ISSN 1035-7521

AUUGN

AUUG Inc. Newsletter

Volume 14, Number 6

December 1993

Registered by Australia Post, Publication Number NBG6524

The AUUG Incorporated Newsletter

Volume 14 Number 6

December 1993

CONTENTS

AUUG General Information 3

Editorial 5

AUUG Institutional Members 7

AUUG President's Page 10

Minutes of the 1993 AGM 11

AUUG articles for the Australian 13

Financial Statement 14

How did AUUG spend the monies at AUUG'93? *Liz Fraumann* 22

Opportunity knocks softly sometimes.. 25

Conference Announcements

Berkeley 4.3 - 4.4BSD Workshops 26

AUUG Summer Conference Series 30

Fifth Annual Canberra Conference and Workshops 32

Annual AUUG-NSW Summer Conference 33

Northern Territory Chapter 35

AUUG Queensland Chapter 38

South Australian Summer Conference 41

Victorian Summer Technical Conference 43

Perth Summer Technical Conference 46

Tasmanian Summer Conference 48

UPDATE - USENIX Conference 49

UniForum NZ'94 - Conference Countdown 52

AUUG Local Chapters

AUUG Inc. - Victorian Chapter 54

Update on AUUG Inc. - Victorian Chapter Activities *Stephen Prince* 55

From the Western Front *Janet Jackson* 56

WAUG - Meeting Reviews 57

Open System Publications 59

ACSnet Survey 60

Book Reviews 63

Prentice Hall Book Order Form 67
WoodsLane Information 68
!AUUGN - from AUUGN Volume 1, Number 1
Letter from Phil McCrea 69
Technical Reports
LINUX - An Open System Opened *Peter Chubb* 70
Distributed Object Management *Harald Reiss* 78
From login: - Volume 18, Number 5
After the Fact 88
4.4BSD Release 91
System Administration Tools.. 94
Internet Hypertext is Here 95
An Update on UNIX-Related Standards Activities 97
Unix Tricks & Traps *Janet Jackson* 107
Summary of Management Committee Minutes - 5th November 1993 108
AUUG Membership Categories 110
AUUG Forms 111
Calendar of Events 115
ADVERTISEMENTS
C&T Computing 6
UniForum NZ'94 9
Softway 24
Prentice Hall Book Order Form 64
WoodsLane Pty Ltd 65
X/Open 116

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

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

AUUG General Information

Memberships and Subscriptions

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

Membership and General Correspondence

All correspondence for the AUUG should be addressed to:-

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

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

AUUG Business Manager

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

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

AUUG Executive

President

Phil McCrea
pmc@atom.ansto.gov.au
ANSAMS
Private Mail Bag 1
Menai NSW 2234

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

Greg Birnie
greg@lna.oz.au
Leeds & Northrup Australia P/L
42 McKechnie Dr.
Brisbane Tech. Park
Eight Mile Plans QLD 4113

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

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

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

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.

AUUG Newsletter

Editorial

Welcome to AUUGN Volume 14 Number 6, the last one for another year. I hope you all have a Good Christmas and a Happy New Year.

In this issue we have the final wrap-up of AUUG93, including the AGM and the details of where your money went. Unfortunately, AUUG these days has activities all the time, no sooner have we finished the winter conference, then we have the announcement of the summer conferences. This year, as an added attraction, we have obtained the services of Kirk McKusick to run tutorials in conjunction with many of these conferences. The Summer Conferences are a major activity for the technically minded members, so you should get behind it by submitting papers and other tutorial topics for presentation.

Also included in this issue are the usual topics, such as chapter activities, book reviews, and papers (from previous summer conferences). One item to note in particular, is the Uniforum NZ conference, to which AUUG is trying organise a contingent.

Jagoda Crawford

AUUGN Correspondence

All correspondence regarding the AUUGN should be addressed to:-

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

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

AUUGN Book Reviews

The AUUGN book review editor is 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

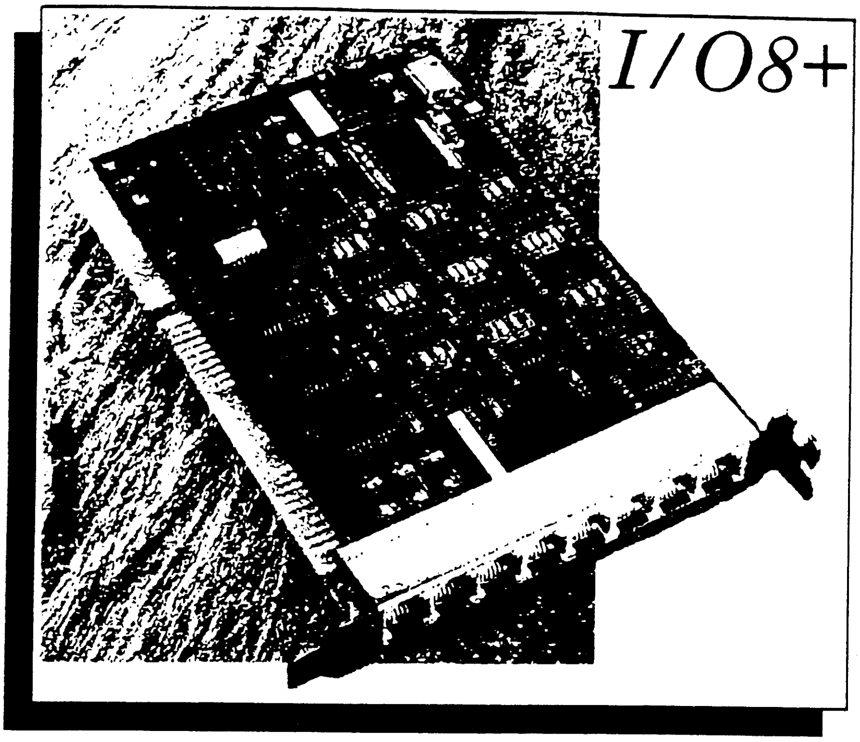
The Newsletter is published approximately every two months. The deadlines for contributions for the next issues of AUUGN are:

Volume 15 No 1	Friday 28th January
Volume 15 No 2	Friday 25th March
Volume 15 No 3	Friday 27th May
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.


I/O8+

AUUG Institutional Members as at 06/12/1993

A. Goninan & Co. Limited
A.N.U.
AAII
Aberfoyle Resource Limited
Actrol Parts
Adept Software
Advanced Software Engineering
Alcatel Australia
Amalgamated Television Services
Amdahl Pacific Services
Andersen Consulting
ANI Manufacturing Group
Animal Logic Research Pty. Ltd.
ANSTO
Anti-Cancer Council of Victoria
Atlas Computer Systems
Attorney Generals' Dept.
Attorney-General's Dept.
AUSOM Inc.
Auspex Systems Australia
Australian Archives
Australian Bureau of Statistics
Australian Computing & Communications Institute
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 National Parks & Wildlife Service
Australian Software Innovations
Australian Submarine Corporation
Australian Taxation Office
Australian Technology Resources (ACT) Pty. Ltd.
Australian Technology Resources (WA) Pty. Ltd.
Australian Tourist Commission
Australian Wool Corporation
AWA Defence Industries
B & D Australia
Bain & Company
BHA Computer Pty. Limited
BHP Information Technology
BHP Minerals Exploration Department
BHP Petroleum
BHP Research - Melbourne Laboratories
BHP Research - Newcastle Laboratories
Bond University
Burdett, Buckeridge & Young Ltd.
Bureau of Meteorology
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
Classified Computers Pty. Ltd.
Clegg Driscoll Consultants Pty. Ltd.
Co-Cam Computer Group
Coal & Allied Operations
Cognos Pty. Ltd.
Colonial Mutual
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.
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 Technologies
Deakin University
Defence Housing Authority
Defence Service Homes
Dept. of Agricultural & Rural Affairs
Dept. of Business & Employment
Dept. of Defence
Dept. of Education, QLD
Dept. of Family Services & Aboriginal &
Islander Affairs
Dept. of Industrial Relations, Employment,
Training & Further Education
Dept. of Justice
Dept. of Planning & Development
Dept. of State Services
Dept. of the Premier & Cabinet
Dept. of the Treasury
Dept. of Transport
DEVETIR
Digital Equipment Corp. (Australia) Pty. Ltd.
DSTO, Lab 73
EASAMS (Australia) Limited
Electronic Financial Services Limited
Engineering Computer Services Pty. Ltd.
Equinet Pty. Ltd.
Equity Systems Pty. Limited
Ericsson Australia Pty. Ltd.
ESRI Australia Pty. Ltd.
FGH Decision Support Systems Pty. Ltd.
Financial Network Services
Fire Fighting Enterprises

AUUG Institutional Members as at 06/12/1993

First State Computing
Flinders University
Fremantle Port Authority
Fujitsu Australia Ltd.
G.James Australia Pty. Ltd.
GEC Alsthom Information Technology
GEC Marconi Systems Ltd.
Geelong & District Water Board
Genasys II Pty. Ltd.
General Automation Pty. Ltd.
GIO Australia
Golden Circle Australia
Great Barrier Reef Marine Park Authority
Gribbles Pathology
Gunnedah Abattoir
Haltek Pty. Ltd.
Hamersley Iron
Hermes Precisa Australia Pty. Ltd.
Honeywell Ltd.
Honeywell Ltd.
Hong Kong Jockey Club Systems (Australia) Pty. Ltd.
I.B.A.
I.P.S Radio & Space Services
IBM Australia Ltd.
Iconix Pty. Ltd.
Ideas International Pty. Ltd.
Information Technology Consultants
Informed Technology
Insession Pty. Ltd.
Insurance & Superannuation Commission
Integration Design Pty. Ltd.
International Imaging Systems
Internode Systems Pty. Ltd.
Ipec Management Services
James Cook University of North Queensland
JTEC Pty. Ltd.
Knowledge Engineering 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
Matcom Technologies
Mayne Nickless Courier Systems
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
MPA International Pty. Ltd.
Multibase Pty. Ltd.
Multiline BBS
Multiuser Solutions Pty. Ltd.
National Library of Australia
NCR Australia
NEC Australia Pty. Ltd.
Northern Territory Library Service
Northern Territory University
NSW Agriculture
Object Oriented Pty. Ltd.
Object Technology International Pty. Ltd.
Objectif Pty. Ltd.
Ochre Development
Office of Fair Trading
Office of National Assessments
Office of the Director of Public Prosecutions
Olivetti Australia Pty. Ltd.
Open Software Associates Ltd.
OPSM
OSIX Pty. Ltd.
OzWare Developments Pty. Ltd.
Pacific Semiconductor Pty. Ltd.
Pacific Star Communications
Paxus
Petrosys Pty. Ltd.
Philips PTS
Platform Technologies Pty. Ltd.
Port of Melbourne Authority
Powerhouse Museum
Process Software Solutions Pty. Ltd.
Prospect Electricity
pTizan Computer Services Pty. Ltd.
Public Works Department, Information Services
Pulse Club Computers Pty. Ltd.
Pyramid Technology Corporation Pty. Ltd.
Qantek
Qantek
QLD Electricity Commission
Quadratron Pacific Pty. Ltd.
Quality Bakers Pty. Ltd.
Quality By Design Pty. Ltd.
Redland Shire Council
Rehabilitation Tasmania
Release4
Renison Golfields Consolidated Ltd.
Repatriation General Hospital, Hollywood
RGC Minerals Sands
Rinbina Pty. Ltd.
Royal Melbourne Institute of Technology
Scitec Communication Systems
Sculptor 4GL+SQL
SEQEB Business Systems
Shire of Eltham
Siemens Nixdorf Information Systems Pty. Ltd.
Smorgon ARC
Snowy Mountains Authority
Software Development International Pty. Ltd.

AUUG Institutional Members as at 06/12/1993

Softway Pty. Ltd.
Sony Technology Centre of Australia
South Australian Lands Dept.
St. Gregory's Armenian School
St. Vincent's Private Hospital
Stallion Technologies Pty. Ltd.
Standards Australia
State Bank of NSW
State Super (SSIMC)
Steelmark Eagle & Globe
Sterling Software
Storage Technology of Australia
Strategic Information Technologies Pty. Ltd.
Sunburst Regency Foods
Swinburne Institute of Technology
Sydney Electricity
Sydney Ports Authority
System Builder Development Pty. Ltd.
Systems Development Telecom Australia
TAB of Queensland
TAFE NSW, Information Systems Division
Tandem Computers
Tattersall Sweep Consultation
Technical Software Services
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 Preston Group
The Roads & Traffic Authority
The Southport School
The University of Western Australia
Thomas Cook Ltd.
TNT Australia Information Technology
Toshiba International Corporation Pty. Ltd.
Tower Software Engineering Pty. Ltd.
Tower Technology Pty. Ltd.
Tradelink Plumbing Supplies Centres
Turbosoft Pty. Ltd.
TUSC Computer Systems
UCCQ
Unidata Australia
Uninet Consulting Pty. Ltd.
Unisys Australia Ltd.
UNIVEL
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
UNIX System Laboratories
Unixpac 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

UniForum NZ'94

The Open Advantage

11th Annual Conference

18-21 May 1994

Quality Hotel, Rotorua

Combine a little work with pleasure! Attend UniForum NZ'94 and see the sights of one of New Zealand's top tourist destinations at the same time.

The exchange rate works in your favour, and AUUG members may attend the conference at the NZ member rate.

Conference Registration	$NZ	$AU Approximate *
before 20 April	$395	$325
after 20 April	$495	$410

(includes attendance at all conference sessions 19-21 May, exhibition, and all meals except breakfast)

Tutorials (18 May)	$150	$125 each
Hotel Accommodation	$100	$85 per night

* Conversion to AU$ will depend on exchange rate fluctuations.

Registration forms will be available mid-March 1994. Contact Julie Jones, UniForum NZ, P.O. Box 27-149, Auckland, New Zealand, Ph 64-25-958-245 or fax 64-9-629-2015.

AUUG President's Page

Another year gone, and a new one has begun - the year of the Family to be exact! I'm sure the years are nowhere near as long now as they used to be!

1993 in retrospect

1993 was a good one for AUUG. The highlight was the formation of several State Chapters - only Tasmania is yet to officially form a Chapter, although there is an active UNIX group there as part of the ACS. During the coming year we will be fostering the growth of these chapters, and in so doing, encourage AUUG members to start to think in terms of their local State chapter first, rather than considering AUUG purely as a national association.

Another highlight was AUUG '93, our annual conference in September. Those of you who attended (and by far the majority of AUUG members did) would probably agree that it was the best yet. Certainly the press gave us a good rap, and because of the excellent exposure the annual AUUG conferences are becoming a major feature of the Australian IT industry. If you ignore the PC and MacIntosh events, the AUUG exhibition is the largest computer show of the year. This is of course due to the popularity of Open Systems, which is the flavour of the decade.

In this issue, Liz Fraumann reports on the financial side of the Conference, and I'm pleased to say that the entire event - ie Conference and Exhibition - was profitable. But only because of the contribution from the profits of the Exhibition. AUUG as an organization does not run the Exhibition - that is carried out by ACMS, who are professional Exhibition organizers. AUUG splits the Exhibition profits with ACMS. This is the only way we can keep the Conference costs as low as they are: next time you receive some junk mail in your In Tray for a two or three day Conference, take a note of the cost - it is generally in the vicinity of $250 - $300 per day.

