving that name securely.

Taking the topology of Figure 3, "Multiple Firewall Topology," again, it is most likely that the two firewalls that are directly connected to the backbone network are the primary candidates for providing DNS services for proxy connections, but it is still necessary for this to be configured into the firewall system between Net #3 and Net #1 somehow, leaving an administration problem.

4.0 The Future

Given the current state of proxy services, and the proliferation of firewall systems, it might be worthwhile to spend some time looking at what future proxy systems may provide to their clients. I should point out that this is primarily a vision of what I think is required to allow proxy services to operate within secured environments, with reasonable administrative facilities, good security, and maximum flexibility in configuration and operation.

4.1 Firewall Intercommunication

To support multiple firewalls, it becomes critical that firewall proxy servers have some mechanism to transfer control to another firewall system when a client requests a service that is provided through that firewall. It should also be possible for firewalls to negotiate what security requirements their service providers need so that access rights can be confirmed at the local firewall, minimising network traffic.

Another major problem with the current generation of proxy servers is the reliance on data copying processes to transfer data between networks. With some careful design, it may be possible to multiplex several proxy sessions over a single virtual circuit, minimising the processing load on some firewall systems.

Another major benefit gained by proper intercommunication facilities is the provision of secure data channels between two firewall systems without the intervention of any intervening firewall system.

4.2 Security

One of the major requirements of most service providers is the adequate identification of users and the provision of security on their systems. The whole concept of a firewall is to deny access to unauthorised users, but any environment that supports multiple firewalls will require two levels of authentication, one for the user, another for the firewall itself.

Unfortunately, the existing networking protocols are not particularly good at providing extended security at the lowest levels. It is certainly possible for a service provider to enquire the connection details from the network interface, but this only identifies the host system that is making the connection, and possibly the identity of the user. The current SOCKS implementations extend this concept by passing user identification information as part of the connection mechanism, and this can enable more complex security mechanisms to be implemented.

All this becomes much more difficult when the user is gaining access from a firewall that may be several systems away and has no direct mechanism to verify user information. In this scenario, intermediate firewalls will need to be identified and trusted for any access attempt. The authentication of a user will only be as good as the worst firewall authentication and it will be necessary for network administrators to vet the *bone fides* of other firewall systems. Providing these facilities, without overloading the network or its administrators, is quite a challenge.

Ultimately, it will become necessary to support a more general user identification and security system, a system similar to KERBEROS or possibly a public-key signature system. Whatever the system, the goal must be to provide good identification and security without the need to inconvenience legitimate users excessively. Of course, really good security requires a greater effort on the part of the user to prove identity, which indicates that any security system requires enough flexibility to allow new methodologies and enhanced security mechanisms to be added as required.

4.3 The Proxy API

The most important requirement for any proxy service in the future is the provision of a standard API for proxy services. One of the major difficulties with interfaces such as SOCKS is their reliance on one particular implementation of a network API, which limits its effectiveness in environments that do not support that API.

With a carefully specified API, client and server applications can be written that can take full control of their environment without being tied to a particular network API, increasing the range of environments that can use these applications.

The Proxy API would also specify the behavior of the firewall servers, their methods of communicating requests, performing authorisations and administration. With a properly specified protocol, the Proxy API could be independent of the underlying network protocols, increasing its flexibility and utility.

With enhanced security requirements, it would also be useful for service providers to be able to negotiate which level of security should be enabled for their service. Currently, if a service provider needs data encryption, the provider must handle the encryption and decryption within their own system which has a side effect of locking the application to a specific encryption standard as well. If the user is willing to accept that traffic between their host and the nearest firewall is secure, then the firewall can be configured to provide the encryption level suitable for that user.

Security policy should be de-coupled from security implementation, since it can be shown that the number of changes in security systems will be notable over the effective lifetime of an application. By de-coupling the implementation, the application need only provide enough hints to the firewall system for the appropriate security level to be established.

5.0 PROXY

I want to finish off this paper describing an implementation of a limited form of multiple firewall proxy server that I call PROXY. It has many limitations and was designed as a vehicle to allow existing SOCKS clients to access services across multiple firewalls.

5.1 Requirement

The basic requirement for PROXY was the need to provide a proxy facility across a two firewall environment. One firewall was designed to provide secure access to the Internet, and the second firewall was designed to protect an internal corporate network. It was a requirement that users on the secure network have similar access to those on the insecure network, preferably without the need to provide user accounts on the secure firewall.

SOCKS had been investigated as a possible solution, but its inability to handle the dual firewall environment was a severe constraint. After a number of discussions, it was agreed that using the SOCKS clients was a definite advantage, so a new firewall server would be designed to accept SOCKS connections and redirect them to the second firewall for handling.

5.2 Concept

The concept that was finally accepted was a general solution to the problem. A new server would be designed that understood the SOCKS protocol, and would provide a new protocol designed for firewall intercommunication.

When a SOCKS request was accepted by a firewall, it would determine where the request must be handled. If the request was local, the firewall emulated a SOCKS server and handled the request. If the request was remote, the request would be forwarded to another firewall for handling. There was no restriction placed on how the other firewall handled the request and it would be possible for the request to be forwarded to another firewall if required.

It was this ability to redirect SOCKS requests that was important, and this worked quite well for most well-behaved client applications. The only major difficulty was FTP, for the reasons explained previously. However, a one line change to the SOCKS support library removed that impediment quickly.

For security, all access rights were checked on the local firewall and access was granted on the basis of 'trusted' intermediates. Each firewall checked the credentials of the firewall requesting service and acted accordingly. This is not a very secure environment, but it is sufficient for networks where administrators trust each other, such as within a single organisation.
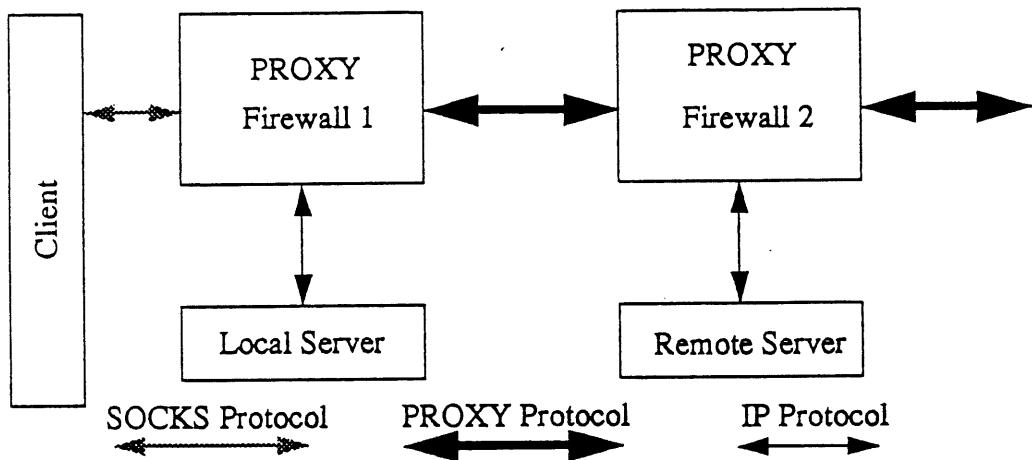
FIGURE 4. A typical PROXY configuration

5.3 Usage

The nice thing about PROXY was the fact that it was transparent to SOCKS clients. The same binary could be used with PROXY and SOCKS without the user knowing any different. Unfortunately, PROXY relies on the same data copying system that SOCKS does, which means that there must be a data copier process running on every intermediate firewall for each connection, a situation that can eat heavily into system resources.

The major difficulty with using SOCKS was its inability to handle name services[3], which was difficult if name resolution had to be handled at the remote firewall rather than at the local firewall. No reasonable workaround was discovered, except possibly allowing DNS to penetrate the secure firewall, which was unacceptable.

The security system for PROXY is not extremely robust and relies heavily on the existing SOCKS model. It is inadequate when requests need to be redirected because there is no mechanism available to allow the remote firewall to verify the user information that was

---

3. A failing that appears to be removed in the current version of SOCKS.

passed to it. It could still perform authorisation checks, but it needed to completely trust the other firewall, a severe failing.