A Change in the Management Committee

Chris Schoettle, who was elected to the Management Committee only in this past year, has relocated to the US to continue working with USL (whoops - Novell I mean) in New Jersey. We will miss Chris, and I would like to thank him for the contribution he made to AUUG in the short time he was on the Committee. However Chris will only be as far away as email, and we anticipate he will continue his involvement with us from afar.

To replace Chris, we have invited Rick Stevenson to join the Management Committee. Rick brings with him a wealth of UNIX experience. Until recently he worked for Pyramid on the Gold Coast on kernel development, but when that operation folded up, he joined Stallion as Marketing Manager. We welcome Rick to the Committee.

Phil McCrea

AUUG
Minutes of the 1993 Annual General Meeting
Auditorium, Sydney Convention and Exhibition Centre
Wednesday 29th September 1993

Officer Bearers Present: Phil McCrea (President), Glenn Huxtable (Vice-President), Peter Wishart (Secretary), Frank Crawford (Treasurer).

Apologies: Robert Elz

Seventy seven (77) AUUG members signed the AGM attendance record as being present at the meeting.

Meeting opened at 5:50 pm.

1. Minutes of Last Meeting

The President (Phil McCrea) advised that the minutes had been published in AUUGN Vol 13 No. 5 Oct 1992.

Motion: That the minutes of the 1992 AGM as published in AUUGN Vol 13 No. 5 (October 1992) be taken as read and accepted. Moved: Peter Chubb/Chris Keane CARRIED

2. Returning Officers Report

Michael Tuke, outgoing returning officer advised the meeting that a full report on the elections had been published in AUUGN Vol 14, no 4 (August 1993).

The results of the election were:
President: Phil McCrea
Vice-President: Glenn Huxtable
Secretary: Peter Wishart
Treasurer: Frank Crawford
Committee Members: Chris Maltby, Michael Paddon, Greg Birnie, and 2 positions vacant.
Returning Officer: vacant.
Assistant Returning Officer: vacant

Motion: That the returning officers report be accepted. Moved: Lawrie Brown/ Adrian Booth. CARRIED

The committee had co-opted members to fill the vacant positions. It was required by the constitution that the general meeting validate these appointments. The following motion was placed on in the published agenda.

"Motion: That this meeting approves of the appointments of Stephen Boucher and Chris Schoettle to the vacant general committee positions and Michael Tuke to the vacant Returning Officer position. Moved: P. Wishart/ F. Crawford"

Subsequent to the publication of the AGM agenda the last vacant position of Assistant Returning Officer had been filled by the committee. It was moved by Peter Wishart and seconded by Chris Maltby that the motion be amended by adding "and Gregory Bond to the vacant Assistant Returning Officer position." The amendment was agreed to and the motion put.

Motion: That this meeting approves of the appointments of Stephen Boucher and Chris Schoettle to the vacant general committee positions and Michael Tuke to the vacant Returning Officer position and Gregory Bond to the vacant Assistant Returning Officer position. Moved: Peter Wishart/ Frank Crawford CARRIED

3. Presidents Report

The President, Phil McCrea, reported that there had been much activity with chapters over the past year. We now had chapters in each mainland state and territory and all were running a series of activities in their region.

There had been a change in the arrangements with ACMS for the conference that had resulted in more work for AUUG but a greater share of the profits. The treasurer would provide more details in his report.

Liz Fraumann, as Business Manager, had done an excellent job and had been invited to extend her contract for services to AUUG. ACMS had performed well in the role as the Secretariat and would continue in that role. Lachie Hill has been appointed as AUUG's PR consultant.

The next major activity would be a membership drive.

Motion: That the President's report be accepted. Moved: John Lions/ Greg Rose CARRIED.

4. Secretary's Report

Peter Wishart, the Secretary reported a 1992/93 Secretary's report had been published in AUUGN Vol 14, No. 4 (August 1993). He summarised that report.

Motion: That the Secretary's report be accepted. Moved: Janet Jackson/Ken McDonald CARRIED.

There was discussion about the location of the conference. It was suggested that Brisbane or Perth be considered as a future venue. It was noted during discussions on vendor reluctance at such locations, that the AUUG was a users group and should not be driven by the vendors. However it was also noted that international speakers may be jeopardised by lack of funds if the vendors were not involved. The conference for 1994 was already committed to Melbourne.

5. Treasurers Report

The Treasurer (Frank Crawford) reported that there was around a $34K loss for last year. This was due to the most part to the lack of a profit from AUUG92 in Melbourne. It had budgeted for a $65K profit and had actually run at a $25K loss.

The arrangements for the exhibition profit had been a 70/30 split favouring the exhibition organiser (ACMS). That had been changed this year to a 50/50 split. However AUUG must now also contribute to expenses for the exhibition.

It was estimated that the 1994 exhibition would contribute $100K to AUUG. The budget for the conference had expected $50K from the exhibition. So it appeared that we would make around $50 overall on the conference and exhibition.

The budget for this year had been based on around $20K from the conference so our position this year should be much better.

Full details of the finances would be published in AUUGN.

Motion: That the Treasurer's report be accepted. Moved: Greg Rose/ Peter Grey. CARRIED.

6. Other Business

The committee reported that it had decided to provide extra support for one summer conference per year. This year that conference would be in Brisbane. Glenn Huxtable reported that an international speaker, Kirk McKusik, will be providing a tutorial and paper at the summer conferences. The logistics and schedule are still being sorted out.

The committee had appointed Adrian Booth as AUUG Historian to capture and document the history of AUUG. Anyone with historical information and items on AUUG is encouraged to contact Adrian.

Andrew McCrae suggested that the relationship AUUG had established with ACS (Aust. Computer Society) for the conference had not worked very well. We had offered discounts to ACS members for the conference, and advertised the confere.ıce jointly but had only got around 50 ACS members. Perhaps the ACS relationship would be better if done through the chapters.

Ken McDonald stated that CMGA (Computer Measurement Group of Australia ?) were interested in a relationship with AUUG. They had been predominantly a mainframe based organisation but were moving to open systems. They had also approached SAGE-AU about a relationship. Ken stated that as he was their first contact he was happy to try to further the relationship.

Motion: that Ken McDonald be appointed point of contact with CMGA. Moved: Michael Paddon/ Glenn Huxtable CARRIED.

Adrian Booth was coordinating a bulk 4.4BSD T-Shirt purchase on behalf of AUUG.

7. Next Meeting

The next meeting will be held in conjunction with AUUG94 at the World Congress Centre in Melbourne, September 1994. The exact date and time will be advised to members.

The meeting closed at 6:50 pm.

AUUG articles for the Australian

As you may be aware, AUUG submits an article to the Australian newspaper on a weekly basis. We'd like to encourage the involvement of our chapters and our members in this activity. The Australian is an important forum for making our views heard in the wider community.

Publishing an article in the Australian is of great value to our individual contributors as well since it is a respected and high-profile national newspaper. We can't promise fame and fortune, but it is possible. :-)

I am currently responsible for managing the submission of pieces to the newspaper and I'd like to put together a schedule so that I can guarantee a steady and trouble free flow of material.

This is, therefore, a call for people who are ready and willing to write an article in the next three or so months. Each article should be on a topic relevant to computing or open systems in general and should be 600-800 words in length. Articles should be submitted to me in some electronic form (email preferred).

If you are interested, please nominate a date from the list below that you are willing to commit to. If you can't think of a topic, but are willing to write, nominate a date anyway and I'll supply you with a list of ideas.

Available article dates (all Wednesdays):

6,13,20,27 January
2,9,16,23 February

Of course, if you have any questions, etc. please don't hesitate to mail or telephone me.

Michael Paddon
Iconix Pty Ltd, 851 Dandenong Road,
East Malvern, Victoria 3145, Australia.

Email: mwp@iconix.oz.au
Telephone: +61 3 571 4244
Facsimile: +61 3 571 5346

A.U.U.G. INCORPORATED

PROFIT AND LOSS STATEMENT

FOR THE PERIOD ENDED 1STJUNE 1992 TO 31ST MAY 1993.

INCOME	1993	1992
Membership	112,159.00	102,552.00
Advertising	780.00	2,300.00
Subscriptions	1,908.00	2,205.00
Mailing List	1,167.00	2,216.38
Interest Received	8,964.61	12,374.11
Other Income	424.00	589.00
Auug 93	11,210.00	-----
Summer 92		
- Canberra	5,555.00	-----
- Melbourne	300.00	3,655.00
- Sydney	60.00	2,190.00
- Adelaide	-----	1,819.00
- Perth	2,295.00	4,085.00
- Brisbane	4,625.50	-----
Security Video	-----	93.00
Summer Conference 92	595.00	-----
Uniforum Rebate	865.52	2,665.98
Netcomm	2,930.74	-----
Summer Conference 93	11,386.50	-----
AARNET Subscriptions	33,400.00	12,745.00
	198,625.87	149,489.47

LESS EXPENSES	1993	1992
Bank Charges		
- Credit Card	629.37	793.86
- Government	307.57	865.00
- General	-	661.15
	936.94	2,320.01
Management Commitee/ Meeting Expenses		
- Air Fares	9,467.18	4,390.00
- Accommodation / Meals	385.40	487.55
- Parking	-----	50.00
- Taxis	276.90	385.75
- Postage	-----	17.90
- Fuel	-----	71.19
- Refreshments	278.80	
	10,408.28	5,402.39
Membership		
- Refunds	12.00	234.00
- Printing	10,831.20	
	10,843.20	234.00
Netcomm	2,890.74	
AARNET Expenses	1000.00	16000.00
A.U.U.G. 90		708.00
A.U.U.G. 91	3,803.00	
A.U.U.G. 92		
- Airfare		476.00
A.U.U.G.N.		
-. Postage/Freight	------	2,983.60
- Printing	33,695.63	28,347.97
	33,695.63	31,331.57

Summer 92		
Perth - Seminar Function Costs	—	3,075.78
- Air fare	—	1,596.00
- Advertising	—	431.79
- Postage	—	89.67
- Printing/ Stationery	—	57.67
	—	5,250.91
Sydney		
- Seminar Costs	—	2,575.50
- Printing	—	215.70
- Photocopying	—	30.00
		2,821.20
Melbourne		
- Printing/ Advertising		1000.00
Adelaide		
- Taxi		54.20
- Air fare		478.00
- Accommodation		201.15
- Seminar Costs		302.35
- Hire of Equipment		200.00
	—	1,235.70
Summer 91		
Melbourne - Function		2250.00
Video Costs		36.00
Secretarit		
- Retainer	2,400.00	2,400.00
- Other	19,468.55	12,762.55
	21,868.55	15,162.55

	1993	1992
Chapters		
- VIC	2,199.54	
- ACT	5,777.01	
- W.A.	663.00	
	8,639.55	
Mailing List		
- Photocopying/ Printing		2,658.04
		2,658.04
Symmetry		
- Retainer	9000.00	27,0000.00
-	6,233.39	19,805.20
	15,233.39	46,805.20
Office		
- L Fraumann	43,216.60	-----
- Advertising	3,212.62	792.50
- Auditors Remuneration	1,888.75	1,950.00
- Freight/ Postage	4,168.11	2,137.44
- Mailing	-----	360.00
- Printing /Sationery	3,002.89	-----
- Telephone	4,780.47	1,770.33
- Secretary Fees	-----	6,084.59
- Wages	12,255.00	-----
- Taxis	236.16	-----
- General Expenses	2,993.00	169.90
- Travel Expenses	-----	3,825.54
- Air fare	320.20	310.00
	76,073.80	17,400.30
TOTAL OPERATING COSTS	**231,303.53**	**151,091.87**
General A/C Net Profit (Loss)	(32,677.66)	(1,602.40)
A.U.U.G. 92 Net Profit (Loss)	(2,063.19)	13,826.64
NET PROFIT	**($34,740.85)**	**$12,224.24**

A.U.U.G. INCORPORATED

PROFIT & LOSS

FOR THE PERIOD 1ST JUNE 1992 TO 31ST MAY 1993

	1993	1992
Conference A.U.U.G. 1992		
Income	10,875.17	22,901.64
Less Expenses		
Advertising/ Promotion	6,670.84	9,075.00
Photocopying/ Printing	1,532.80	-----
Travel & Accommodation	4,734.72	-----
	12,938.36	9,075.00
NET PROFIT (LOSS)	**($2,063.19)**	**$13,826.64**

A.U.U.G. INCORPORATED

BALANCE SHEET

AS AT 31ST MAY 1993

	NOTE	1993	1992
ASSETS			
CURRENT ASSETS			
Cash		8,149.43	4,046.46
Recievables	(3)	5,553.62	10,581.32
Investments	(2)(4)	121,222.24	155,038.36
		134,925.29	169,666.14
NON CURRENT ASSETS			
Inatangibles		988.10	988.10
TOTAL ASSETS		**$135,913.39**	**$170,654.24**
LIABILITIES & CAPITAL			
ASSOCIATED FUNDS			
Accumulated Profits		135,913.39	170,654.24
TOTAL LIABILITIES & CAPITAL		**$135,913.39**	**$170,654.24**

A.U.U.G. INCORPORATED

NOTES TO AND FORMING PART OF THE ACCOUNTS

FOR THE YEAR ENDED 31ST MAY 1993

1. ACCOUNTING POLICIES
The accounts are prepared in accordance with the historical cost convention. The Accounting policies adopted are consistent with those of the previous year.

2. INVESTMENTS

Investments are shown at Market Value, Capital Gains tax is not taken into account in determining the investments unless a definate decision to sell has been taken and the related Capital Gains Tax can be reliably estimated.

Dividends and other distributions from investments are taken to income on recievable basis.

3. CURRENT RECIEVABLES	1993	1992
Withholding Tax	5,553.62	3,701.32
Prepayment		
Membership		2,340.00
Newsletter		4,540.00
		$10,581.32

4. CURRENT INVESTMENTS		1993	1992
Quoted Investment			
C.B.A.	- Term Deposit	34,596.00	32,673.87
	- Cash Management	86,626.24	122,364.49
		$121,222.24	$155,038.36

AUDITORS REPORT TO THE MEMBERS OF

AUSTRALIAN OPEN SYSTEMS USERS GROUP INC.

In our opinion -

1. The accounts which have been prepared under the historical cost convention are properly drawn up in accordance with provisions of the Companies (Victoria) Code, and so as to give a true and fair view of :-

   (a) the state of the affairs of the organisation as at 31 st March 1993 and of the results of the organisation for the year ended on that date, and

   (b) the other matters required by section 269 of that Code to be dealt with in the accounts.

2. The accounting records and other records and the registers required by that code to be kept by the organisation have been properly kept in accordance with the provisions of that code.

NICOL & NICOL
C.P.A.

....
STUART C NICOL
PARTNER

Dated at Melbourne this 28 day of September 1993.

GET SERIOUS

GET RESULTS

RESULTS THROUGH OPEN SYSTEMS

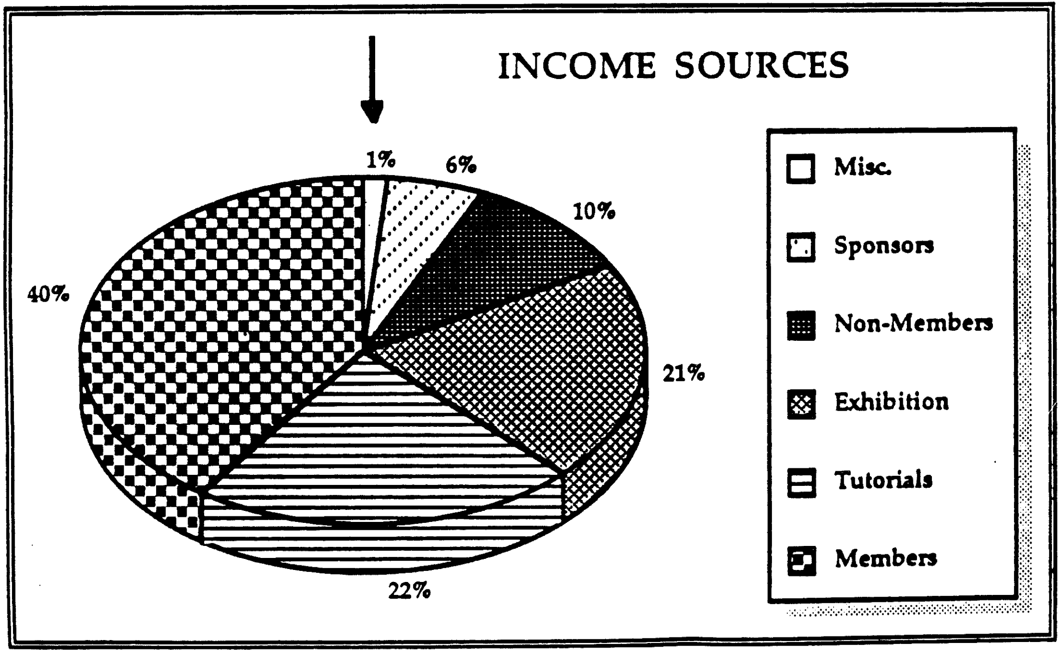
AUUG '93

So, how did AUUG spend the monies at AUUG '93? or Results through Open Communication...

by: Liz Fraumann

Each year the AUUG Executive Committee, most recently under the leadership of Phil McCrea, does its best to ensure the annual conference and exhibition is both high quality and of general interest. It is always a challenge to plan an event a year in advance attempting to predict what will be of significance and valuable to a wide and varied audience, especially in an industry who's life can and does change dramatically in the twinkle of an eye. The Programme committee, lead by Piers Lauder, should be extremely proud. The statistics clearly reflect AUUG '93 gave the members what they wanted.

As the chart (below) indicates, 40% of the income for AUUG '93 was derived from Member Registration. This is a positive and significant contribution. The next largest income was from the tutorials, of which AUUG members accounted for slightly over one half.



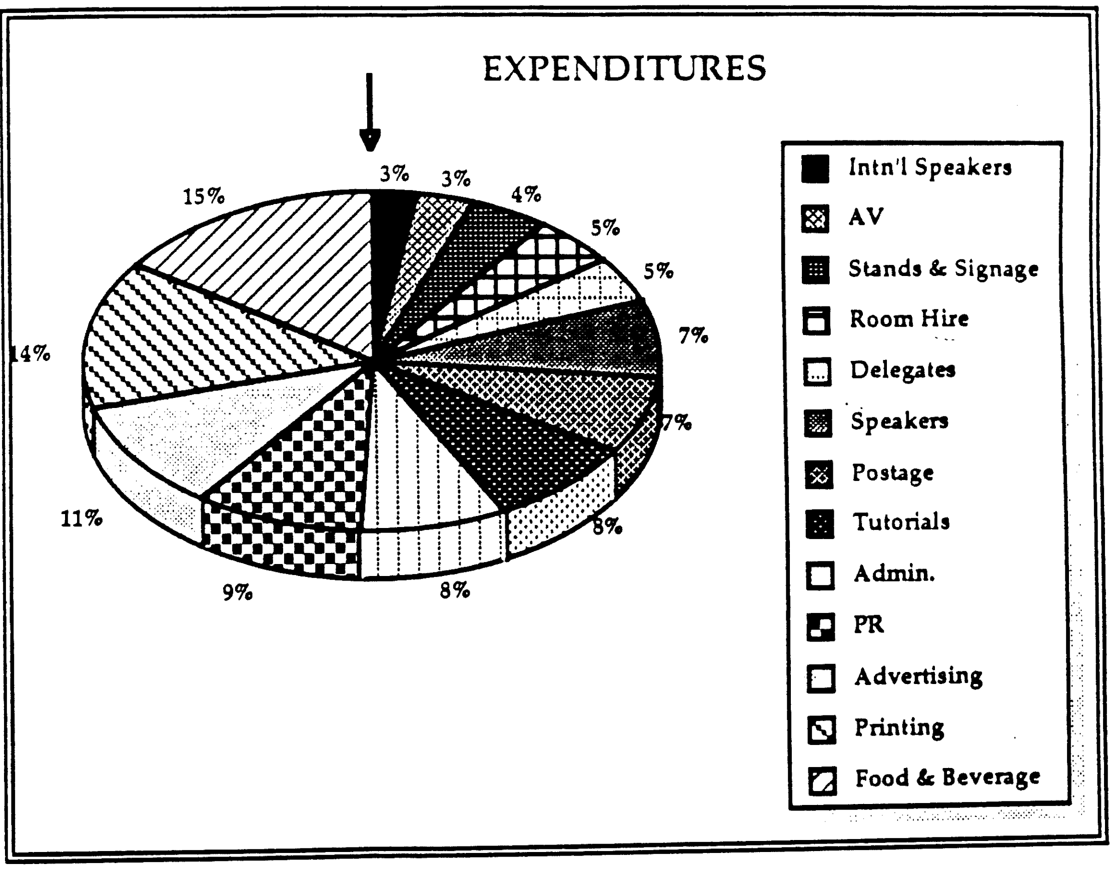
The exhibition profits, which from time to time comes under close scrutiny by some members, also puts forth a significant contribution to the overall success of the event.

An area where AUUG leadership must make some difficult decisions is to what degree should the pursuit of non-members attending the event take place. It is noteworthy at this point to say the insertion of the membership application in the registration brochure secured over 100 new members.

Results through Open Communication -- *Continued...*

As with all things in life, there is a "flip side." In the case of income, it is expenditure. Firstly, let me say, yes, the event was profitable. More important though, the Executive Committee felt it was appropriate to give the membership an indication of how the monies were spent.

This chart reflects only conference expenditures, as ACMS handles the exhibition. The largest expenditure is food and beverage. It should be noted this represents coffee/tea breaks, receptions, dinner, and all other consumables. It is, at times frustrating, to not be allowed to "bring in" food from other sources which may be less costly, but then there is the health factor which weighs heavily into any establishment providing food supplies. It was interesting many of the suggestions for improvements were to increase the choice and frequency of coffee/tea, juices, etc. for delegates. The '94 Programme committee, which is headed by Ian Hoyle of BHP, will undoubtedly take these and all other suggestions under advisement.



The next largest expense area is printing. In this category, the Conference Registration and Information Brochure and the Proceedings were the most costly. The '94 Programme committee will consider things like distribution of the proceedings on disk for future events. This may not only have a significant impact on production costs but it would be environmentally friendly.

A few people had indicated on the evaluation forms that we should spend less on international speakers. The majority of the speakers coming from overseas were sponsored by their organisations or by solicited sponsors. The percentage figure indicated here includes the package offered to all speakers. Anyone who has participated knows, it is not a commitment to be taken lightly and AUUG recognises the effort which is put forth and appropriately rewards it.

Results through Open Communication
Continued...

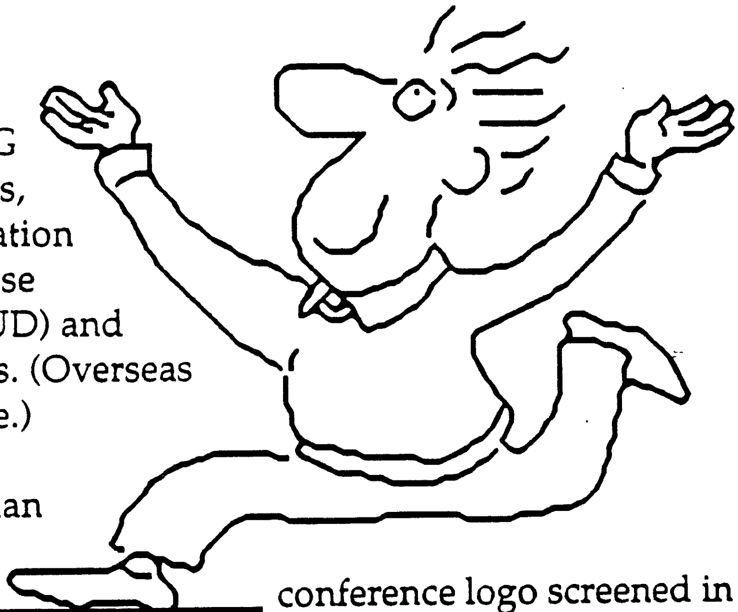
AUUG '94 has already commenced in planning. As indicated earlier, Ian Hoyle was selected to be the Chair for the event, and he in turn has selected his Conference Committee. As always we look forward to the participation from the membership at large and welcome suggestions which will continue to improve the overall success. If you would like to offer your ideas, please e-mail them to: auug94c@bhp.com.au If you do not have e-mail access, please post them to :

AUUG '94 Committee
P.O. Box 366
Kensington, NSW 2033

The Call for Papers will be issued and posted on aus.org.auug during the month of December. Watch closely so you too may seize the opportunity to share your latest developments or strategies with delegates at AUUG '94.

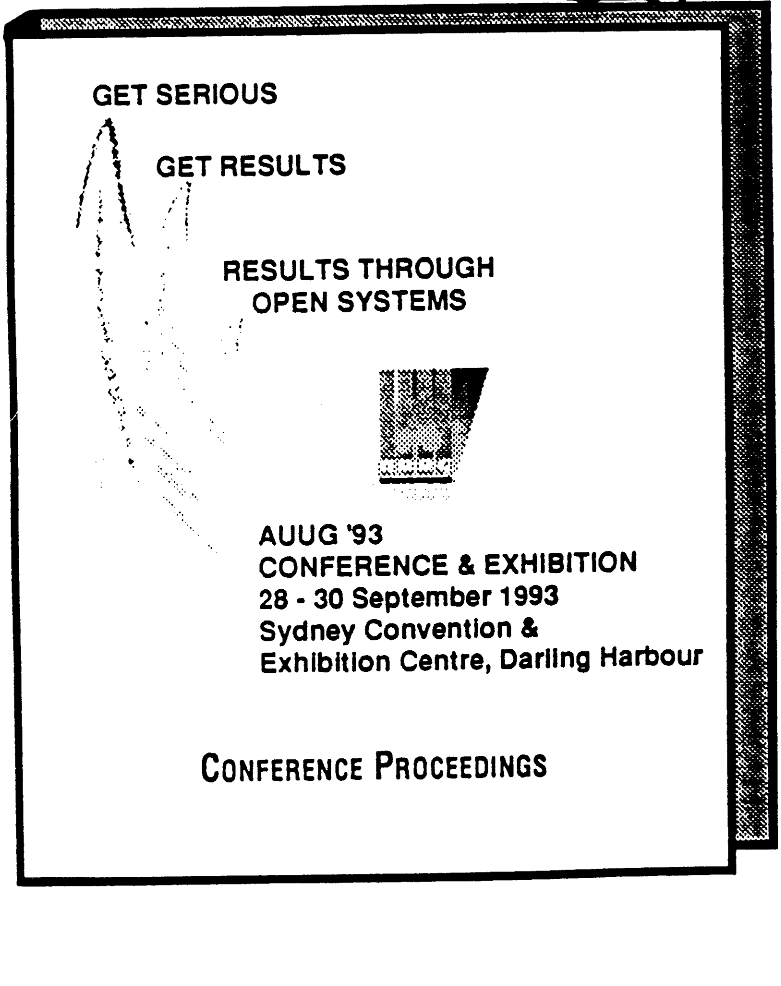
Opportunity knocks softly sometimes...

If you missed AUUG '93 there is now a chance for you to at least purchase one of the few remaining copies of the proceedings; also one of the T-shirts!



The proceedings contain the full papers presented at AUUG '93, a volume of over 300 pages, complete with contact information for authors of the papers. These proceedings sell for $50.00(AUD) and include postage/handling fees. (Overseas orders will be charged postage.)

T-shirts, 100% cotton, Australian made, have the AUUG '93 conference logo screened in burgundy and black on the left chest. Tastefully done and classy enough to even ware to work or the Local Chapter meetings. T-shirts sell for $10.00 each, including postage & handling. Please specify L or XL size when ordering. Sizes are limited, first ordered, first served.



GET SERIOUS

GET RESULTS

RESULTS THROUGH
OPEN SYSTEMS

AUUG '93
CONFERENCE & EXHIBITION
28 - 30 September 1993
Sydney Convention &
Exhibition Centre, Darling Harbour

CONFERENCE PROCEEDINGS

TO ORDER:

Send a fax with item, quantity, size if needed to: AUUG '93 Secretariat +61 2 332-4066. Credit cards will be accepted with proper authorisation and signature.

An~~nouncement~~ →

Summer Conference Series
AUUG
'94
ACT • NSW • NT • QLD • SA • TAS • VIC • WA

For further information contact:
Organisers listed or,
Glenn Huxtable - VP - (09) 380-2878 tel
Liz Fraumann - Business Manager (02) 953-3542 tel/fax

WHAT: Berkeley 4.3 - 4.4BSD Workshops
WHEN: February - March 1994
WHERE: All states

The AUUG '94 Summer Conference Series promises to be very exciting for most participants. This year, the local chapters are providing a one to three day workshop by Dr. Marshall Kirk McKusick, in addition to the conference and workshops/tutorials.

Kirk's experience with the UNIX Operating System hails back to the early 1970's. He was instrumental in the development of Berkeley UNIX 4.3 and 4.4BSD. In fact, the was the principal developer of that operating system. He is often a guest speaker at the USENIX Conferences, and is a past president of that organisation. He received a Ph. D. in the area of programming languages from the University of California at Berkeley.

In the Berkeley tradition, Kirk is intelligent and witty. Cliff Stoll had his microwaved sneakers... Kirk has his pink flamingos. It is another great story and worth an extra coffee!

On a serious note, Kirk's workshops will be both informative and intense. Following is an overview and schedule for both the one and three day versions.

One Day Workshop:

An Introduction to UNIX Kernel Internals: Data Structures and Algorithms

Who Should Take this Course

This course provides a broad overview of how the UNIX kernel provides its basic services. It will be most useful to those who need to learn how these services are provided. Individuals involved in technical and sales support can learn the capabilities and limitations of the system; applications developers can learn how to effectively and efficiently interface to the system; systems programmers without direct experience with the UNIX kernel can learn how to maintain, tune, and interface to such systems. This course is directed to users who have had at least a year of experience using the UNIX system and the C programming language. They should have an understanding of fundamental algorithms (searching, sorting, and hashing) and data structures (lists, queues, and arrays). Students will not need to prove relationship with a source license holder, as actual source code will not be presented.