Another major difficulty involves connection management. The current version of PROXY has no facility to properly handle premature termination of a connection unless that connection was handled locally. Once a connection is created through another server, it becomes more difficult to inform intermediate servers that a connection has terminated. The obvious solution for this is a more robust protocol between servers that understands connection details.

6.0 Conclusion

It is obvious that the way in which we are using our computer networks is changing, and that the security of our networks is becoming more important as access becomes more prevalent. Unfortunately, security has its costs, both financially and socially. A cost effective security system could be impossible to use, while a very simple security system could be compromised with ease.

The choices that we make, while not set in concrete, will dictate how our computer networks will evolve over the next few decades, and we need to be aware of the hidden costs involved in providing effective network security. As always, the technical community are addressing these issues and providing innovative solutions that can be of benefit to many. Let us hope that the end result of these endeavors will be practical and effective solutions to our network security problems.

Delving into the UNIX kernel

Adrian Booth

Adrian Booth Computing Consultants
abcc@dialix.oz.au

ABSTRACT

Why would anyone care about what happens "under the hood", inside the UNIX kernel? This paper tries to answer this question, and to briefly describe ways in which the operation of the kernel can be examined.

1. Introduction

It has happened to all of us at some stage: during the busiest prime time period, a UNIX server panics with an incomprehensible error message and reboots. Your boss calls you into his office and asks, "So what happened? Can you prevent it from happening again? The users want an immediate explanation".

Consider two possible answers:

- "I have no idea what happened. I don't even know where to start looking. I've called vendor support; they claim it is because we are using the third-party peripherals I recommended. I hope they're wrong".

- "This was clearly caused by a bug in the operating system (although it took about ten minutes of arguing before vendor support would agree with my diagnosis). It appears to be triggered by a particular set of circumstances which we can work around until a fix arrives - one is on its way."

To maintain our credibility with users and management, we want to be able to give answers like the latter as often as possible. The situation doesn't have to be as extreme as a prime-time panic. When the system suddenly slows to a crawl, immediate investigation is required. Being able to rapidly pinpoint and fix the cause of performance problems - which can dramatically impact productivity and profitability company-wide - is another sure-fire way of establishing and maintaining credibility.

Consider also the case of your company wishing to upgrade one of its servers. The money has been approved and a second CPU is about to be ordered. You recommend that given the server's operating system level and its job mix, it would be more beneficial to upgrade the existing CPU rather than add a second one. Benchmarking with both configurations proves your point, and the money is spent more effectively.

Dealing effectively with these sorts of issues requires you to have some knowledge of UNIX kernel internals. This doesn't mean that you have to regularly tinker with the source code, or write exotic device drivers over breakfast; it simply means that a basic understanding will help you do your job better. Any time you spend investigating the "nuts and bolts" will pay itself off many times over in the future.

2. "Generic" kernel delving

UNIX is uncommon amongst operating systems in that it is quite simple for programs to directly extract information from the kernel.

This is usually done by using an interface to `/dev/kmem`, which contains an image of the kernel virtual memory. The simplest interface is `/dev/kmem` itself, which can be `open()`'d like any regular file. Programs then `lseek()` to the kernel virtual address of the data in which they are interested and `read()` it.

To find the kernel virtual address at which the data is located, the program uses the `nlist()` library call:

```
#include <nlist.h>

struct nlist nl[2];

nl[0].n_name = "_nprocs";
nl[1].n_name = NULL;

nlist("/vmunix", nl);
```

Assuming that the call succeeds, the `n_value` field of `nl[0]` will be the kernel virtual address at which the value of `nprocs` is stored.

As the above example shows, we need to supply `nlist()` with the name of the symbol used to store the data. We can find this out using either kernel source code, or take an educated guess after perusing the output of `nm /vmunix` (which will dump the kernel's symbol table).

3. Berkeley kernel delving

Berkeley UNIX variants include several commands which can be used to examine various aspects of the kernel, including `pstat` and `vmstat`. Everyone's favourite debugger, adb, can also be used. For example, `adb` can be used to find the value of `nprocs`:

```
# adb -k /vmunix /dev/mem
physmem OxNNNN
nprocs/D
nprocs = 128
```

Many UNIX variants that supply an `adb` that can be used to examine the kernel also supply a set of macros that can be used to dump the contents of the more complex kernel structures.

4. System V kernel delving

As well as the *extremely* useful `sar` utilities, System V UNIX implementations provide the `/proc` filesystem, which is a different mechanism aimed at simplifying the programmatic interface into the kernel. Each active process is represented by what appears to be a regular file (with a name corresponding to its process ID) within `/proc`. The virtual memory of a process can be examined simply by `open()`ing the appropriate file and `lseek()`ing to the appropriate address within that file. Process control functions, such as extracting the process structure or renicing a process' priority, are performed by an `ioctl()` on the file descriptor:

```
/*
 * Skeleton renice code: "renice nicedelta pid"
 */

int pd;
int dnice;
char p_path[12] = "/proc/"; /* pathname of /proc file */

strcat(p_path, argv[2]);
pd = open(p_path, O_WRONLY);

dnice = atoi(argv[1]);
ioctl(pd, PIOCNICE, (void *) &dnice);
```

5. Other kernel delving

Other UNIX variants simplify the programmatic interface into the kernel in different fashions. One mechanism is the provision of a library of kvm routines, such as kvm_open and kvm_getu:

```
#include <kvm.h>
#include <fcntl.h>
#include <proc.h>

kvm_t *kernel;
struct proc *process;

kernel = kvm_open(NULL, NULL, NULL, O_RDONLY, argv[0]);

/*
 * Get the process structure for process PID
 */

process = kvm_getproc(kernel, pid);
```

Such an interface hides the details of searching the process table for the process structure of a particular process: opening /dev/kmem, finding out the address of the process table, lseek()ing to that address, reading in the process structure at that address, and following the pointer to the "next" process structure until we find the process in which we have an interest.

6. Information sources

Several books are available that describe UNIX internals; Bach's somewhat dated "The Design of the UNIX operating system" is a common starting point. "The Design and Implementation of the 4.3BSD UNIX operating system" by Leffler, McKusick *et al* offers a readable and more modern alternative. A new book describing the internals of System V Release 4 will be released shortly†. Several operating system textbooks contain a chapter which provides a good introductory overview of UNIX and its design philosophy.

Many vendors produce white papers that describe the theory and implementation of their UNIX variant. These papers are usually available from the vendor, and often also via ftp. USENIX conference proceedings are another excellent source of such papers.

There are several newsgroups that contain discussions of UNIX internals and related issues, such as comp.unix.internals and comp.unix.wizards. It is worth peeking into comp.os.research as well.

The truly enthusiastic may be satisfied with nothing less than the complete source code; there are several UNIX implementations available at no or low cost, such as Linux, 386BSD, BSD/386, FreeBSD, and hopefully soon 4.4BSD-Lite. These versions often have very active newsgroups, user groups, and mailing lists associated with them.

There are a number of public domain utilities that can be used to examine various aspects of the kernel's operation. One of the most useful is top, which displays a summary of the processes consuming the most CPU over time. top's source code is an excellent starting point for learning more about examining the kernel's data structures from within a program.