AUUG INC. SECRETARIAT
PO BOX 366 KENSINGTON NSW 2033
70 GLENMORE ROAD
PADDINGTON NSW 2021 AUSTRALIA
INC. IN VICTORIA ACN A0016636N
PHONE: +61 2 361-5994
FAX: +61 2 332-4066

more-more-more

Berkeley 4.3 & 4.4BSD Workshops

Description

This course will provide a firm background in the UNIX kernel. The course includes coverage of most BSD-derived kernels including USL's System V Release 4, Sun's Solaris, and DEC's Ultrix. The POSIX kernel interfaces will be used as examples where they are defined. Where they are not defined, the BSD interfaces will be described and then related to other vendors interfaces. The course will cover basic kernel services, process structure, virtual and physical memory management, scheduling, paging and swapping. The kernel I/O structure will be described showing how I/O is multiplexed, special devices are handled, character processing is done, and the buffer pool is managed. The implementation of the file system and its capabilities will be described. The file system interface will then be generalized to show how to support multiple file system types such as Sun Microsystem's Network File System (NFS). The course will conclude with a brief introduction to the interprocess and networking capabilities of the system. It will provide an overview of networking terminology and an example use of the socket interface. The presentation will emphasize code organization, data structure navigation, and algorithms. It will not cover the machine specific parts of the system such as device drivers.

Session 1 - Kernel Overview
- Kernel terminology
- Basic kernel services
- Process structure

Session 2 - Kernel Resource Management
- Virtual memory management
- Scheduling
- Signals

Session 3 - Kernel I/O structure
- Special files
- Terminal handling
- File system implementation
- Network File System (NFS)

Session 4 - Networking and Interprocess Communication
- Concepts and terminology
- Basic IPC services

Course Text

Samuel J. Leffler, Marshall Kirk McKusick, Michael J Karels, and John S. Quarterman, ``The Design and Implementation of the 4.3 BSD UNIX Operating System'', Addison-Wesley Publishing Company, Reading, Massachusetts, 1989, 496 pages.

more-more-more

Berkeley 4.3 & 4.4BSD Workshops

Three Day Workshop (presented only in Melbourne):

UNIX Kernel Internals:
Data Structures, Algorithms, and Performance Tuning

Dr. Marshall Kirk McKusick
University of California at Berkeley

Who Should Take this Course

This course provides a broad overview of how the UNIX kernel provides its basic services. It will be most useful to those who need to learn how these services are provided. Individuals involved in technical and sales support can learn the capabilities and limitations of the system; applications developers can learn how to effectively and efficiently interface to the system; systems programmers without direct experience with the UNIX kernel can learn how to maintain, tune, and interface to such systems. This course is directed to users who have had at least a year of experience using the UNIX system and the C programming language. They should have an understanding of fundamental algorithms (searching, sorting, and hashing) and data structures (lists, queues, and arrays). Students will not need to prove relationship with a source license holder, as actual source code will not be presented.

Description

This course will provide a firm background in the UNIX kernel. The course includes coverage of most BSD-derived kernels including USL's System V Release 4, Sun's Solaris, and DEC's Ultrix. The POSIX kernel interfaces will be used as examples where they are defined. Where they are not defined, the BSD interfaces will be described and then related to other vendors interfaces. The course will cover basic kernel services, process structure, virtual and physical memory management, scheduling, paging and swapping. The kernel I/O structure will be described showing how I/O is multiplexed, special devices are handled, character processing is done, and the buffer pool is managed. The implementation of the file system and its capabilities will be described. The file system interface will then be generalized to show how to support multiple file system types such as Sun Microsystem's Network File System (NFS). Other related topics include performance measurement, system tuning, and security issues. The introduction to the interprocess and networking capabilities of the system will provide an overview of networking terminology and an example use of the socket interface. The presentations will emphasize code organization, data structure navigation, and algorithms. It will not cover the machine specific parts of the system such as device drivers.

Day 1 morning - Kernel Overview
- Kernel terminology
- Basic kernel services
- Process structure

Day 1 afternoon - Kernel Resource Management
- Virtual memory management
- Paging and swapping
- Scheduling
- Signals

AUUG INC. SECRETARIAT
PO BOX 366 KENSINGTON NSW 2033
70 GLENMORE ROAD
PADDINGTON NSW 2021 AUSTRALIA
INC. IN VICTORIA ACN A00166З6N
PHONE: +61 2 361-5994
FAX: +61 2 332-4066

more-more-more

Day 2 morning - Kernel I/O structure
- Special files
- Terminal handling
- Multiplexing I/O
- Autoconfiguration strategy
- Structure of a disk device driver

Day 2 afternoon - File System Overview
- File system services
- Block I/O system (buffer cache)
- File system implementation
- Support for multiple file systems
- Network File System (NFS)

Day 3 morning - System Tuning
- Performance measurement
- System tuning
- Crash dump analysis
- Security issues

Day 3 afternoon - Interprocess Communication
- Concepts and terminology
- Basic IPC services
- Example use of IPC and network facilities

Course Text

Samuel J. Leffler, Marshall Kirk McKusick, Michael J Karels, and John S. Quarterman, ``The Design and Implementation of the 4.3 BSD UNIX Operating System'', Addison-Wesley Publishing Company, Reading, Massachusetts, 1989, 496 pages.

For dates, and venue of these workshops please consult the AUUG Inc. Summer Conference Series Announcement in this issue of *AUUGN*. For further details, contact your Local Chapter representative.

AUUG INC. SECRETARIAT
PO BOX 366 KENSINGTON NSW 2033
70 GLENMORE ROAD
PADDINGTON NSW 2021 AUSTRALIA
INC. IN VICTORIA ACN A0016636N
PHONE: +61 2 361-5994
FAX: +61 2 332-4066

Ann~~ouncement~~ →

Summer Conference Series
AUUG
'94

ACT • NSW • NT • QLD • SA • TAS • VIC • WA

For further information contact:
Organisers listed or,
Glenn Huxtable - VP - (09) 380-2878 tel
Peter Wishart - Sec. - (06) 275-0908 tel
Liz Fraumann - Business Manager (02) 953-3542 tel/fax

WHAT: AUUG Summer Conference Series
WHEN: February - March 1994
WHERE: All states

AUUG Inc. is pleased to announce its summer technical conference series for 1994. This year, in addition to the conference and workshops AUUG members are familiar with, the Local Chapters are offering a special tutorial by Dr. Kirk McKusick, principal developer of Berkeley UNIX® 4.3 and 4.4BSD on Berlekey UNIX.

Local organisers are hosting 1-3 day sessions in each state across Australia. Interested parties should contact the conference organiser listed below.

Melbourne: *Venue:* Clunies Ross House
Feb. 3-5 McKusick Workshop
Organiser: Stephen Prince
Chancery Lane Computer Services
Level 25
385 Bourke St.
Melbourne, 3000
(03) 608-0911 ph 608 0505 fax
email : sp@clcs.com.au

Darwin: *Venue* - NT University
Feb. 9 McKusick Workshop
Feb. 10-11 Conference & Workshops
Organiser: Phil Marker
Dept of Computer Science
Northern Territory University
P.O. Box 40146
Casuaruna, NT 0811
(089) 466-382 tel (089) 270-612 fax
e-mail: pjm@pandanus.ntu.edu.au

Hobart: *Venue* - University of Tasmania, Hobart, Centenary Lecture Theatre
Feb. 15 Conference
Organiser: Steven Bittinger
ITS
University of Tasmania
GPO Box 252C
Hobart, TAS 7001
(002) 207-406 tel • (002) 207-488 fax
e-mail: Steven Bittinger@its.utas.edu.au

Canberra: *Venue* -Work shops - Copland Bldg., ANU
Venue -Conference - National Convention Centre
Feb. 14 McKusick Workshop
Feb. 15-16 Workshops & Conference
Organiser: John Barlow
CISR, IBLOCK
ANU 0200 ACT
(06) 249-2930 tel • (06) 249-0747 fax
email : john.barlow@anu.edu.au

Perth: *Venue:* - Orchard Hotel
Feb. 22 McKusick Workshop
Feb. 23 Conference
Organiser: Adrian Booth
Adrian Booth Computing Consultants
7 Glenrowan Place
Willetton, WA 6155
(09) 354-4936 tel • (09) 354-4936 fax
e-mail: abcc@dialix.oz.au

UNIX is a registered trademark of X/Open in the U.S. and other countries.

AUUG INC. SECRETARIAT
PO BOX 366 KENSINGTON NSW 2033
70 GLENMORE ROAD
PADDINGTON NSW 2021 AUSTRALIA
INC. IN VICTORIA ACN A0016636N
PHONE: +61 2 361-5994
FAX: +61 2 332-4066

more-more-more

AUUG Inc. Summer Conference Series Announced

Adelaide: *Venue* - TBD
Feb. 28 McKusick Workshop
Mar. 1 Conference
Organiser: Michael Wagner
Systems Services Pty. Ltd.
32 Grenfell St.
Adelaide, SA 5000
(08) 212-2800 tel • (08) 231-0321 fax
e-mail: mhw@syserv.com.au

Sydney: *Venue* - TBD
Mar. 3 McKusick Workshop
Mar. 4 Conference
Organiser: Brenda Parsons
P.O. Box 43
Darlinghurst 2010
(02) 131-003 Ext. 2403 tel
(02) 808 2797 fax
e-mail: parson@coulomb.pcc.oz.au

Hobart: *Venue* - University of Tasmania Staff Club (Hobart)
Mar. 9-10 McKusick Workshop
Organiser: Steven Bittinger
ITS
University of Tasmania
GPO Box 252C
Hobart, TAS 7001
(002) 207-406 tel • (002) 207-488 fax
e-mail: Steven Bittinger@its.utas.edu.au

Melbourne: *Venue*– Clunies Ross House
Mar. 15-16 Tutorials & Conference
Organiser: Arnold Pears
Computer Science and
Computer Engineering
La Trobe University
Bundoora, VIC 3083
(03) 479-1144 tel • (03) 470-4915 fax
e-mail: pears@latcs1.lat.oz.au

Tutorials: Robert Sturrock
email: rns@deakin.edu.au

Brisbane: *Venue* - TBD
Mar. 24 McKusick Workshop
Mar. 25 Conference
Organiser: Michael Henning
CiTR
University of Queensland 4072
(07) 365-4321 tel • (07) 365-4399 fax
e-mail: michi@citr.uq.oz.au

AUUG Inc. as a user group, exists to provide UNIX and open systems users with relevant and practical information, services, and education through cooperation among users.

AUUG INC. SECRETARIAT
PO BOX 366 KENSINGTON NSW 2033
70 GLENMORE ROAD
PADDINGTON NSW 2021 AUSTRALIA
INC. IN VICTORIA ACN A0016636N
PHONE: +61 2 361-5994
FAX: +61 2 332-4066

Summer Conference Series '94

ACT • NSW • NT • QLD • SA • TAS • VIC • WA

AUUG Inc. • P.O. Box 366, Kensington, NSW 2033
(02) 361-5994 tel • (02) 332-4066 fax

Call for Papers

Fifth Annual Canberra Conference and Workshops

Canberra Chapter, AUUG Inc.
14/15/16th February 1994

A conference on UNIX and Open Systems.

The Canberra Chapter of AUUG Inc. is holding a summer conference on Wednesday February 16th 1994, in Canberra, ACT. This notice is a "call for papers" - we would like all people interested in giving a presentation to contact us and discuss the ideas you have. The conference talks should be about 20 minutes long (or possibly 50 minutes). The Program Chair for the 1994 Summer Conference is Jeremy Bishop. Please contact him on (06) 265 0431 (ph), (06) 265 0485 (fax) or jeremyb@helotrix.defcen.GOV.AU (e-mail) if you are interested in presenting a paper.

Workshops will be presented on Tuesday the 15th of February and should be 3 hours long (maybe 6 hours if required). If you are interested in presenting a workshop, please contact the workshop organiser David Baldwin on (06) 2495026 (ph), (06) 2493992 (fax), or David.Baldwin@anu.edu.au.

General Enquires:
John Barlow
Parallel Computing Research Facility, I-Block,
Australian National University
Canberra, 0200, Australia
(06) 249-2930 wk • (06) 282-1925 hm • (06) 249-0747 fax,
e-mail: John.Barlow@anu.edu.au

Summer Conference Series '94

AUUG Inc. • P.O. Box 366, Kensington, NSW 2033
(02) 361-5994 tel • (02) 332-4066 fax

ACT • NSW • NT • QLD • SA • TAS • VIC • WA

CALL FOR PAPERS
CALL FOR PARTICIPATION

Annual AUUG-NSW Summer Conference

3-4 March, 1994.

Each year AUUG hosts an annual summer conference in a number of regional centres around Australia. The next summer conference for NSW will be held on Friday, 4th March, 1994. In conjunction with the conference, a one day programme of workshops is also planned for Thursday, 3rd March, 1994. The aim of the event is to provide a local forum for the exchange and discussion of ideas and work in the Open Systems arena. It is a supplement to the AUUG National Winter Conference, but with particular emphasis on technical issues and content.

Participation in the conference is open to everyone, not just AUUG members. Presentations are invited from anyone involved in areas of UNIX or Open Systems activity, as well as people interested in giving a full or half day workshop.

SUBMISSION INFORMATION

Participants are invited to submit full papers or abstracts, preferably by email, to the program organizer, on or before Friday, 17th Dec, 1993. Email submissions should be in troff or postscript form. Postal submissions should consist of at least two A4 paper copies of the paper or abstract. Submissions should be written in English. Final papers should be no more than 12 pages in length and should be camera-ready.

Proposals for workshops topics should outline the format and content of the presentation, and clearly list the required presentation aids.

NOTIFICATION

Authors will be notified of acceptance by Monday, 17th January, 1994.

IMPORTANT DATES

Fri, 17 Dec, 1993.	Deadline for submission of proposals.
Mon, 17 Jan, 1994.	Notification of acceptances.
Mon, 7 Feb, 1994.	Deadline for final papers and workshop papers.

PROGRAM CHAIR

Brenda Parsons
P.O. Box 43
Darlinghurst 2010

Email: parson@coulomb.pcc.oz.au

Phone: (02) 131-003 Ext. 2403 (bh)
(02) 807 3862 (ah)

Fax: (02) 808 2797

ORGANIZING COMMITTEE

For more information about the conference, please contact any of the NSW Chapter Committee members. We can be reached by email at:

nsw_auug@dwt.csiro.au

or individually as follows:

Brenda Parsons (Secretary/Treasurer)
Email: parson@coulomb.pcc.oz.au
Phone: (02) 807 3862 (ah)
Fax: (02) 808 2797

Julian Dryden (President)
Email: julian@dwt.csiro.au
Phone: (02) 809 9345 (bh)
Fax: (02) 809 9476

Farrell McKay
Email: Farrell.McKay@nms.otc.com.au
Phone: (02) 452 2486 (ah)

Summer Conference Series '94

AUUG Inc. • P.O. Box 366, Kensington, NSW 2033
(02) 361-5994 tel • (02) 332-4066 fax

ACT • NSW • NT • QLD • SA • TAS • VIC • WA

AUUG Inc. Northern Territory Chapter

Annual AUUGwet Conference and Workshops
9, 10, 11 February 1994

Call for papers.

The Northern Territory Chapter of AUUG Inc. is holding its "AUUGwet 94" summer conference in Darwin in February 1994.

Important Dates:

Wed 9th	Kirk McKusick Workshop 4.4Bsd
Thu 10th	Main Conference - presentation of papers (+ Conference dinner)
Fri 11th	Local Workshops

General Information:

An invitation is now extended for presentation of technical papers on day 2 of the conference. The standard conference talk will be about 20 minutes long , with 10 minutes question time following.