The author has written a simple program that demonstrates the use of the kvm library. For example, when invoked on a user's login shell (csh):

---

† "The Magic Garden Explained" - Berny Goodheart and James Cox Prentice Hall, 1994. ISBN: 0130981389. 670pp!

```
# pdiag 16570

Process Name: -csh        Pid: 16570        PPid: 16569
Process arglist:
0:        -csh

UID = 2075, Saved UID = 2075, Saved GID = 1000

Current directory = /home/adrian

User Priority=51        Priority=40        Nice=20

Text Size  Data Size  Stack Size  Resident Set
    34          13          8          104

No current signal
No pending signals

Signal mask: 0x2
SIGINT interrupt

Signals being ignored: 0x324004
SIGQUIT quit
SIGTERM software termination signal from kill
SIGTSTP stop signal from tty
SIGTTIN to readers pgrp upon background tty read
SIGTTOU like TTIN for output if (tp->t_local&LTOSTOP)

Signals being caught: 0x1880003
SIGHUP hangup
SIGINT interrupt
SIGCHLD to parent on child stop or exit
SIGXCPU exceeded CPU time limit
SIGXFSZ exceeded file size limit

Process state: SSLEEP (awaiting an event)

Process flags:
SLOAD (in core)
SOMASK (restore old mask after taking signal)
SPAGI (init data space on demand, from inode)
SORPHAN (process is orphaned (can't be ^Z'ed))
SSEL (selecting; wakeup/waiting danger)
SEXECED (this process has completed an exec (posix))
```

This program has proved itself to be especially useful when trying to determine why a process (or, more often, a group of unrelated processes) has stopped running. Source code is available on request to the author.

User Support Mailbox

Dear Janet,

Wouldn't It Be Nice If on exiting from windows the history of the commands obeyed while in the window was remembered. At the moment everything after opening the window is lost.

I have been told that it cannot be done.

<div align="right">Nostalgic.</div>

Dear Nostalgic,

Who told you that?

(I presume that by "window" you mean a Sun `commandtool`, Sun `shelltool`, `xterm`, etc, in which `csh` is running.) While it is running, `csh` saves recent command history in memory. The variable `history` tells it how many lines to save — the default is 1.

When it exits, if the variable `savehist` is set (and non-zero), it saves the last `$savehist` lines in the file `.history` in your home directory. The next time you run `csh`, it loads the lines from `.history` into its in-memory history list.

`history` and `savehist` are usually defined in `.cshrc`.

There is an assumption here that you only ever run one shell at a time. With multiple windows at once, I find the `savehist` mechanism useless, but the in-memory history mechanism is handy.

Anyway, to come close to what you want, you could put something like

```
history >.history.$$
```

or

```
history >histories/csh.$$
```

in your `.logout` file. This will save the last `$history` commands in a more-or-less-uniquely numbered file. `$$` is the shell's notation for its own process ID, which is unique until the system reboots or the process IDs cycle round (after booting, each new process gets the next highest ID that hasn't been used, until it reaches a maximum, probably 32767; after this the system starts re-using low IDs that aren't still in use).

So if your shell is process 12008, and `history` is 40, the last 40 commands will get saved in `.history.12008` (or `histories/csh.12008`).

This will work only if the shell is a login shell. Sun `commandtools` and `shelltools` don't run login shells; `xterms` can be made to with the `-ls` option.

For non-login shells you could define (in `.cshrc`) an alias

```
alias myexit "history >.history.$$; exit"
```

and then remember to use that instead of `exit` or `logout` or Control-D. (You probably have `ignoreeof` set anyway, so that end-of-file, which is usually Control-D, doesn't log you out.) (I would probably call it x, or something similarly short, instead of `myexit`.)

Note that these will save only the `csh` commands.

Also note that you'll only get the last `$history` commands, and you can't make it infinite. Making it big is a bad idea anyway, as it chews up memory.

To save everything, ie, a complete transcript of shell and application commands, use the program `script(1)`. It spawns a new shell and saves a transcript from that shell into a file (the default file is `typescript`, but you can specify one on the commandline). Even this will only work for command-line programs; screen-mode

programs (such as the editor vi (1)) will make a mess of it. You probably wouldn't want to use script routinely, just for the occasional transcript.

For more information about csh history processing, see the manual page csh (1). There's a lot you can do to save typing.

Janet Jackson (Tricks & Traps / User Support Sub-editor) <jackson@cwr.uwa. edu.au>, (09) 380 2408

Articles From the Australian Newspaper

The following articles are as submitted for publication in the AUUG column of the Australian. There might be some minor differences, from what appeared in the Australian as they may have been edited to fit the available space.

What Makes a System Open?

Michael Paddon

A dedicated pollster could find as many distinct answers to this question as people to ask. The variance of opinion is surprising given that "openness", however defined, is dominating the design and marketing of products in our industry. Unless you have very compelling reasons, any computer you buy today will be an open system; whether it be a PC running DOS, a supercomputer running Unix or any of the possibilities in between.

It takes but a moment's thought to realise that any term that can be applied to such a broad spectrum of hardware and software is bound to be misused. I've been warned by salesmen to avoid being "locked in" to open systems. I've also been barraged by endless lists of standards numbers, ported applications and anything else to prove that one system was "more open" than another. On the end user side, I've heard it said more than once that vendors would like to convince us that open system means open wallet!

All these arguments are specious but miss the point when it comes to the crunch of deciding which open system is right for you. The core fact to remember is that a true open system severs you from reliance on your vendor, a situation taken for granted in other markets. For instance, if you buy a popular model of car, you can reasonably expect to be able to source spare parts from a number of manufacturers. You also expect that healthy competition will result in you paying a fair price for those parts. In the traditional computer industry, by way of contrast, vendors used to go to great lengths to ensure that their systems were incompatible, and hence create captive and profitable markets.

Low cost workstations and the humble PC have changed the economics of the information industry, and in passing have changed the world. In particular the PC is the archetypical open system at the hardware level; you can source compatible components, boards, and firmware from many manufacturers. Interestingly enough, the issue that is most talked about is now least important. Differences between microprocessors, whether you have an Intel or a Motorola or a MIPS chip at the heart of your system, are becoming less important as software standards, application binary interfaces (ABI's) and object code translators become common. What remains a critical issue is the bus system used, which dictates what you can plug into your machine and how much it will cost.

Applications programmers prefer to ignore the hardware altogether. After all, why limit the market for your software? And why make your customers unhappy by tying them to a particular machine? At the same time, it costs far too much to maintain multiple versions of a product. Source code standards are the traditional solution to this dilemma. The most influential of these at present are POSIX and X/Open, although many others exist. Not surprisingly, a majority of the widely accepted standards are closely tied to the Unix operating system, which has been breaking ground in the domain of software portability for more than twenty years. In addition, there have emerged two major commercial variants of Unix itself, System V and OSF.

Binary portability is the other side of the open software coin. The ability to move shrink wrapped programs from machine to machine without regard to the hardware is every end user's dream. After all, the real measure of openness to the typical user is: does it run my spreadsheet, wordprocessor and database? ABI's are commonly touted as a silver bullet in this environment, but they tie you down to a given processor family which is not quite the open systems ideal. Today we are seeing more interest in emulators and object translators which can retarget shrink wrapped software to new architectures.

Unix and open systems have been intimately entwined for years. What other operating systems make the grade? Firstly, I'll discount the attempts by vendors to relabel their old proprietary systems as "open". What about DOS? It meets the criteria of multiple vendors, software portability at the binary level and open programming interfaces. Unfortunately, it ties you to a rapidly aging and soon to be obsolete architecture and provides very few system services. I've even heard people claim that DOS isn't an operating system at all, but rather a glorified program loader. In short, DOS is

arguably open and fundamentally flawed, an eloquent testimony to the fact that being open isn't everything.

Any discussion of the shortcomings with DOS leads naturally into speculation about its designated successor. Microsoft would no doubt like to see Windows/NT become the dominant open system of the nineties and beyond. A measure of how seriously the rest of the industry takes this threat is the overnight birth of the COSE consortium, an unprecedented effort by Unix suppliers to provide their products with a set of standards that guarantee portability of software. By the yardstick of advancing the cause of open systems, Windows/NT has already been a resounding success. Whether it will succeed as a product remains to be seen, and depends on whether Microsoft cooperates with other key players in the industry on standards development.

Networking is the last major piece in the open systems puzzle. Despite a lot of interest and verbal support for the OSI stack, in reality most enterprises are adopting a mixture of internet protocols (TCP/IP, and friends) and PC specific networking products.

I have never seen or heard of a system that fulfills all of the above criteria of an open system. You'll have to make the decision on what aspects are important to you, and what paths will leave your future choices as wide as possible. My guess is that most businesses will be running heterogeneous networks of Unix servers, Unix or NT workstations for medium to high computing requirements and DOS machines at the low end. Rather than seeing any one technology dominate, we will see all three types of system increase in sales; a validation of the open systems philosophy.

The role of standards and how they affect the future of the information industry is just one of the subjects of topical interest within AUUG at present. AUUG Inc. is Australia's largest and most active open systems user group, holding regular chapter meetings around the country and publishing a substantial newsletter. Each year AUUG holds a major winter conference and exhibition and minor conferences in all states. For more information on any of our activities, please contact our secretariat on (02) 332-4622 (phone), (02) 332-4066 (fax).

Client-Server Technology Empowers the User and Provides Corporations With a Competitive Advantage

William J. Eisneman

Analysts estimate that today's companies can access only two to five percent of their information resources online, and that about two to five percent of total revenues are spent on their information systems. Taking simple arithmetic to the extreme suggests that to fully automate 100 percent of the enterprise would mean spending 100 percent of revenues - obviously not an alternative. The conclusion implied here is that automation productivity must be improved by at least an order of magnitude. At NCR we believe this order of magnitude gain can and will be achieved as a result of trending away from conventional centralised computing to client-server processing. Client-server offers an alternative to processing that allows companies to develop a cost effective, flexible information system that spans the enterprise. The end result is users are empowered with the ability to turn large volumes of data into information they can use, online, as a competitive advantage.

Everyone has a different definition of client-server. Though at first this seems a problem, it actually makes sense when you realise that if client-server processing is to fulfill its promise and provide systems solutions that empower users throughout the enterprise, it must ultimately be whatever the users needs it to be.

The fundamental aim of client-server processing is to provide end users with a single image of the total information systems service. End users should not have to specify how to find data or where each program should run. Ideally they would see one seamless environment and not have to switch back and forth among systems. Although most client-server implementations do not yet provide the desired single image of the enterprise, complete transparency is not a prerequisite to moves that will ultimately transition the enterprise to a true client-server processing environment.

The key design components of the model are a cooperative computing platform that manages a message passing environment and masks the technical configuration of clients; a client, which addresses specific tasks and provides all services at the local level; and a server which provides services to clients and is accessed via a standard

interface, enforcing integrity within the domain.

The basic assumptions of the conventional mainframe paradigm are expensive hardware, relatively static business and technically naive users. Characteristics of the mainframe implementation are shared resources; disjointed, limited interfaces; huge development backlogs; compromised functionality for processing efficiency; and users bypassing MIS.

The basic assumptions of the PC paradigm are inexpensive hardware, very dynamic business and MIS bottleneck for new applications. PC implementation results in islands of automation, runtime management and local service.

The client-server model has several basic assumptions. Cooperative computing across processors require adherence to standards, flexible interfaces and portability, resulting in greater programmer efficiency. Business requires very dynamic processes and relatively static data structures. And, organisation responsibilities are divided so that users manage the clients, while MIS manages the servers and the cooperative computing platform.

The foundation of the client-server model lies in leveraging cost-effective technologies that enable significant increments of price/performance improvements, paradigm shifts and customer usability. Quite simply, client-server processing realises the potential of a number of key technology breakthroughs: the rise of standards and open systems; the development of robust operating environments (multithreaded, multitasking, systems management, and so on), such as Unix, that can stand up to the demands of online transaction processing (OLTP), networks and truly distributed computing applications; the development of the microprocessor into a totally capable integrated computing engine; the development of networking technologies that can support distributed information architectures; the constant improvement of price/performance at every level of system components; advances in the design of user interfaces; and advances in techniques for application development.

The most important concept of client-server computing is scalability. In marked contrast to the "chunks of power" offered in a mainframe-concentric model, scalable systems increase the life of the computer system by allowing users to preserve their initial investment and add "slices

of power" as their needs grow. The net effect is a bigger return on investment through planned expansion.

A scalable system also provides customers with the quality assurance they need; Their applications are virtually guaranteed to run wherever they are needed on the network. Applications, and the operating environment in which they operate, can move easily and transparently across a range of systems and locations, serving single users at their workstations, or multiple users via servers of various types. The importance of scalability is immediately evident when one considers the role of a true cooperative server in a growing business. A growth demands that a second computer be introduced into the business, managing the existing information and applications they contain becomes a factor in the efficiency of the business.

A tightly coupled multiple architecture contributes the the scalability of computer systems by supporting multiple processors that can be added over time to increase the range of performance. This type of architecture can be deployed to all levels of the enterprise by using the same family of processors and a single, compatible operating system. This brings depth as well as breadth to the client-server environment.

Beyond the power of a tightly couple server, even greater scalability and linearity can be achieved through massively parallel topologies, such as cubes, meshes and loosely couple multiprocessing clusters, to meet the performance demands of today's emerging large commercial enterprise and scientific applications. Whereas a tightly couple architecture can scale efficiently two to ten times that of a single processor, massively parallel architectures can be configured to achieve performance levels of 1000 times or more than a single processor. This range of performance will be required for an emerging class of applications characterised by large multi-terabyte databases, thousands of transactions per second, online decision support processing and multimedia capabilities.

The bottom line is: Client-server processing equates to better performance at lower cost and lower technology risk by providing users with expandable performance. Clients can be added as needed, and servers are very efficient. Personal service is tailored to the user, with local

delivery. Development efforts eliminate the middleman, address even the smallest projects and provide quality through reuse.

These are the types of systems being demanded by users today, as a result of the changing competitive needs within their organisations. Vendors are faced with the challenge of providing cooperative computing environments, effective networking support, interoperable tools, and cost-effective client-server processor components that allow users to build a heterogeneous environment best suited to their needs.

Client-server has evolved rapidly, and with the continuing emergence of new technology, open systems and standards, the pace can increase. Organisations that embrace today's client-server processing options are moving forward consistent with industry acceptance and the accelerating emergence of new client server options. This means users are leveraging client-server systems and solutions developed and endorsed throughout the information technology industry.

This emergence will come not only from the industry infrastructure but from the information technology user community as well. Over the next 12 months, we will see the emergence of even more powerful workstations, communications and network services, and massively parallel enterprise servers. These technologies, along with open systems and standards, will enable the creation of enterprise-wide solutions and transparent data access supporting coexistence/interoperability with existing networks.

William J. Eisenman is vice president of NCR's Large Computer Products Division. He will be speaking later this month at the AUUG 93 Conference and Exhibition at Darling Harbour, Sydney. For more information about the conference please contact the AUUG secretariat on (02) 332-4622 (phone) or (02) 332-4066 (fax).

EUnet: An Infrastructure-Building Alternative

Glenn Kowack

European international networking has emerged at the intersection of three tidal changes in technology and European society: the development of generally available open systems products, the partial relaxation of national

communications monopolies in Europe, and the accelerating development of European institutions.

EUnet was begun in 1982 by Unix enthusiasts participating in the European Unix Users Group. At that time in Europe, there were no generally available international networks, and a few national-scale networks. The independent network users and system administrators looked across the Atlantic at the experience of the USENIX community in the US in forming the UUCP net and USENET. They realised those idea could be adapted to the European environment. However, the US model of "anarchistic connectivity" could not simply be imported wholesale into an environment of multiple languages, different telecommunications regulations, and a wide range of degrees of technical sophistication. Instead they developed an informal two-layer hub-and-spoke network among the countries of Western Europe.

Participating national groups in Western Europe designated single national locations for concentration of national E-mail and other traffic, and forwarded it from those points directly into the EUnet European Network Operations Center in Amsterdam. Normally, a star topology would not be economical. However, European PTT-dominated line pricing policies are historically anything but rational: in general, once a line crosses a national boundary, it does not affect the price to cross additional boundaries.