The third day of the conference on Friday 11th will be devoted to workshop topics. Anyone who might like to present a technical paper or run a workshop is invited to contact us to develop a presentation theme.

The theme of this year's conference will be "Open Systems in North Australia"; a wide topic designed to encourage a broad range of presentations.

The 1993 conference was attended by around 75 delegates mostly from the commercial, government and education sectors. Most had some active professional involvement in the movement towards open systems computing; with a strong leaning towards the UNIX environment. We had a broad mixture of technical and non-technical people.

Interstate or International speakers are most welcome. Assistance with tour itineraries taking in the sights of the magnificent Top End wet season will be given with pleasure.

Technical Papers

The 1993 presentation topics were:
Software Development Environments for Distributed Processing.
An Application Development in Stereo-Graphics.
System Building for a GIS Application.
Computing Applications in the Medical Research Environment.
Multimedia Developments for Open Environments. (Student prize winner)
Standards and Open Systems Technologies.
An Application Development with TITAN Text Retrieval Software.
GOSIP.
OSI Immigration at the Bureau of Meteorology.
Open Systems- The Maturing of the IT Industry.
Application of IT to DME Library System.
Relational Database and Distributed Processing.
An Application Development using MS Windows Environment.

Most 1993 presentations covered particular aspects of the named topics; presentations with similar themes would be quite welcome in 1994.

There will be a prize for the best program presented in the "Student" category.

Workshops

The length of these will be dependant on the topic, the presenters preferred approach, and the conference timetable. Workshops of 1 - 3 hours length are generally envisaged.

The topics last year were:
CSIRO Project LISA - Land Information System for South Australia.
Unix - an overview.
System Administration
Networks and Communications

Organisational Details

Programme Chair:	Phil Maker
Phone:	089-466666
Fax:	089-270612
Email:	pjm@it.ntu.edu.au
Post:	NTU School of IT, PO Box 40146 Casuarina 0811

*****NOTE** Dates have been changed since previous notices.**

Bernie Dowd
Secretary
AUUGnt
24 November 1993

Summer Conference Series '94

ACT • NSW • NT • QLD • SA • TAS • VIC • WA

AUUG Inc. • P.O. Box 366, Kensington, NSW 2033
(02) 361-5994 tel • (02) 332-4066 fax

AUUG Queensland Chapter

1994 Summer Technical Conference

Preliminary Announcement and Call for Papers

The AUUG summer conference series provides a forum for discussion of technical issues and developments relating to UNIX and all aspects of open systems. The AUUG Queensland Summer Technical Conference will be a one day conference, to be held on Friday, 25th March 1994, in or near the Brisbane CBD (exact details to be announced).

The intended format is a number of presentations to an audience of people interested in the technical aspects of UNIX and UNIX applications. A number of invited interstate speakers will provide insight into topics of particular interest.

Call for Papers:

Authors are invited to submit papers presenting new or interesting results of current research, development, porting or implementation experience. This includes submissions on any other topic relating to technical issues of UNIX and open systems.

Speaker Incentives:

Presenters of papers are afforded free conference registration. Limited funding is available to interstate and remote Queensland speakers to assist with travel expenses.

Form of Submissions:

Please submit an abstract for your paper, together with an outline. The outline should contain enough detail to allow the programme committee to make a reasoned decision about the final paper. Of course, a full paper is also perfectly acceptable. A submission should be from 2-5 pages and include:

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

2. A biographical sketch not to exceed 100 words.

3. Abstract: 100 words.

4. Outline: 1-4 pages giving some details of the approach used, algorithms implemented, sections of kernel hacked, or any other relevant information.

5. References to any relevant literature.

6. Audio-visual requirements: Please indicate your requirements for overhead projector, 35mm slide projector or video/television equipment.

Acceptance:

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

Organising Committee:

Sean Appleby - Sean Appleby Consulting
Greg Birnie - Leeds & Northrup Australia
Tim Butterfield - C&T Computing
John Havers - Stallion Technologies
Michael Henning - CiTR
David Hughes - Bond University
Stuart Remphrey - Wang
Rick Stevenson - Stallion Technologies
Ian Turner - IBIS Computer

Relevant Dates:

Abstract and outlines due: 28 January 1994
Notification to authors: 4 February 1994
Final papers due: 11 March 1994

Addresses:

Please submit hard copy or electronic copy of abstracts and outlines to:

QAUUG Conference Committee
CiTR
Gehrmann Laboratories
Research Rd
University of Queensland 4072

e-mail: michi@citr.uq.oz.au
Phone : (07) 3654321
FAX : (07) 3654399

For enquiries on conference registration, accomodation arrangements, promotion, venue and other matters not relating to the submission of papers, contact:

Tim Butterfield
C&T Computing
Unit 2, 240 Vulture St
South Brisbane 4101

Phone: (07) 8443263
FAX : (07) 8445592

Summer Conference Series '94

AUUG Inc. • P.O. Box 366, Kensington, NSW 2033
(02) 361-5994 tel • (02) 332-4066 fax

ACT • NSW • NT • QLD • SA • TAS • VIC • WA

CALL FOR PAPERS

AUUG SUMMER '94

1994 AUSTRALIAN UNIX AND OPEN SYSTEMS USERS GROUP SOUTH AUSTRALIA SUMMER CONFERENCE

28 February - 1 March, 1994

Each year AUUG Local Chapters host annual summer conferences across Australia. The next summer conference for South Australia will be held on Tuesday, 1 March 1994. The day preceding the conference, 28 February 1994, this year is dedicated to a special tutorial on Berkeley UNIX 4.3 & 4.4BSD, presented by the principal designer, Dr. Marshall Kirk McKusick. In addition to Dr. McKusick's tutorial the group will offer a few additional tutorials. The aim of the two day event is to provide a local forum for the exchange and discussion of ideas and work in the open systems arena. This is a supplement to the AUUG National Winter Conference.

Participation in the conference is open to everyone, not just AUUG members. Presentations are invited from anyone involved in the areas of Open Systems or UNIX activity, as well as people interested in giving a full or half day tutorial.

SUBMISSION INFORMATION

You are invited to submit full papers or abstracts, preferably by e-mail to the programme organiser, on or before Friday, 7 January 1994. Submissions should be written in English. Final papers should be no more than 15 pages in length and should be camera-ready.

Proposals for tutorial topics should outline the format and content of the presentation and clearly list the required presentation aids.

NOTIFICATION

Authors will be notified of acceptance by Monday, 24 January 1994.

IMPORTANT DATES

Friday, 7 January 1994 - Deadline for submission of papers/tutorial topics
Monday, 24 January 1994 - Notification of acceptances
Monday, 14 February 1994 - Deadline for final papers and workshop papers

PROGRAMME CHAIR

Michael Wagner
Systems Services Pty. Ltd
Level 6, 32 Grenfell St.
Adelaide, SA 5000

(08) 212-2800 tel
(08) 231-0321 fax

e-mail: mhw@syserv.com.au

ORGANISING COMMITTEE

Alastair Dick
Telecom Australia
GPO Box 342
Adelaide, SA 5001

(08) 230 7062 tel
(08) 410 0023 fax

e-mail: adick@stnecss.telecom.com.au

Bob Korbel
Amdel Ltd.
P.O. Box 338
Torrensville, SA 5031

(08) 416-5355 tel
(08) 234-5882 fax

Rodger S.B. Chan
The University of Adelaide
Adelaide, SA 5005

(08) 303-5873 tel
(08) 303-4361 fax

e-mail: catsmgr@aelmg.adelaide.edu.au

Summer Conference Series '94

AUUG Inc. • P.O. Box 366, Kensington, NSW 2033
(02) 361-5994 tel • (02) 332-4066 fax

ACT • NSW • NT • QLD • SA • TAS • VIC • WA

CALL FOR PAPERS

****

AUUG SUMMER '94

1994 AUSTRALIAN UNIX USERS GROUP
VICTORIAN SUMMER TECHNICAL CONFERENCE

March 15 - 16 , 1994

Clunies Ross House

Parkville, Melbourne, Australia

****

Program Committee Chair: Arnold Pears, La Trobe University, Melbourne.

Tutorials : Jon Eaves, TUSC Computer Systems, Melbourne.

****

The purpose of the AUUG regional summer technical conference is to supplement the AUUG winter conferences by providing an informal, technical forum without vendor exhibitions for the presentation and exchange of current work in the area of UNIX. It is expected that the content of presentations at this conference will address technical issues which are relevant to programmers, system administrators and experienced users.

The Victorian chapter AUUG summer conference this year will follow a two day programme.

Day 1: Tutorials
Day 2: Conference

Proposals are invited for both tutorials and papers for the conference. Area of interest include, but are not limited to the following.

* Unix
* Open Systems
* Emerging and New Technologies
* Software Advances
* Networking

Authors are invited to submit three copies of a 500 word abstract to the Programme Committee Chair by the submission deadline. Final papers should be written in English, and should be at most 12 pages. Each paper should include a short abstract and a list of keywords indicating subject classification.

Tutorial proposals should be submitted to the Tutorial Co-ordinator. Tutorials are expected to run for approximately three hours, and the proposed format and required teaching aids should be clearly outlined in the proposal.

Email submission of abstracts, papers and tutorial proposals in postscript, or troff is encouraged.

Authors of accepted papers will be asked to prepare a final camera ready version of their paper for publication in the proceedings. Copies of the proceedings will be issued to delegates, and a limited number of additional copies will be available at the conference. Speakers accepted to make a presentation will receive complementary admission to the papers section of the conference.

Programme Chairs:
****

Programme Committee:
========

Arnold Pears
La Trobe University,
Bundoora, 3083.
Email: pears@latcs1.lat.oz.au
Ph: (03) 479-1144
Fax: (03) 470-4915

Tutorial Co-ordinator:
========

Jon Eaves
TUSC Computer Systems,
666 Doncaster Road,
Doncaster 3108.
Email: jon@tusc.oz
Ph: (03) 840-2222
Fax: (03) 840-2277

Program Committee:
=================

Chair: Arnold Pears, La Trobe University.
Tutorials: Jon Eaves, TUSC Computer Systems,
Steven Prince, Chancery Lane Computer Services.
Neil Murray, Webster Computer Corporation.
Michael Paddon, Iconix.
John Carey, Labtam Australia.

=*=

IMPORTANT DATES

Dec. 6, 1993 (Mon) Abstract/Tutorial submission deadline
Dec. 20, 1993 (Tue) Notification of acceptance
Feb 21, 1994 (Mon) Final papers/tutorials due
==

For further information, please contact:

Arnold Pears
Organizing Committee Chairman
Computer Science and Computer Engineering
La Trobe University
Bundoora, Vic. 3083.

Phone: (03) 479-1144
Fax: (03) 470-4915
Email: pears@latcs1.lat.oz.au

Summer Conference Series '94

AUUG Inc. • P.O. Box 366, Kensington, NSW 2033
(02) 361-5994 tel • (02) 332-4066 fax

ACT • NSW • NT • QLD • SA • TAS • VIC • WA

Announcement and Call For Paper
for
Perth Summer Technical Conference

Wednesday, February 23, 1994

The Perth AUUG 1994 Summer Technical Conference will be held on February 23, 1994. Speakers are invited to speak on technical issues related to Unix and Unix-like systems, including but not limited to application development, system and network administration, and operating system research and implementation.

Prospective speakers are invited to submit abstracts to the Perth conference organiser, Adrian Booth, whose contact details appear below. Selected interstate speakers will have their travel and accommodation paid for by AUUG Inc. All speakers will receive complimentary conference registration.

Speakers at the Summer Conferences around Australia may be invited by AUUG Inc. to deliver their paper at the AUUG 1994 Winter Conference in Melbourne, with airfare, accommodation and registration paid by AUUG Inc.

A tutorial program will also be run in conjunction with the conference. A highlight will be a tutorial from Kirk McKusick (one of the principal architects of Berkeley Unix) on the internals of 4.4BSD. However, other tutorial presenters are also sought.

IMPORTANT DATES:

Closing date for abstracts:	January 7, 1994.
Finished papers due:	February 4, 1994.
Conference date:	February 23, 1994.

PROGRAMME CHAIR

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

email:	abcc@dialix.oz.au
Ph:	(09) 354 4936
Fax:	(09) 354 4936

Summer Conference Series

'94

AUUG Inc. • P.O. Box 366, Kensington, NSW 2033
(02) 361-5994 tel • (02) 332-4066 fax

ACT • NSW • NT • QLD • SA • TAS • VIC • WA

Tasmanian Summer Unix schedule

For details on paper submissions to the Annual Tasmanian Summer UNIX Conference please contact:

Annual Tasmanian Summer Unix Conference
15 February 1993
Venue: University of Tasmania, Hobart, Centenary Lecture Theatre
Time: 9am - 5pm, lunch included
Contact: Steven Bittinger, convenor
(002) 207406 fax (002) 207488
Steven.Bittinger@its.utas.edu.au

Tutorial: 9-10 March 1993
An Introduction to UNIX Kernel Internals:
Data Structures and Algorithms
Dr. Marshall Kirk McKusick
Venue: University of Tasmania Staff Club (Hobart)
Contact: Steven Bittinger, convenor
(002) 207406 fax (002) 207488
Steven.Bittinger@its.utas.edu.au

Announcement

For further information contact:
Organisers listed or,
Liz Fraumann - Business Manager (02) 953-3542 tel/fax

WHAT: UPDATE - USENIX Conferences
WHEN: Jan 17 - 21 1994 and June 6 - 10, 1994
WHERE: U.S.A.

In the spirit of cooperation, AUUG Inc. is pleased to announce advanced registration for the USENIX Winter 1994 Conference. The event will be held Jan. 17 - 21, 1994 in San Francisco. CA, U.S.A. Technical sessions will run from Monday, 17 January through the 19th, followed by two days of tutorials, 20 January through Friday, 21 January 1994. Delegates can save $100.00(US) by registering BEFORE 27 December 1993.

Conference Sessions headings are as follows:

17 Jan '94	Simple Database Tools
	UNIX on Wall Street
	Wide-Area Information Access
	Microkernels & Other Kernel Architectures & Experiences with Distributed Multiprocessor Systems (SEDMS IV)
	Interprocess Communication
	Commercializing Garbage Collection
18 Jan '94	Efficient Scheduling & Networking
	Beyond LEX & YACC: How to Generate the Whole Compiler
	Kernel Performance
	The Facts About Fax
	System Development Tools
	USENIX Symposium on Mobile and Location-Independent Computing
	7th USENIX System Administration Conference (LISA VII)
19 Jan '94	NFS
	Video Compression - What Do You Do When Everything is Changing?
	Disks and File Systems
	Objecting to Objects
	Dealing With the PC World
	TCL/TK Workshop

Tutorials:

20 Jan '94	Essential UNIX Programming
	Window NT - An Architectural Overview
	Topics in System Administration
	Achieving Security in an Internet Environment

UNIX is a registered trademark of X/Open in the U.S. and other countries.

more-more-more

AUUG INC..SECRETARIAT
PO BOX 366 KENSINGTON NSW 2033
70 GLENMORE ROAD
PADDINGTON NSW 2021 AUSTRALIA
INC. IN VICTORIA ACN A0016636N
PHONE: +61 2 361-5994
FAX: +61 2 332-4066

20 Jan '94	OSF's Distributed Computing Environment (DCE)
	How Networks Work
	Client-Server Development with DCE/RPC
	Porting to Solaris 2.X
	TCL and TK: A New Approach To X11 And GUI Programming
21 Jan '94	UNIX Network Programming
	Windows NT - Developing Client-Server Applications
	Topics in System Administration
	UNIX Power Tools - Getting The Most Out of UNIX
	Distributed Object Computing With COBRA
	The Law and the Internet
	The Kerberos Approach to Network Security
	Chorus and SVR4 UNIX
	Introduction to Threads, POSIX PTHREADS, and OSF/DCE Threads
	Sendmail inside and Out

Fees for Technical sessions, 17 - 19 Jan are $295 for USENIX and AUUG members BEFORE 27 December 1993. After this date add $50.00 (all fees are listed in U.S. dollars).