In its early years, EUnet groups operated informally out of computer science departments and institutes across Europe. In recent years, most EUnet groups have spun out of their institutes and have formed commercial companies and associations. EUnet has evolved from its grass roots origins to become Europes's largest open network services provider, serving 10,000 sites and networks from Iceland to Vladivostok and from the Artic Circle to Northern Africa. EUnet services today include Electronic Mail (RFC 822 and X.400), Internet Protocol services such as remote login and file transfer, Network News, and Archive Access. EUnet has historically taken the lead in making networking available to new communities and regions. EUnet was either the first or one of the first western based networks to operate in Hungary, Bulgaria, and the former Czechoslovakia, Yugoslavia, and Soviet Union regions. EUnet is proud to have been an important source of press reporting during the

failed Soviet coup in 1991.

Most European networking initiatives have aimed to create large, high-bandwidth networks at one go; this approach requires significant government funding for "kick-starting". This has usually consisted of substantial subsides to national academic network service providers. These networks have also been restricted for use by academics, government, and corporations involved in academic research.

EUnet has taken a significantly different approach to infrastructure building. First, EUnet funding has been based upon end-user subscription fees. Second, EUnet has not sought major grants and the subsidies they entail; such grants have the damaging effect of distorting network service prices, which can confuse users and reduce a network's efforts to drive down costs. Third, our goal in infrastructure building has been to build major capacity in many small steps while leading user requirements by always providing one "epsilon" (that is, a very tiny amount) more capacity than is required.

Finally, EUnet has sought to bring networking to any interested parties, networking is not a technology that naturally sees a distinction between the academy, the office, or the home.

We are beginning to see the emergence of ubiquitous connectivity provided by computer networking. This has already begun to create significant changes in the structure of enterprises, and the relationship of people to their work. Many observers predict the emergence of many new types of networking services, provided by an entirely new industry of providers. EUnet provides an example of how this works and the funds it generates feed back to the "provider side" of networking, and make possible the creation of powerful, multinational networking infrastructure without major governmental participation. This success suggests that, as networking continues to grow and new services and types of providers begin to appear, that there will be a significant role to be played by many small private initiatives. This includes working alongside governmental initiatives, and may even serve to replace them. This last point is of real significance to countries with little governmental money to spend on infrastructure building initiatives. They do not need to imitate the western model of major expenditures; they can "get there in small steps".

Glenn Kowack is the president of EUnet. He will be speaking at the AUUG 93 Conference and Exhibition at Darling Harbour, Sydney. For more information about the conference please contact the AUUG secretariat on (02) 332-4622 (phone) or (02) 332-4066 (fax).

Future Trends in Computing:
The Distributed Computing Environment

David Tory

The dominant trend in information processing today is toward "open" computing, loosely defined and multi-vendor, multiple operating system environments. In these open environments, made up of machines ranging from PC's to supercomputers, the greatest impediment to commercial success has been the lack of interoperability. An additional layer of complexity is added when these environments are networked. And yet, the context for computing in the 1990's is the networked, heterogeneous environment.

One model for the evolutionary path to open computing is based on standardized components: a standardized operating system, which would automatically provide standardized application binary interfaces, even standardized hardware. This model ignores the fact that users have made massive investments in hardware from many vendors, which have provided the primary impetus behind the extraordinary rate of competitive innovation from these hardware vendors. More importantly, it does not take into account the real investment users make: applications software, the thousands of programs that the user either cannot or is unwilling to convert to one standardized hardware platform.

Throughout the 1980's vendors and users tried to solve the related issues of application portability and interoperability by attempting to standardize, particularly in the area of operating systems. This tack failed, primarily because standardization around one operating system tended to stifle innovation, with the result that proprietary implementations of that standardized operating system quickly came into being. The promise of open systems is to encourage innovation, while enabling users to preserve their existing investments in hardware and applications.

The successful model for open computing has as its primary goals interoperability and portability.

It is important to note that standards contribute to interoperability and portability. Standards define the interfaces between operating systems and applications. Given that the standards are followed, existing, although more likely new, applications and operating systems can be made portable and interoperable.

The most important notion in this new model of computing is the move away from the idea of a single operating system as the basis for open systems. The real infrastructure of today's corporate computing environment is the network, which connect the diverse computing resources in a corporation. Putting the network to work, and realizing interoperability across networks, will put corporations to work.

A new initiative, the Open Software Foundation's Distributed Computing Environment, contains all the elements, technical and philosophical, to support the industry's growing need for open systems. OSF's DCE, a collection of technologies which enable programmers to write portable applications for distributed, heterogeneous systems, offers freedom and flexibility to programmers and users alike.

DCE provides the tools developers need to write applications which use the network, and its resources, transparently. Application developers don't have to know about the underlying complexities of the network to write portable, distributed applications; users don't have to understand the complexities of the network to use those applications.

Distributed computing recognizes what users have known all along: diversity is not a bad thing. The notion of specialized systems, both hardware and software, makes sense in a large organization, where there may be an assortment of file servers, disks, graphics servers, print servers and compute servers -- all from different manufacturers, all with different operating systems. The key to success -- and the most significant aspect of OSF's distributed computing environment -- is giving the developer and the end user the tools they need to deal with each of these specialized systems in a consistent way.

Complexity is a reality in the corporate world. So is the need to preserve investments, in hardware and software, while enabling corporations to add technologies they need without jeopardizing what they currently have. In

the 1970's and 1980's, the industry talked about the promise of distributed processing. The reality is here, today, in OSF's Distributed Computing Environment.

David Tory is President and CEO of the Open Software Foundation, and was recently in Sydney to speak at the AUUG 93 Conference. For more information about AUUG and its activities please contact our secretariat on (02) 332-4622 (phone) or (02) 332-4066 (fax).

**I have seen a glimpse of the future...
and the future is good!**

Dr Philip McCrea, AUUG President

There were many highlights of AUUG '93, almost too numerous to mention. The Conference keynote speakers were just great. I will never forget the Clifford Stoll event - one hesitates to call it a talk! And I have a confession to make - when the suggestion was made late in 1992 to have Cliff speak at the conference, my first impression was Cliff who?? A copy of The Cuckoo's Egg was hastily thrust into my hand, and after pouring through it in an evening, I was convinced we needed Cliff at AUUG '93!

On the non-cabaret side of things, we had the voices of the Open Systems industry in the form of Geoff Morris of X/Open from the UK, Dave Tory from OSF, and Don McGovern of USL/Novell. It's a pity we couldn't actually get the alliance between X/Open and Novell formally announced at AUUG '93. It was leaked all over the place, but we appear to have been a week or two too early.

Only Ken Thompson could get away with being a plenary speaker at a UNIX conference, and not talk about UNIX! It was great to have Ken there - he was ubiquitous at all the conference activities, and it was a pleasure to have him in our midst.

The Exhibition side of AUUG '93 was pleasing from my perspective - perhaps because I had put a fair amount of effort into that side of things. We had a larger number of exhibitors than before, but in the main, they tended to take smaller booth sizes - no more 12 metre yachts like IBM had at AUUG '91 in Sydney! This is partially a reflection of tougher economic times, and partially a reflection that the interest in open systems is now more focused in the middleware area, rather than at the UNIX operating system

level.

But the highlight of AUUG '93 for me was WABI! I came across it in operation at the IBM boat - yes, IBM had a ferry moored all the time just a stone's throw from the Exhibition Hall, attendees being lured aboard by leaflets distributed by several roller- bladed emissaries! Initially I thought it was the usual 'look how fast our machine is' pitch, but I became intrigued after seeing Windows applications running on an X terminal attached to an RS6000. Yes, this was the much vaunted 'Windows Application Binary Interface', or WABI, which allows Microsoft Windows applications to run, untouched, on an X terminal, regardless of the architecture of of the machine underneath.

My interest (read excitement!) in WABI is for two reasons: firstly, a personal one - I am still suffering withdrawal symptoms from losing my X terminal, an event precipitated by a change of job. Seeing applications which I now use regularly, such as Microsoft Word and Power Point, running blindingly fast on an X terminal, which could also be used for other activities such as reading mail and news etc, had me more than intrigued. I have become disillusioned with MS Windows based mail and news readers, and have now resorted to using two terminals - a Windows based PC for word-processing, and a Motif based X terminal for traditional UNIX activities. As you can imagine, this is quite a hassle - and hence my interest in seeing WABI work.

But my excitement at seeing WABI actually work is fuelled by a second and much more far reaching reason - WABI is truly liberating technology! It will remove the stranglehold that Microsoft currently has on the desktop. Microsoft's position is to dominate all sectors of the computing hierarchy, from the operating system, through communications software, to the applications area. There is nothing wrong with this ideal - indeed capitalist economies are fuelled by it. However, even aficionados of Microsoft would agree that the IT world is not a level playing field at present (to use another currently popular analogy). True, WABI is proprietary technology - it is owned by SunSelect, one of the Sun subsidiaries, and that is some cause for concern, I guess. But this is overshadowed by the liberation that it will provide to users who will now be able to have the best of both worlds on their desktop - being able to run the popular Windows packages, and

having access to the benefits of a real operating system underneath!

The consequences of WABI are far reaching. It has been selected by quite a few hardware manufacturers, and is well on the way to becoming the desktop standard. Imagine the next major Federal Government re-equipment tender that comes us (such as Defence...). It is not too far fetched to imagine UNIX sitting on desktops where, in the eyes of this (totally impartial) observer, it belongs.

At AUUG '93 I saw a glimpse of the future ... and the future looks good!

For more information about AUUG and its activities please contact our secretariat on (02) 332-4622 (phone) or (02) 332-4066 (fax).

Will NT Take the Future By Storm?

Michael Paddon

Microsoft's new operating system, Windows/NT, is generating huge amounts of interest. Earlier this year, a Victorian AUUG chapter meeting was packed out by three times the usual attendees when we had a representative from Microsoft present the newest open system. Whether or not this interest will materialise into sales (as Microsoft hopes) or whether it is simply a function of the marketing blitzkrieg remains to be seen.

My first impression of Windows/NT was that of professional and slick product. The machine boots up into a full windowing environment and the user has to log in, which provides multiple user security, even though the system is not truly multi-user. System administration is performed via a suite of tools integrated into the windowing environment, making it easy for the everyday user to run his machine effectively. It must be pointed out, of course, that these "innovations" have been available on Macintoshes and various Unix machine for years; nevertheless, the improvement over DOS is enormous.

The multi-tasking support work well, and the machine was impressively stable. After an unexpected power failure, NT recovered automatically and gracefully.

Where NT really fails, in my opinion, is in the networking arena. The communications that are supplied amount to little more than file sharing.

While this is a useful feature, it doesn't begin to address the networking requirements of sophisticated users. For instance, suppose I buy one expensive fast CPU and lots of workstations and run Windows/NT on all of them. I can use that expensive resource as a file server, but if I want to run a program on it I have to go and sit in front of its console. There is no mechanism for remote execution of programs, a fact that is intimately related to the fact that NT is not multi-user. This feature will make hardware vendors happy, but it didn't impress anyone at the demonstration.

Another problem with the networking was the surprising omission of a full TCP/IP suite, including programs like FTP and TELNET. Today, most major universities, corporations and governments are building their networks upon a mix of IP and OSI; leaning a heavy emphasis on the former since OSI's promise has failed to materialise. My understanding is that Microsoft is relying on third parties to provide this functionality. Does this mean that such a critical component of a modern system will be subject to wildly varying API's from different third parties?

The POSIX support in NT was the third major weakness I identified. Unlike Unix vendors, Microsoft has chosen to ignore draft standards and implement a very minimal subset of POSIX. Furthermore the POSIX compatibility mode is compartmentalised from the rest of the system, preventing applications from utilising the windowing system. As a professional programmer, I can only describe this arrangement as generally unusable.

Despite the improvement of its predecessor, is Windows/NT worth buying? DOS is cheaper, OS/2 has better Windows 3.1 compatibility, and Unix is far more functional and a better development environment. It seems to me that Microsoft are busy filling an underpopulated market niche.

The Unix vendors' response to the threat of NT (the urgency of which is apparently diminishing) was to form the COSE consortium. I, like many users, held great hopes that this was the dawn of a new cooperation and shared vision for Unix. A lot of this hope was dashed when I attended the COSE panel at the recent AUUG conference. One after one the vendors told us that they were one hundred percent behind COSE and, by the way, our operating system is actually already COSE compliant. One vendor representative

went so far as to claim that the rest of the industry was finally adopting his company's commitment to open systems. It seems the rhetoric changes, but the reality is constant. I wonder what the next consortium is going to be called (does anyone remember ACE)?

Meanwhile, the real work of producing open systems is being done by two standards bodies: POSIX and X/Open. The trademark "Unix" has recently passed into the hands of X/Open, taking it out of control of a vendor for the first time in history. I don't know what the ubiquitous operating system of the year 2003 will look like, but I'll wager that it will be called "Unix" and not "NT".

For more information about AUUG and its activities please contact our secretariat on (02) 332-4622 (phone) or (02) 332-4066 (fax).

A Leopard Rally Can Change Its Spots

Phil McCrea

Well, it probably had to happen at some stage I guess. Twenty three years in the IT industry (assuming one's IT career starts on graduating from Uni or wherever) and I've managed to avoid it all that time. Not that it was there in the beginning - in fact it's been around for only 10 years or so.

What on earth is he on about you say? Well I should bare all, I guess, and tell you. The fact is, the AUUG president has finally become a DOS user!!! Great howls of angst, flog me with a wet lettuce (apologies to Paul Keating), etc. At the ripe old age of 43, I have finally joined the rest of the computer industry and started using DOS. I can now understand some of the things my sister's kids talk about to their friends!

So what prompted all this? Well, it is mainly associated with a change of job. I felt that Softway needed some fresh leadership, so after taking appropriate steps to ensure that the management of the company was in good hands, I went out to find something else to do (actually the plan was to have a long holiday, which was thwarted - but that's a different story). In my new job at ANSAMS (that'll get you all scurrying for the Top 1000 list...), I had the option of using a Unix environment for office type things, or using the DOS PC on my desk. So I thought "What the heck - let's find out what the rest of the industry has to put up with!" I

should point out that my programming days are long past, and my use of a computer is purely for word processing.

Of course I'm not really using DOS - noone really does these days. It's Windows that everyone uses. Even the owners of DOS gave it up as a joke years ago.

What do I like about it? Well my impressions are coloured by the fact that I am a Unix user of the old school - yes, folks, that means vi and troff! Using Microsoft Word in place of vi is the single biggest change. Not having insert mode is a real change - you actually get to use all those keys on the keyboard which have useful words on them - like "home", "end" and "delete"! If I had a dollar for every time I accidentally typed "a" to add a word, or "e" to step to the end of a word, I'd be able to buy a tape backup system by now for my PC!

The truth is, I'm actually starting to like my new computing environment, and was beginning to consider myself a traitor to the cause, till I realised what was really happening - it's not the operating system which will win the day. In the current battle for the desktop (where Microsoft is running hot favourite), the operating system which will run the popular packages will win in the end. For those of us who have an emotional attachment to Unix, let's hope initiatives like WABI, which will enable Windows applications to run on Unix, are successful.

Dr Philip McCrea is president of AUUG. For more information about AUUG and its activities please contact our secretariat on (02) 332-4622 (phone) or (02) 332-4066 (fax).

Will COSE create easier-to-run systems?

Adrian Booth

Some of the quietest achievers in your information technology section are a few very valuable but sadly neglected people. Decisions and actions taken by these people on a day-to-day basis can dramatically impact your company's productivity and profitability. A short list of their tasks includes regular system backups (and occasional restores), user account management, system security management, system performance analysis and tuning, end user support and problem resolution, system and application software support, installation and upgrades, and capacity management and planning.

System administrators - the people we are discussing - have always had these responsibilities. Today, though, the complexity of system administration is mushrooming. Instead of having a single centralised system or a network of homogenous systems to look after, administrators of today have to cope with distributed, heterogenous, client/server environments, containing a frightening mixture of modern and not-so-modern systems that must all be made to interoperate as seamlessly as possible.