Fees for Tutorials are: two full day tutorials, $495.00, One full-day , $275. Late fee applies if postmarked after 27 December ($50.00).

For additional Conference or Registration information contact:
USENIX Conference Office
22672 Lambert St., Suite 613
Lake Forest, CA, 92630 U.S.A.

Tel: +1 714 588-8649
Fax: +1 714 588-9706
e-mail: conference@usenix.org

CALL FOR PAPERS - USENIX SUMMER '94

A Call for Papers has been issued by USENIX for the Summer 1994 Technical Conference, 6 - 10 June 1994, Boston Massachusetts.

Dates of Submission:

Submissions Due:	28 January 1994
Authors Notified:	21 February 1994
Camera-ready papers due:	8 April 1994

more-more-more

AUUG INc. SECRETARIAT
PO BOX 366 KENSINGTON NSW 2033
70 GLENMORE ROAD
PADDINGTON NSW 2021 AUSTRALIA
INC. IN VICTORIA ACN A0016636N
PHONE: +61 2 361-5994
FAX: +61 2 332-4066

How To Submit:

It is important that you contact the USENIX association office to receive detailed guidelines for submitting a paper; e-mail to *summer94authors@usenix.org* In addition, enquires about submissions to the USENIX Summer 1994 Conference may be made by e-mail to *margo@usenix.org*

The schedule for reviewing submissions for the conference is very short, so preferred submission is an extended abstract (although we will accept full papers). The object of an extended abstract is to convince the reviewers that a good paper and 25 minute presentation will result. They need to know that authors:

- are attacking a significant problem
- are familiar with the current literature about the problem
- have devised an original or clever solution
- if appropriate, have implemented the solution and characterized its performance
- have drawn appropriate conclusions about what they have learned.

The extended abstract should be 5 manuscript pages (single-sided) or less in length. It should represent the paper in "short form." Include the abstract as it will appear in the final paper. Include references to establish that you are familiar with related work, and if appropriate, performance data to establish that you have a working implementation and measurement tools. If the full paper has been completed, it may be submitted instead of an extended abstract. Full papers should be limited to 12 single-spaced pages. Every submissions should include one additional page containing:

- the name of one of the authors, who will act as the contact for the programme committee.
- contact's surface mail address, daytime and evening telephone numbers, e-mail address, and (if available) fax number
- an indication of which, if any, of the authors are full-time students

Authors of accepted submissions will be notified by 21 February 1994. They will promptly receive instructions for preparing camera-ready copy of an 8 - 12 page final paper, which must be received by 8 April 1994.

Where to Send Submissions:

Submit one copy of an extended abstract or full paper by 28 January 1994 via *at least* two of the following methods:

- e-mail to *summer94papers@usenix.org*
- FAX to +1 510 548-5738
- MAIL to: Summer 1994 USENIX
  USENIX Association
  2560 Ninth St., Suite 215
  Berkeley, CA 94710
  U.S.A.

UniForum NZ '94 - Conference Countdown

"It's a cool autumn evening, you're in the hotel bar quietly sipping a drink and catching up with friends and colleagues. Outside the mist wreaths the trees and there's a strange, almost sulphurous odour in the air. Suddenly the peace is shattered by the eerie call of a Maori Challenge..."

Don't panic, you haven't been transported into a Kiwi version of a fantasy adventure - it's just the start of UniForum New Zealand Incorporated's 1994 conference which runs from Wednesday 18 May to Saturday 21 May 1994. The conference is being held in Rotorua, a place renowned for its thermal activity and Maori culture.

The format of UniForum NZ conferences is very similar to AUUG conferences with only one major difference - the conferences are held in the minor centres around New Zealand rather than the main cities. This enables delegates to get right away from their normal day-to-day distractions and totally focus on the conference. Year after year our conference surveys show that this is one of the most popular aspects of our conferences. Another aspect that is always commented on is how friendly and relaxed the conferences are.

The conference commences on the Wednesday with a day of half and full day tutorials. These cover a wide range of topics from introductory style presentations, both management and technical, to in-depth technical sessions. The tutorials are optional extras and are not included in the main conference registration fee.

The conference programme also covers a wide range of technical and management papers, presented by top overseas and local speakers. There is usually vigorous debate from the floor during UniForum NZ conference sessions. These sessions are broken up into one hour keynote/plenary sessions, forty minute stream sessions and one and a half hour workshop sessions. Birds-of-a-feather sessions run during the latter part of the lunch break and immediately after the main programme finishes in the afternoon. The programme is interspersed with carefully planned social events designed to provide the ideal environment for networking. Details of the programme were not available at the time this article was written but all will be revealed in a subsequent article. A brief description of the conference format follows.

The programme commences on the Wednesday evening with the President's Cocktail Party. This is a casual event, ideal for catching up with old friends from previous conferences or breaking the ice and meeting new people. As many delegates travel to the conference venue during the Wednesday, it's a relaxing way to wind down from the trip with a drink and some substantial munchies.

Thursday usually opens with a keynote session prior to morning tea and then breaks to stream sessions before and after lunch. After afternoon tea we all combine again for a plenary session, followed by the UniForum NZ AGM. The dinner on Thursday is usually a theme buffet, with delegates encouraged to wear something appropriate. This can vary from just a hat (and normal casual clothes) through to a full theme costume, providing great introductory conversation topics.

Friday follows the same pattern as Thursday, but with two plenary sessions at the end of the day. The evening events start with a champagne cocktail hour followed by the formal conference dinner. There is usually some form of after-dinner entertainment, and plenty of time to network.

Saturday starts with stream sessions before morning tea followed by the final keynote session. The conference winds up with the closing lunch, leaving plenty of time for delegates to travel home in the afternoon, if they choose. With venues such as Rotorua many delegates stay over for a day or so to do some sightseeing.

The conference is extremely good value-for-money. The conference registration fee includes the cocktail party, morning and afternoon teas, lunches on all three days, and the two evening social events. The only thing it doesn't include is breakfast, and many delegates just wait for our scrumptious morning teas anyway. The dinners include a selection of top quality wines, often from some of New Zealand's boutique wineries and, therefore not commonly found in many wineshops.

AUUG members may register for the conference at UniForum NZ member rates, NZ$395 (AU$325 approximately, at current exchange rates) if received by 20 April, otherwise NZ$495 (AU$410 approx). The half day tutorial fee is NZ$150 (AU$125 approx) and the full day fee is NZ$250 (AU$205 approx).

We have negotiated a very good tariff for accommodation, $NZ100 per night (AU$85 approx), so the only unknown costs are your travel costs, sightseeing and any personal spending. Move over Club Med! Of course the Australia/New Zealand exchange rate makes the costs even cheaper when calculated in Australian dollars.

We would love to see you at UniForum NZ '94. The New Zealand contingent that attended AUUG '93 had a marvellous time, we'd like to return the hospitality. Registration brochures will be available mid-March. Contact Julie Jones, PO Box 27-149, Mt Roskill, Auckland, New Zealand, phone 64-25-958-245, fax 64-9-629-2015.

AUUG Inc. - Victorian Chapter

(formally 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 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:

Tue, 25 January '94	Social
Wed, 2 March '94	Technical
Thu, 14 April '94	Social
Tue, 24 May '94	Technical
Wed, 6 July '94	Social
Thu, 18 August '94	Technical

Hope we'll see you there!

To find out more about AUUG-Vic and its activities, contact the committee or look for announcements in the newsgroup **aus.auug**, or on the mailing list **sesspoole@clcs.com.au.**

AUUG-Vic Committee *<auugvic-exec@clcs.com.au>*			
President:	Stephen Prince Chancery Lane Computer Services Phone: (03) 608 0911 Email: sp@clcs.com.au	**Secretary:**	Neil Murray Webster Computer Corporation Phone: (03) 560 1100 Email: neil@wcc.oz.au
Treasurer:	John Carey Labtam Australia Phone: (03) 587 1444 Email: john@labtam.oz.au	**Programme Chair:**	Michael Paddon Iconix Phone: (03) 571 4244 Email: mwp@iconix.oz.au

Update on AUUG Inc. - Victorian Chapter Activities

by Stephen Prince
President, AUUG-Vic.
<sp@clcs.com.au>

It's been a busy time for AUUG-Vic committee since my last column in v14n5. Three events seem to be dominating business, which I'll describe briefly.

Regular Meetings

The technical meetings continue to be a success. The third meeting for the year was held on 12th October at the *MTIA*. Thank you to *Stephen Boucher* for organising the venue. An attendance of about 25 people heard a very interesting talk from *Enno Davids* on the challenges in GPS based vehicle tracking systems. This presentation was longer than previous talks, due to large amount of ground covered. since Enno covered everything from the basic background, through to explaining how GPS works and then methods for improving the results from such systems. *Michael Gigante* then gave a short "work in progress" update (complete with video) of his virtual reality research efforts at the Advanced Computer Graphics Centre, CITRI.

As promised in previous columns, the fourth and final technical meeting for the year was held on 20th November at the Wool Exchange Hotel in Geelong, for the benefit of our country members. Many thanks to *Michael Paddon* and *Craig Bishop* for organising this one.

The afternoon started with a talk by *Warren Stokes* from the Geelong & District Water Board, about his experiences with the Kalpana Etherswitch and how they used it to link together multiple servers and multiple workstations. *Enno Davids* then presented the second half of his GPS talk. The first half was presented at the third technical meeting. Enno concentrated on the challenge of vehicle tracking using GPS data. The day finished off with almost everyone adjourning to Cafe Dodo for dinner and a social chat. The general consenus, is that the away meeting has been well worth the effort. A great day was had by all.

The next meeting (on December 16) will be a social night, to enable members to have christmas drink with one another.

Kirk McKusick Workshop

In my last column, I mentioned we were planning a three day workshop on BSD UNIX† internals. This has now been confirmed, with the course being structured as:

3+4 February 1994	Introduction
5 February 1994	Advanced Course

I hope to have a registration brochure for this event available shortly. If anyone would like a copy sent to them when it's ready, please let me know.

Summer94 (Vic) Conference

Things have certainly been busy in this area; we nearly didn't have an organiser at one point. Anyway, we now have the dates and venue confirmed, as well as a volunteer to organise a tutorial programme:

15 March 1994	Tutorials
16 March 1994	Summer Conference

The venue will once again be *Clunies Ross House*, the same place as last year. The respective contacts for the two events:

Tutorials: Robert Sturrock
`rns@deakin.edu.au`
+61-052-27-2108
Computing and Communications Services
Deakin University
Geelong VIC 3217

Conference: Arnold Pears
`pears@latcs1.lat.oz.au`
+61-3-479-1144
Computer Science Department
La Trobe University
Bundoora VIC 3083

So if you would like to run a tutorial or present a paper, please contact either of these two people directly.

Other News

Well, it's getting close to that time of year again when elections will be held. So now is the time to start thinking seriously about nominating for a position on the committee for next year. The elections will be held in conjunction with the summer conference. All of the current committee will **not** be renominating again. As all feel that over four years on the committee is enough, and since AUUG-Vic is up and running, most feel they need to move on.

If you have any thoughts, ideas, comments on any matters, please feel free to contact the committee, preferably via email: *auugvic-exec@clcs.com.au*.

† UNIX is a trademark of Bell Laboratories.

From the Western Front

The other day I found out via the net that my favourite band's lyricist uses a Powerbook to write his lyrics. Sigh. :-)

But more seriously, these days I'm managing Macs as well as Unix systems. Navigating the Mac user interface reminds me what it is I like so much about the Unix computing environment — the philosophy of building functionality from small tools, and letting the computer do the hard work.

It strikes me that the "religious" arguments between Mac and Unix aficionados stem from a disagreement as to what constitutes hard work. The Mac-happy (and many GUI marketing people) think the hard work is in learning and remembering commands, so the GUI interface does that part for you. To the Unix lover, the hard work is in repetitive tasks, so the system, especially through the shell, provides powerful facilities for automating them. No prizes for guessing which activity I call hard work.

GUIs certainly have their uses. On my X screen right now, I have `xclock`, `xcalendar`, `xpostit`, xmh, and three shell windows. The calendar is an interesting example, because I use the graphical interface for day-to-day scheduling, but have a cron job that runs on the 25th of each month to update the calendar's ASCII data files for the next month with my regular appointments, like the WAUG meeting on the third Wednesday. I set up the cron job because the amount of automation I want is beyond the scope of the regular-appointment facilities of any calendar program I've seen, and because I can extend it or make it work with a different calendar program (or even simply email me details of appointments, if I should prefer it that way).

The problem I have with the Mac environment is that this kind of general-purpose automation is out of reach of the average user. One can't even do a simple loop, for example to append "Backup" to the name of each of ten files. I cringe internally when I see one of our researchers perform the same tedious action ten times.

I cringe even more when the person is using a Unix system, via a graphical file manager, when the labour-saving power of the shell is only a mouse movement away!

Well, that's my little rant out of the way; onto news of AUUG's Perth activities. We've had a good bunch of speakers lately, and attendances are on the increase. In this AUUGN we have a previously-lost review of the August meeting, a second review of the particularly excellent September meeting, and a review of the October meeting, which ended up being an informal, enthusiastic Q&A session on 386BSD. Julian Elischer, author of the 386BSD SCSI system, was our one-man panel.

Like October's, the November presentation included several demonstration systems. It's always fun to have something to look at and play with, although I think it sometimes makes for a superficial coverage of the topic.

November's meeting was held at IBM's Perth office. Our friendly local IBM Unix people, Indulis Bernstein and Cameron Ferstat, described the PowerPC architecture and exhibited demos of that, the MS-Windows emulator WABI, and a neat-looking Mac emulator. Such emulators will be useful for poor souls whose departmental newsletter editors insist on MS-Word input. :-(

IBM have abandoned their "dry office" policy and provided beer and wine, as well as an excellent spread of sandwiches, fruit and other nibbles.

The Summer Conference, which WAUG can at last be *officially* involved in, is looming up. Adrian Booth is organising it. What with Kirk McKusick and, hopefully, a good bunch of local speakers, it should be a worthwhile day away from the office.

Janet Jackson <jackson@cwr.uwa.edu.au>
From WAUG, the WA Chapter of AUUG

FYI: WAUG's postal address is PO Box 877, WEST PERTH WA 6005.
Email addresses: waug@uniwa.uwa.edu.au, waug-meetings@uniwa.uwa.edu.au, waug-newsletter@uniwa.uwa.edu.au.

WAUG Meeting Reviews

August
Unix in Real Time Systems

David Basell, Industrial Micro Products Pty Ltd

I must admit to some trepidation when David started his talk — I always get nervous when someone starts a talk with a description of their company, and then reassures the audience that despite the way it might come across, the talk isn't supposed to sound like an advertisement.