Despite the pace of standardisation in the computing industry, system administration standards are still sadly lacking. Even within the "Open Systems" marketplace, almost all of the tasks listed in the first paragraph must be performed differently on each type of system. The effort required to incorporate a different vendor's system into your environment - even if that system confirms to identical "standards" as your existing systems - can be an order of magnitude higher than simply purchasing another system from your existing vendor(s). This costs you real money, and creates an artificial barrier preventing you from simply selecting the ideal platform for each particular task. We find ourselves trading off the ideals of "commodity technology" and "rightsizing" against the known, the tried-and-true, the comfortable, simply because the administration of different systems is so difficult.

Thankfully, this long-neglected area is finally receiving some attention thanks to the COSE alliance. COSE plan to standardise system administration, or at least, "...the six companies will form a working group to facilitate the rationalisation and rapid acceptance of industry specifications in the systems management arena...".

It is easy to be dubious about the end product of this standardisation process. Take the COSE Common Desktop Environment specification for the user interface. According to the original press release, this environment includes features of HP's Visual User Environment (VUE), IBM's Common User Access model and Workplace Shell, OSF's Motif toolkit and Window Manager, SunSoft's OPEN LOOK and DeskSet productivity tools, and USL's UNIX SVR4.2 desktop manager components and scalable systems technologies. This failure to simply

select a technology, but instead to merely conglomerate them, will not make the resulting user interface any simpler to use. It is entirely possible, given the diversity of the system administration tools available across Unix platforms today, that the end result of the system administration standardisation process will be a similar conglomeration of existing tools into an incomprehensible unity.

What will this mean for the Unix marketplace? Several third-party products have appeared on the market, incorporating various degrees of functionality and platform availability. Will the formal standard adopted by COSE be adopted by the marketplace, or will one of the third-party solutions become the industry's de facto standard? Could COSE even adopt one of these third-party products?

On a broader note, it is tempting to hope that the formation of COSE is the beginning of an unstoppable process of Unix standardisation and eventual unification. After all, once COSE standardises the user interface, graphics, multimedia, networking and administration, as they are promising, what else is left? And how will the individual members hope to differentiate themselves in the marketplace?

Unfortunately in this industry of alliances and consortia, we will have to wait and see if COSE can deliver on their promises. COSE's credibility wasn't helped by it forming when it did - just two years earlier, and it wouldn't be seen by many as a panicked reaction to the imminent arrival of Windows NT, and may have even succeeded in crippling the NT juggernaut.

Despite these concerns, it is clear to this observer that Windows NT is the best thing to happen to the Unix marketplace in the past several years - any unification of the Unix marketplace is welcome after years of bitter sniping and infighting. Hopefully though the marketplace will start demanding some leadership and direction on technological issues instead of accepting the "casseroles" being dished out by Unix vendors today.

Adrian Booth is an independent consultant in Perth specialising in high-end Unix system administration. For more information about AUUG and its activities please contact our secretariat on (02) 332-4622 (phone) or (02) 332-4066 (fax).

The Internet - A Modern Social Experiment

Frank Crawford

Most people using Unix systems have heard of the Internet, the mammoth computer network, encircling the globe connecting millions of hosts and carrying terabytes of data a month. Unix systems generally come with software to access the Internet, and, once the financial and administrative hurdles are overcome, they can plug straight in. Those people with more knowledge of the history of the Internet realise that it started out as an experiment that proved to be tremendously successful. Very few people realise that the Internet is still an experiment, but of a vastly different nature. It is no longer simply an experiment in technology, but now also a social experiment.

As with any social system, it is often the effects of scaling simple procedures by many orders of magnitude that produces new and unusual results. Currently, in the PC market, there is considerable interest in workgroups, ie. the ability to share resources and information simply, between a related group of workers. This facility has been available under Unix (and most multiuser operating systems) almost since they were first introduced. Concepts such as data sharing, electronic mail and instantaneous communication were pioneered in Unix and similar systems. Today, most Unix users are wondering why all the fuss about the workgroup concept, as it is expected to be part of the system.

A more interesting aspect of these concepts is how well they work when you are dealing with very large groups, ie. millions of people able to join such groups. This is where the Internet is proving to be such a fertile test bed. There are many instances of such groups growing on the Internet, and much opportunity to look at how such things operate in a large environment.

Software development is one such area. In the past it was often assumed that public domain software was likely to be buggy, with no support and little documentation. These assumptions generally stemmed from the fact that the work was carried out by one or two people, usually part time, and only tested in some limited fashion. While this may still be true in some cases, today most public domain software available on the Internet is often more stable, better tested and documented than that developed

by a commercial organisation. The reason for this is intimately tied with the Internet.

First and foremost is the speed of communication available over the Internet, and the large number of people contactable. For even the smallest program, the author is able to obtain literally thousands of people willing to test, extend and document any program that they find useful to them. Further this can be achieved in a very short time frame, often measured in days for a small code, to months even for fairly large systems. In many cases problems can be identified, fixed and tested in a shorter period than traditional software vendors can even report the problem.

More recently another step has been taken in the utilisation of the Internet with regard to very large projects. There is increasing use of the wealth of skills available. One example is the Gnu C Compiler, available from the Free Software Foundation. This compiler is today the standard for portable C compilers, being available on such systems as MS-DOS, VMS and Unix, across such different hardware platforms as Intel, MIPS, Sparc, Alpha, Motorola and IBM System/370. Further, the development keeps pace with the current standards (ANSI, POSIX, etc), adapts to new releases of operating systems (eg. SunOS 4 to Solaris) and is the basis for other extensions (eg. a C++ compiler, FORTRAN 77 compiler, etc.).

All this is notionally controlled by one person (Richard Stallman), although, in reality, he is really in control of the constant communications of experts scattered throughout the world. Further, there are then hundreds of others who quickly pick up any changes and test them, correct problems with their local system and then return the changes to be folded back in the master source. To give a more concrete example of the speed of these changes, gcc version 2.5 was released at the start of October, by mid-November we are up to version 2.5.4, ie. four subsequent maintenance releases. This was not caused by any major problems, but rather a number of minor corrections performed and released in time for the production of a CD-ROM.

This is not the only example of such major projects, others include Linux (a publicly available Unix-like system), PVM (a system for simulating a parallel machine over a number of workstations) and Perl (a language for file and

pattern manipulation).

One thing all these projects have in common is either a single individual or a small group keeping a tight focus on the objectives of their project, and yet using all the people and facilities available on the Internet to achieve their goal.

For more information about AUUG and its activities please contact our secretariat on (02) 332-4622 (phone) or (02) 332-4066 (fax).

Open Systems? Or Unix?

Michael Paddon

The idea of an "open system" was one of the great changes in the computer industry during the 80's. When that decade opened, giant corporations told their customers what they wanted and extracted hefty prices as tribute to their autocracy. Each type of computer ran its own operating software and required entire departments of specialised staff to tend to its idiosyncrasies.

Unix was a curiosity, confined to the halls of academia and the occasional research institution. I recall being confidently told, on many occasions, that the Unix operating system was not "built for business", and that it was incapable of providing the computing facilities needed in the real world.

Since then, of course, vast changes have been wrought in the industry. All the major computer vendors offer a version of Unix, and the requirement for an open system is non-negotiable in the minds of most buyers. Some of the largest computer manufacturers held out for years against this trend (though if you can find a company that won't claim that they've been behind open systems from year one, I'll be amazed), and as a result suffered enormous financial loss.

The story so far is, therefore, a happy one. The user community has wrested control of their computing systems away from the vendors. This, in turn, has promoted a synergy of software development. Developers can build upon each others efforts, and the results can be cheaply integrated now that the divisive tactics of vendors attempting to segment and isolate the market have been laid to rest.

Make no mistake. Unix has been the biggest

disaster to computer vendors in history. Nowadays, they have to compete on comparable terms and their products are subject to the mechanics of the commodity economics. They are desperate to create the locked and protected markets that they once grew fat on, and it is my observation that the open system concept is being bent to this task.