In this case, the talk wasn't advertising the PSOS real-time integrated kernel that Industrial Micro Products sell, and use to develop industrial engineering systems. PSOS was just being used as an example of a real-time system.

David mentioned that there is little consensus on exactly what constitutes a real-time system, but mentioned that the sequence and duration of external event are of prime importance in such a system. Features that must be provided by a real-time system are:

Determinism: the behaviour of the system must be entirely predictable;

Throughput: the code on each critical path must be optimal;

Intensive I/O: I/O is fundamental to real-time systems, and mustn't be an incidental add-on;

Reliability.

Traditional ways of structuring real-time systems include one monolithic single-threaded application, foreground/background processes, and true multitasking. Multitasking has the advantages that it facilitates a cleaner design, makes maintenance easier, is deterministic and responsive, and suitable even for complex systems.

David showed an example of a multiprocessor system running the PSOS kernel on one processor, and UNIX on another. The UNIX is not real-time, but has interfaces to the real-time layer. This offers great programmer convenience — the non-real-time aspects of an application can be developed in a familiar UNIX environment that hasn't been hacked to support real-time.

I enjoyed David's talk for its technical content, although slightly more breadth about other real-time systems might have rounded it out better.

September
Getting Better Value out of UNIX I/O

Jim Longwood, Senior Systems Engineer, Fujitsu Australia

This was one of the better talks we have had recently, and one of the better attended also. Jim told us that his talk would concentrate on I/O issues as he claims that 70% of interactive response time is consumed by I/O (which means that adding CPU will often not significantly improve matters).

On average, the time it takes for random I/O to complete is 64% composed of *access* time (the time it takes to move the disk head(s) to the correct cylinder), 35% composed of *latency* (the time that elapses while the disk controller waits for the disk to rotate the desired block under the disk head), and around 1% *transfer* time from the disk.

Hence a simple (and low-cost) optimisation of random disk I/O is simply to place commonly accessed files in the middle of the disk, and to place less frequently accessed files at either extreme. The easiest way to achieve this is to split each disk into several partitions, placing the heavily accessed files into the central partition(s).

How can you determine how your disks are performing? The main tools in Berkeley-derived versions of UNIX are `vmstat` (for macro data) and `iostat` (for device data). Under System V systems, the tools are `sar -u` and `sar -d`, respectively. You can also use tools supplied with your RDBMS.

Jim suggested that a disk utilisation greater than 25% (as reported by `iostat -D`) is symptomatic of a performance problem. (I have seen claims that disk utilisations of up to a sustained 50% are reasonable, but this obviously depends upon how much money you are willing to spend on performance).

Other data that may point out a possible disk performance problem is the `msps` column in the output of `iostat tds`. This shows the average number of milliseconds that elapsed per disk I/O request — you can compare this figure to the average seek times quoted by your disk manufacturer to get a feel for how well your drives are performing.

If you feel you may have an I/O bottleneck, things to investigate are high paging rates and/or high CPU usage, I/O response times greater than those expected (see the previous point), I/O caching levels, device utilisation levels, and symptoms of disk fragmentation (such as erratic I/O response times).

If the whole disk is busy, and you can't simply scatter files across it to localise I/O, you are faced with two problems — continual movement of the disk heads, and I/O request queueing at the controller. Solutions fall into two types — splitting the files across more than one disk, or striping the data across several disks.

You can achieve higher reliability *and* performance by both mirroring and striping your data — "Software RAID".

Specific tips for tuning RDBMS I/O include tuning the application, tuning shared memory (by caching database and dictionary buffers and avoiding bottlenecks within the RDBMS), using UNIX raw devices for I/O, placing database tables on different disks and controllers to the indexes, and tuning log files (the size of redo logs and the segmentation of rollback files).

I thought Jim's talk was excellent — technical enough to be satisfying, but well-rounded by its pragmatism. Why don't *you* give a similar talk in the future?

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

From WAUG, the WA Chapter of AUUG

October
386BSD Operating System

Julian Elischer, member of the 386BSD development group

Be warned — do not take notes within sight of the Hon. Secretary of WAUG! You will end up later like me, desperately trying to remember why you agreed to write the review and just what all those crumpled chicken scratchings on your scribble pad really meant! However, despite my grumbling now it was well worth spending an hour listening to Julian Elischer last month. Julian's interesting and easy-to-follow introduction to 386BSD (or FreeBSD) served to whet the appetite for installing it ourselves.

During the talk Glenn did his best to stimulate discussion (and interruptions for Julian) by offering freebies from the recent AUUG conference such as T-shirts and conference packs. I don't know that the quality of the questions improved but it did prompt a few quiet people to speak up.

Apart from all the information circulated before the meeting regarding supported devices, Julian talked about support for video cards via the X11 server. X386V1.3 (available for about 3 months now) supports many cards already and X386V2.0 with support for more cards is coming out "real soon".

For those of us with less cash, the good news was that our old 386SX with 4Mb can run FreeBSD — slowly, but it will run. It will even support those old 5Mb ST502 drives we have littering up the place at home. For SCSI devices, Julian's generic SCSI device driver will support just about any kind of device available. I have it on good authority that said SCSI device driver is acknowledged by the Unix community to be a classic in design for ultimate portability as it separates those segments of code unique to the CPU from all device-specific code. Only minor changes are required to port this driver to a new processor (minor to folk who write device drivers anyway!).

Also available is a public domain utility, `booteasy`, which displays a boot menu to prompt you to boot from Windows or 386BSD. Of course this assumes you have first managed to configure your system so that DOS and 386BSD can live happily together. Julian spent part of his talk on tips to help with this aspect of the installatiốn; swap space (generally double the real memory) and a reminder to try really hard to pare the kernel down to around 500k or risk some inconvenience, and loss of the first 640k of RAM, in loading a larger kernel.

The evening concluded with a demonstration installation for those interested and a reminder that copies of FreeBSD not requested before the talk will be a mite slow in arriving since Julian has been unexpectedly dragged back to the US for 3 months. So if you did not ask for your copy before the talk then you will need to wait a while.

Maxine Croft <maxine@chi.uwa.edu.au>
From WAUG, the WA Chapter of AUUG

Open System Publications

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

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

For example:

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

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

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

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

ACSnet Survey

1.1 Introduction

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

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

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

Please send replies to:

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

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

Technical enquiries to:

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

Thank you

======

1.2 Contact Details

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

1.3 Site Details

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

Host Name: ____________________

New Connections

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

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

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

A14. Additional Comments:

Existing Sites

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

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

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

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

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

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

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

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

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

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

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

B10. Additional Comments:

Book Reviews

Here are this months collection of book reviews. Instead of the ones from our usual Publishers, we have a selection from X/Open. With all the recent changes to the ownership of Unix and the Unix trademark it is appropriate to see what sort of standards X/Open specify. All the X/Open standards are available from:

X/Open Publications
PO Box 475
Ringwood 3134

Phone: (03) 879 7412
Fax: (03) 879 7570

We also have a second review of *Learning the Korn Shell*, a Nutshell Handbook. Occasionally we receive two copies of books and this is such a case.

Finally, lately we have had lots of books to review. Keep an eye on the newsgroup *aus.org.auug* for details.

Frank Crawford

X/OPEN COMPANY LIMITED (January, 1992): CAE Specification X/Open Transport Interface (XTI)

by X/Open Company Ltd,
200pp., (Soft Cover)
ISBN 1-872630-29-4

Reviewed by
Keith Lugsden
LORIL Pty Ltd
<keithl@research.otc.com.au>

X/Open is widely respected within the Unix open systems community for trying to bring some order to the implementation of open systems. It is producing a comprehensive environment based on present and developing standards, called the Common Applications Environment (CAE). With NCR proposing to assign its Unix copyrights to X/Open, the organisation and its standards are likely to achieve even more relevance to open systems implementors and users.

The X/Open Transport Interface (XTI) is concerned mainly with the ISO Transport Service Definition, but also includes the more familiar Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). The XTI, as presented in the specification, is implemented as a set of C library functions and is designed to be Unix-variant independent.

The specification provides six chapters and eight appendices. The first five chapters provide a well-written background to the XTI and guidelines on how to use it. For those of us who like concise, unambiguous and informative specifications, Chapter 6 is excellent. It lists each of the C library functions with a synopsis, description, list of valid states, possible errors, return values and cross-references to related functions. There are in all 27 functions described. To remove some of the confusion often involved in such situations, Appendix F provides a full listing of xti.h, the include file required by user and kernel developers alike to access any of the C functions.

For those of us who are used to standard network functions such as accept(), bind(), connect(), listen(), etc, it is reassuring to see that the new equivalents all have a similar name, usually just with a t_ prefix, as in t_bind(). Their usage is also similar.

This specification will be an essential reference for anyone intending to implement or use the XTI, or who simply has an interest in such things. I will certainly be keeping a copy amongst my reference texts. It is not a general tutorial on the transport layer and should not be seen as such.

X/Open Guide
Distributed Transaction Processing
Reference Model

by X/Open Company Limited.
21pp., (Soft Cover)
ISBN 1-872630-16-2
X/Open Document number: XO/GUIDE/91/020

Reviewed by
Michael Fazzolare
University of Wollongong
<faz@cs.uow.edu.au>

This is an extremely short "document". The document is a high level description of the Distributed Transactional Processing (DTP) model as defined within the X/Open Common Applications Environment (CAE). An application that is complaint with the X/Open CAE is assured "portability", "interchangeability" and "interoperability" with other X/Open applications.

This is not "a reference for", but rather a "description of" the X/Open DTP model. It is aimed at application developers, system administrators, resource manger providers and transaction manager providers who need a working understanding of the X/Open DTP model. As such, this document is requisite reading for such people. The document is meant to be read in association with other relevant X/Open documents (itemised in the document).

> *Potential readers should have a working knowledge of transactional systems in order to understand these models. A suitable introductory book such as Gray and Reuter's Transaction Processing: Concepts and Techniques [1] is, in the reviewer's opinion, pre-requisite reading. The document achieves its limited purpose as an introduction to the maze of X/Open acronyms, and illustrates the structure that X/Open compliant applications should possess.*

The document presents the functional, process and communications models of the X/Open DTP model. The functional model describes the inter-relationship of the functional components of the model. These are the application program (AP), the transaction manager (TM) and the resource managers (RMs). The process model discusses the acceptable, as well as un-acceptable ways that the AP, TM and RMs can inter-operate. Furthermore, groups of processes called domains are introduced. Such groupings are introduced to allow administrative control for DTP applications.

The communication model introduces the suggested forms of inter-domain and intra-domain communication. Basically, inter-domain communication is encouraged through special communication RMs. Communication within a domain is via shared memory, streams and DTP communication protocols (not specified). X/Open is still defining portable application interfaces for the DTP communication model. Nested transactions are supported in the communications model.

Reference.

[1] Gray J., and Reuter R., "*Transaction Processing: Concepts and Techniques*", Morgan Kaufmann, USA, 1993, ISBN 1-55860-190-2.

Preliminary Specification of:
Distributed Transaction Processing:
The TX (Transaction Demarcation) Specification

by X/Open Company Limited.
69 pp., (Soft Cover)
ISBN 1-872630-65-0
X/Open Document number: P209

Reviewed by
Michael Fazzolare
University of Wollongong
<faz@cs.uow.edu.au>

This document is a a preliminary specification of the Distributed Transactional Processing (DTP) model's transaction demarcation (TX) interface as defined within the X/Open Common Applications Environment (CAE). A preliminary specification is a "trial use" standard. This means that it is as stable as X/Open can make at present, but it is not supported by a base of conformant product implementations. Any changes to the specification will be (if possible) made in an upward-compatible manner. The TX interface is the interface between an application program (AP) and a Transaction Manager (TM). An application that is complaint with the X/Open CAE is assured "portability", "interchangeability" and "interoperability" with other X/Open applications.

This document is aimed at application developers, who wish to write portable programs that use ''global'' transactions. Global transactions are best described as distributed nested transactions (reviewer's opinion). Naturally, the document is co-requisite reading for such people. The document is meant to be read in association with other relevant X/Open documents (itemised in the document). The document achieves its limited purpose of specifying the TX interface.

The document firstly positions the TX in the broader scope of the X/Open DTP model. This is followed by an overview of the TX interface. The overview is terse and no justification of the introduced features is presented. Having said that, the model itself seem to be quite general and useful.

The detailed description is presented as a C language header file followed by Unix manual page, C library style descriptions for each routine in the TX interface. This is followed by Unix manual page, Cobol library style descriptions of the routines. Nothing equivalent to a header file for Cobol is presented. This appears to be an oversight. Furthermore, no indication of why the particular combination of C and Cobol were chosen to present the TX interface is given.

On the positive side there is a useful state transition table that shows the expected calling sequences for routines in the TX interface. This overcomes one of the problems with the Unix manual page method of presenting information. Namely, TX interface users can determine from the document, not only what the routines do, but also in what order they should be called.

As there is no base of TX interface conformant applications, the authors might have presented an example demonstrating a hypothetical TX interface conformant application. This would show potential application developers the steps required to implement an such applications without have to wade through all the interface descriptions and the state transition table.

Systems Management: Management Protocols API (XMP)

by X/Open Company Ltd.
204pp., (Soft Cover)
ISBN 1-872630-32-4

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

The thought of curling up on a cold, rainy evening with a copy of a preliminary X/Open API specification is not the thing of which dreams are made. This books isn't exactly what one would call "a good read" unless you are of the habit of leaving copies of RFC's by your bed for a bit of light reading. That said, the material covered will be of interest to those of us who work closely with network and systems management.

The X/Open Management Protocols API (XMP) is a specification for providing a common API for accessing management information. The framework caters for the two most popular and deployed management protocols: SNMP and CMIP. Although there tends to be a bias towards the OSI way of doing things throughout the specification (and also in the terminology used by the book), the Internet management framework is covered well.

The style adopted by the book is the usual, dry, standards-speak that we have come to expect from a specification of this kind. It presents the material covered in a well organised, three part approach including 40 pages of textual description, 110 pages of function and interface reference, an the remainder of the book dedicated to example code, possible error conditions and a good sized glossary of the terminology gleaned from OSI, other X/Open specifications and the management community in general.

While reading through the spec, I started wondering if this API was just another attempt by a standards body to formalise an emerging area of our field and how it was being viewed by the vendors and researchers alike. The API covers key areas such as management services, session maintenance, synchronous and asynch management approaches, and management contexts. It does not, however, attempt to deal with the question of security within management frameworks. After a little prodding around on the net, it appears that the API has received support from both the OSF and HP. Apparently, the OpenView Distributed Management Developers Kit offers an XMP interface to the CMIP and SNMP implementations available within that environment. The much publicised Distributed Management Environment from OSF will also offer an XMP compliant interface to the developer.

Seeing as there is a level of support from two of the major players in the management game, this API may become the programmatic standard for management applications. As such, this book should be of interest to anyone wishing to utilise the API through the HP product or just wanting to keep in touch with developments in the management world.

Learning the Korn Shell

by Bill Rosenblatt
O'Reilly & Associates, Inc.
June 1993, 338 pages,
ISBN 1-56592-054-6

Reviewed by
Gordon Turner
<gordont@dialix.oz.au>

Yet another O'Reilly book, do we really need it? My answer is yes. Having worked with several shell's, including the standard Bourne shell, the Berkeley csh, and public domain offerings like tcsh and bash, reviewing the book would answer the question, what makes the Korn shell different, and why would one want to use it?