It seems that any and every system has been tagged "open" in a dishonest attempt to confuse buyers. Closer inspection shows that an open system in this context mean anything from "this system can be networked" to "we implemented a handful of disparate standards that were convenient". Neither of these criteria allows the user to tap the vast resource of commercial and free software that is readily available.

In fact, a dispassionate analysis can only conclude that Unix in its various forms (and modern Unix variants aren't that much different from each other in this day and age) is the best example of an open system extant. As I've said previously, DOS is inarguably an open system, but hardly the framework on which we could build tomorrow's architectures.

Unix, on the other hand, provides a software environment that runs on anything from a 386 PC to a RISC workstation to a Cray supercomputer. Not only is Unix portable, but it is simple, consistent and easy to use. The fear campaign created by marketeers years ago that Unix was "hard to use" has been well and truly debunked, and in the age of full featured graphical interfaces is hardly relevant anyway.

An informal straw poll I ran showed that experienced DOS users can readily switch to Unix given only a few hours of training. The biggest problem is that DOS users want to keep on using the flock of applications familiar to them rather than switch to alternatives that may differ in minor ways. This is a problem that the whole industry is facing, now that DOS is finally being discarded (although it may be ten years before this process is complete). PC-based Unix machines can already run DOS applications directly, while more sophisticated methods such as object translators or interpreters are required on different hardware architectures.

What Unix offers above and beyond the burgeoning flotilla of open systems is a proven evolutionary platform. Operating system portability, world wide interneting, client-server

architectures, file serving and windowing systems are all examples of technology that has grown alongside, and with the help of, Unix. It was Unix that made vendor-specific systems obsolete.

In the same breath, I have to point out that Unix is not the perfect open system. Any competent systems engineer can reel of a list of its faults (although two engineers' lists may not agree). There is an unfortunate tendency in the computer industry to throw away what isn't perfect and start again, but this is an approach which cannot work for an operating system which requires man-decades to develop. Despite not being perfect, I believe Unix has proved its worth as a platform for cooperative evolution of our software systems. It is also the most cost effective, which makes life a little easier for those of us who must justify our decisions.

In the long run, we have to recognise that perfection is always just out of reach. Instead we need to make sound business and engineering decisions and be prepared to change with time. This is what Unix has always been about. Can your open system claim this distinction?

For more information about AUUG and its activities please contact our secretariat on (02) 332-4622 (phone) or (02) 332-4066 (fax).

AUUG MANAGEMENT COMMITTEE
SUMMARY OF MINUTES OF MEETING 22nd April 1994

Location: AUUG Business Office, North Sydney.

Present: Phil McCrea, Glenn Huxtable, Frank Crawford, Michael Paddon, Rick Stevenson, Stephen Boucher, Peter Wishart.

Apologies: Chris Maltby, Greg Birnie.

Guests: Catrina Dwyer.

1. President's Report

We have now formally employed Catrina. It is a pleasure to have her on board and working as part of the team. Signed up ACMS as conference organiser for AUUG94.

Represented AUUG in launching Bernie Goodheart's book. Invited to the CADEX forum to represent AUUG.

2. Secretary's Report

Non renewals from the membership period expiring on 1st Jan 1994 now stand at 28 institutions and 78 members. This is down on non renewals from previous years. Committee decided that list of non renewing members should be distributed and contact made to determine why they did not renew.

Proposed to develop a model of membership processing in the Secretariat and develop automated procedures which reduce to a minimum manual steps in processing. Decided to seek help from Sydney chapter.

There was discussion about the membership card. It was noted that the logo and byline used on the current membership card were old and no longer appropriate. It was also noted that the current membership card was considerably larger than a credit card, making it difficult to fit into most wallets etc. It was decided that the membership card should be redone, trying to make it smaller and incorporating the correct logo.

3. Treasurer's Report

Although several sponsors (IBM, Sun) have been signed up, we have not yet received any money from them.

The Treasurer tabled financial estimate documents. He estimated that we would break even over the 1993/94 financial year. There are no more major expenses expected for the rest of the financial year.

Monies had be received from summer conferences in Vic, SA, Tas, Qld, but not NSW, ACT, NT. Chapters will need to supply financial statements for the end of year. Chapter profit/loss affects AUUG profit/loss.

4. Business Carried Over

4.1. AUUG96 Tender Specifications

Final document should be sent to conference organisers and tourism companies.

AUUG itself could bid for the exhibition, so could chapters. These bids would be considered against any others. We need to decide what is happening with AUUG96 before Sept 1994 so we can advice ACMS at AUUG94. The final tender document should go out at the beginning of May.

5. Conference Committee Update

This year do a single page brochure of 50,000 copies and get onsert into ACS, IEAust etc magazines. Offer their members the membership rates. Make it a two stage approach, with one pager, then a bigger brochure (aim for 8 or so pages).

Jagoda Crawford has been appointed editor of the AUUG94 proceedings. It had been agreed to use A5 size, but there was now concern about not being able to read documents of that size. Decided to go back to A4 size. If slides are to go into the proceedings then they should be 4 per page. Editor to have discretion on whether slides should go into proceedings.

A budget overview from ACMS was circulated. Need to look at why sponsorship income projection is down on AUUG93.

6. AUUG95 Theme

Need to decide AUUG95 theme soon so it can be used at AUUG94. Needs to include "Internet". It would be good to have some structured demo at the exhibition which showed connectivity. The theme needs to attract both technical and management.

The committee conducted a brainstorming session and resolved to use the theme "The Internet Means Business".

7. Other AUUG Programs

The Kirk Tutorial was a great success but hard to organise in conjunction with the summer conferences. We should aim to hold similar events in the future. June is the best time, being northern hemisphere summer holidays. AUUG should decided to make this a regular event. We should aim to make money from it. Also consider some events after the winter conference to make use of conferences invited speakers.

Discussion on organising Gene Spaffords visit. He should buy his ticket overseas since it is cheaper. We should organise his ticket and recover costs from the other people involved.

Linus Torvalds will visit LUGs (Linux User Groups) around Aust. We should push this relationship with LUGs as a way to raise membership. The LUGs should be encouraged to contribute to the costs and chapters should be offered the opportunity to run local Linus visits.

8. Membership Representatives/Benefits

We should have membership card for each individual in AUUG, this includes institutional representatives. Those with voting rights should be identified on the card. There should be distinct numbers.

There should be a mechanism to add institutional representatives. We should aim to cover our costs for their membership. The institutional representatives should be classified as:

(a) administrative - where we send the bill

(b) primary - voting/nominating rights, a copy of AUUGN, discounts, attend meetings

(c) secondary - a copy of AUUGN, discounts, attend meetings

(d) others - discounts, attend meetings

Motion: The cost of additional institutional representatives shall be $70 per representative. These representatives shall not receive a copy of AUUGN. Moved: MP/GH. CARRIED.

9. Other Business

9.1. Recognition

Motion: That AUUG congratulate Greg Rose on his appointment to the USENIX board. Moved: FC/GH. CARRIED.

9.2. Australian Articles

We have submitted 25 articles. None has been knocked back. Our clipping service is covering the articles. There has been only one controversial article. We need to have more people involved, we are currently relying on too few people. All invited speakers to AUUG94 will provide some material. Should ask Jeremy Horey to do something official at AUUG94 (like chair a session).

The committee agreed that MP was doing a good job in organising the Australian articles.

9.3. Membership Drive

Discussion on need for and ways to run a membership drive. Need to generate some money to fund the drive. There was some discussion on why we would want more members.

Chapters should be encouraged to distribute AUUG collateral and to give membership info to those non-members who attend meetings.

We should use the conference publicity to push membership. Possibly have a SCO Developers Conferences in conjunction with AUUG94. We could use the exhibition better by sending mail to all people who attended. Corporate sponsors may be prepared to onsert AUUG material into their publications.

10. Next Meeting

The next meeting will be on Fri 1st July 1994 at a location to be determined. Meeting date and location will be confirmed when the new committee is known.

Peter Wishart AUUG Inc - Secretary

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.