The Korn shell was developed by David Korn of AT&T Bell Laboratories in the mid 1980's. The Korn shell was distributed as part of the "Experimental Toolchest" in 1986. Eventually it was distributed in SVR4. The version described in the book is from 1988. Typically most vendors versions are from this release. The Korn shell is obtainable from USL, at a cost, there is a public domain version available, which lacks some of the features but is otherwise a functional Korn shell.

The first few chapters and it was clear that the Korn shell could replace all the other efforts. With command line editing (in vi or emacs style), aliases, history, job-control, csh syntax, Bourne shell almost compete compatibility†, what more could you want. A scripting language of course, and with Korn shell one could almost do away with perl :-)

Details of the enriched scripting capabilities are described in good detail. Including the basics of a shell script debugger. All this could make the book worthwhile. Moving to SVR4 with Korn shell will certainly make life easier.

The book as are most of the O'Reilly series, is set out in an excellent manner. Each chapter builds on the knowledge from the previous. For those experienced at using one or more of the Unix shells interactive or in scripting it is easy to jump into any chapter and learn how the Korn shell can be used.

The book also represents a good learning tool on using the Unix shell, script programming, and a number of other concepts such as writing a compiler. Particularly useful to a computer literate person coming to unix for the first time. Personally I rate the book at 8 out of 10.

† For the one non-compatibility read Appendix A pg. 292 (if you cannot get your hand's on a copy then E-mail me)

AUUG BOOK CLUB
&
PRENTICE HALL AUSTRALIA

20% DISCOUNT TO AUUG MEMBERS

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

New from Douglas Comer!

□ ***Internetworking with TCP/IP Vol.3 - BSD Socket***
ISBN: 0134742222 , Cloth, 1993 , RRP $49.95

□ ***Unix Software Development Tools SVR4.2***
ISBN: 0130176907, Paper, 1993, RRP $65.95

□ ***Internet: Mailing Lists***
ISBN: 0133279413, Paper, 1993, RRP $49.95

□ ***Networking Applications on Unix System V Release 4***
ISBN: 0136135552, Cloth, 1993, RRP $59.95

*Deduct 20% from listed retail price

Name:____________________ Organisation:________________
Address:____________________________________
(Street address only)
________________________ Telephone:______________

□ Please send my book/s on 30-day approval *(tick box)*

Enclosed cheque for $ __________ (Payable to 'Prentice Hall Australia')
Please charge my: □ Bankcard □ Visa □ MasterCard
Credit Card No: □□□□ □□□□ □□□□ □□□□

Expiry Date: __________ Signature:____________________

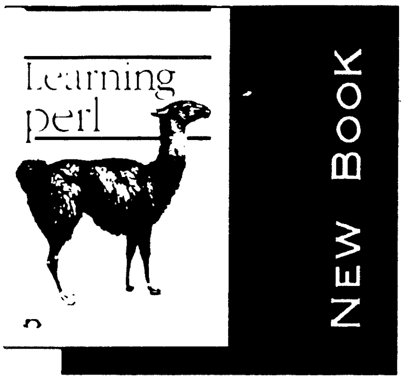
Mail or fax completed order form to Prentice Hall Australia, PO Box 151, Brookvale NSW 2100

OR ☎ **Use our FAST PHONE SERVICE by calling Liz Guthrie.**
SYDNEY (02) 939 1333

A.C.N. 000 383 406

Prentice Hall Pty. Ltd.
7 Grosvenor Place, Brookvale NSW 2100.
Tel: (02) 939 1333 Fax: (02) 905 7934
A Paramount Communications Company

O'REILLY & ASSOCIATES, INC.



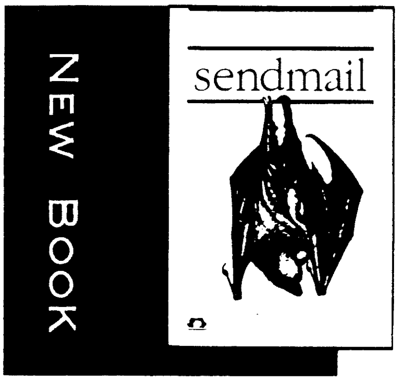
LEARNING PERL

AUTHOR: RANDAL SCHWARTZ

Perl is rapidly becoming the "universal scripting language." Learning Perl is a step-by-step hands-on tutorial designed to get you writing useful Perl scripts as quickly as possible. In addition to countless code examples, there are numerous programming exercises, with full answers.

PRICE: $49.95

ISBN: 1-56592-042-2



SENDMAIL

AUTHOR:COSTALES ET AL

This new Nutshell Handbook is far and away the most comprehensive book ever written on sendmail, a program that acts like a traffic cop in routing and delivering mail on UNIX-based networks. It provides a complete tutorial, plus extensive reference material, and covers both major versions available on most systems.

PRICE: $65.00

ISBN: 1-56592-056-2

AVAILABLE FROM AUSTRALIA'S LEADING TECHNICAL BOOKSELLERS

☛ NSW:	**Dymocks**	George St Sydney	224 0411
	Dymocks Pocket & Technical	Wynard Sydney	224 0444
	Hotline	Chatswood Sydney	415 2265
	Dymocks	North Sydney	956 6771
☛ VIC:	**Tech Books**	Melbourne	(03) 663 3951
	McGills	Melbourne	(03) 602 5566
	Dymocks	Melbourne	(03) 663 1344
☛ QLD:	**Dymocks**	Edward St Brisbane	(07) 220 0152
☛ SA:	**The Bookshelf**	Adelaide	(08) 223 3899
	Dymocks	Adelaide	(08) 223 5381
☛ WA:	**Rellim Booksellers**	Perth	(09) 321 5684
	Boffin Bookshop	Perth	(09) 321 5755
☛ A.C.T.	**Uni Co-Op Bkshp**	ANU Canberra	(06) 257 2673

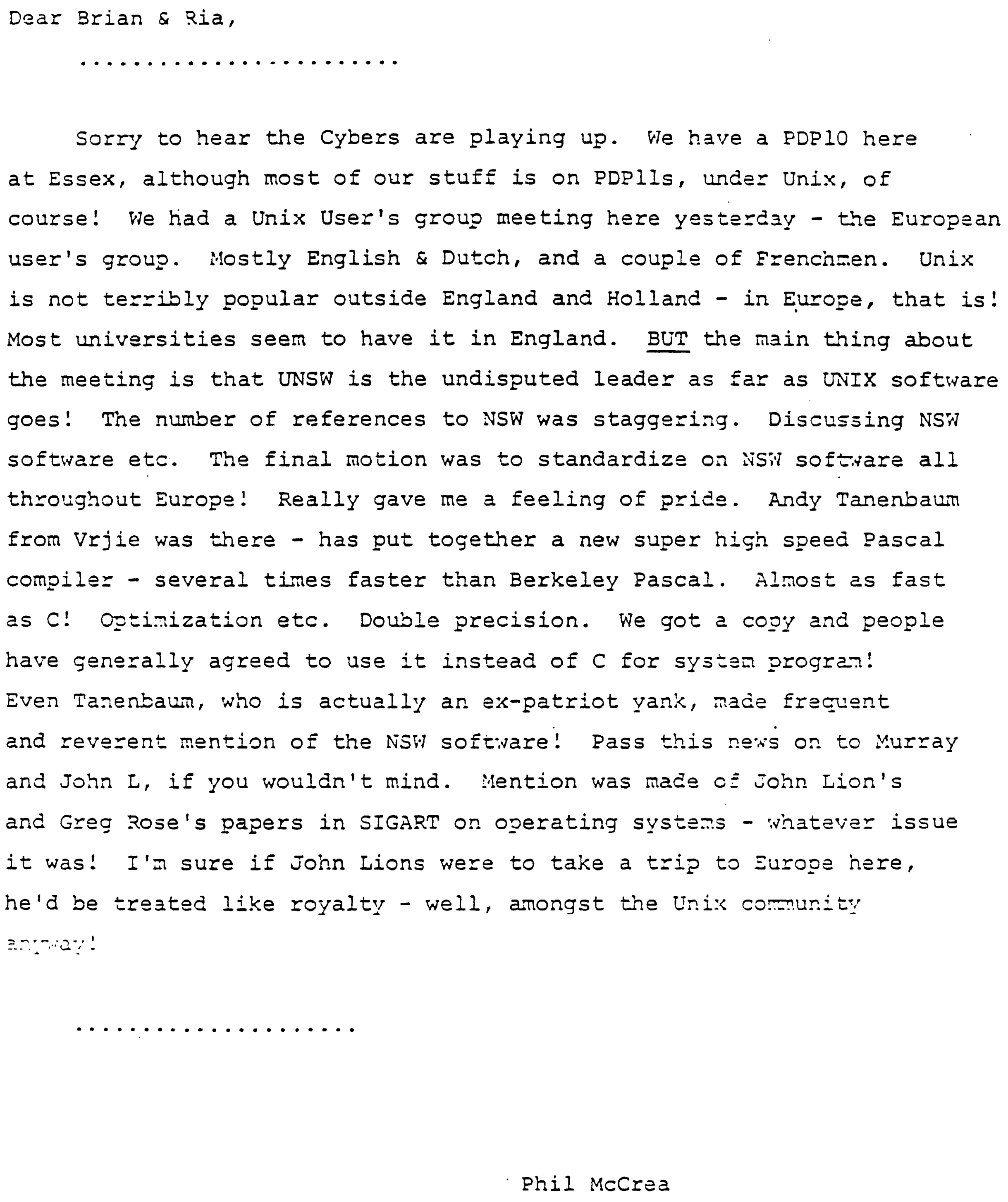
Or call WoodsLane for the stockist nearest you.

Distributed by **WoodsLane Pty Ltd** A.C.N. 003 667 549

Phone: (02) 979 5944 Fax: (02) 997 3348 EMail: woods@tmx.mhs.oz.au

!AUUGN

The following letter is reprinted from AUUGN Volume 1 Number 1. As this letter from Phil McCrea shows, Australia has a long history of work in Unix, and our president has a long connection with AUUG.



25.9.78

Dear Brian & Ria,

....

Sorry to hear the Cybers are playing up. We have a PDP10 here at Essex, although most of our stuff is on PDP11s, under Unix, of course! We had a Unix User's group meeting here yesterday - the European user's group. Mostly English & Dutch, and a couple of Frenchmen. Unix is not terribly popular outside England and Holland - in Europe, that is! Most universities seem to have it in England. BUT the main thing about the meeting is that UNSW is the undisputed leader as far as UNIX software goes! The number of references to NSW was staggering. Discussing NSW software etc. The final motion was to standardize on NSW software all throughout Europe! Really gave me a feeling of pride. Andy Tanenbaum from Vrjie was there - has put together a new super high speed Pascal compiler - several times faster than Berkeley Pascal. Almost as fast as C! Optimization etc. Double precision. We got a copy and people have generally agreed to use it instead of C for system program! Even Tanenbaum, who is actually an ex-patriot yank, made frequent and reverent mention of the NSW software! Pass this news on to Murray and John L, if you wouldn't mind. Mention was made of John Lion's and Greg Rose's papers in SIGART on operating systems - whatever issue it was! I'm sure if John Lions were to take a trip to Europe here, he'd be treated like royalty - well, amongst the Unix community anyway!

....

Phil McCrea

LINUX — An Open system Opened

Peter Chubb
Softway Pty Ltd

February 12, 1993

Abstract

The question to ask with all the hype about open systems is, "Open to whom?" Most proprietary "open" systems are frozen to input from ordinary people.

LINUX is a POSIX conformant operating system that runs on 386 and 486 machines. It has almost all the features one could ask for in an *open* system: comes with full source code. free (in both the GPL and monetary senses), wide support via the Internet, many utilities and applications; it has multiple filesystems (including /proc and MS-DOS filesystems), sockets and TCP/IP, SCSI disc, tape and CD-ROM support, virtual memory. a MS-DOS emulator, etc., etc., etc.

LINUXhas been developed cooperatively by many people world-wide. It isn't too late for *your* contribution to make it into the system — you can't get more open than that!

1 Introduction

There has been a lot of hype about open systems. People have given many definitions. including things like:

- An Open System is one that conforms to *de facto* or *de jure* standards that have been created with input from the wider community.
- An Open System is one that has been developed according to an Open Process.
- An Open System is one that can communicate using the OSI communications protocol stack
- An Open system is one that is available from more than one vendor.
- An Open System is one that comes with source.
- Whatever an Open System is, 'our system' is (for 'our system', replace any vendor's system, depending on who one talks to)

It's clear that today, when used by computer vendors, 'Open System' means UNIX or UNIX-lookalike — except when used by a salesman from Microsoft, when it means NT. However. the idea of an 'Open System' obviously involves the idea that more than one person or group of people can be involved in its creation; that it is not restricted to a particular vendor's hardware, and that it can communicate with other systems. Also, the idea that ordinary people can have input into the system.

Now, UNIX once upon a time was an open system by all these definitions. Source licenses were relatively cheap; people working more-or-less on their own could provide patches to the system; and so on. Now, however, one needs a small fortune and a lot of free time to do anything off one's own bat that will make it into 'official' UNIX.

2 UNIX lookalikes

In recent times, with BSD386 and LINUX, Open Systems live again. People can start hacking kernels with a fair chance that anything worthwhile might make it into the 'official' releases.

3 LINUX

In early 1991, Linus Torvalds decided to create a UNIX-like operating system that, unlike MINIX, provided some of the advanced features of a modern UNIX (such as paged virtual memory). To do this, he started with an Intel 386 box running MINIX (as an aside, the most stable of the file systems available under LINUX is the MINIX filesystem).

He made this available to the world in November 1991, as LINUXversion 0.10. Since then there have been one to four updates a month, as Linus has been both writing code himself, and integrating code from other people. Because the source has always been available, people all around the world have been trying to get it to work on *their* machine - which has meant that drivers for SCSI discs, ethernet cards, and VGA cards have rapidly become available for most common varieties. Also they have been busy adding the features they most want — TCP/IP, better file systems, user level code, etc.

Currently, LINUX has the following features:

- File system switch, with Xenix, Amigados, Extended, Minix, /proc and two or three experimental filesystems.
- TCP/IP
- NFS — a kernel client, and a user-level server.
- Virtual memory, swapping either to a dedicated partition or to a file, and paging text directly from the program image.
- Shared libraries, with jumptables so that libraries can be upgraded without recompiling all the programs that use them.
- Copy-on-write fork.
- Gnu utilities, including GCC 2.3.3.
- X11r5 (it's the XFree 1.2 release)
- TEXand **groff** typesetting systems.
- University Ingres database system
- etc., etc., etc.

4 Architecture

The kernel falls quite neatly into a number of parts. These are:

- The file systems
- The memory system
- The network code
- The rest.

The directory tree looks like:

```
$ ls -F
COPYING       Makefile.old  boot/         init/         mm/
Configure     README        config.in     kernel/       net/
Image         System.map    fs/           lib/          tools/
Makefile      TAGS          include/      makever.sh*
$ ls -F fs
Makefile       fcntl.c        ioctl.c        namei.c        read_write.c
block_dev.c    fifo.c         isofs/         nfs/           select.c
buffer.c       file_table.c   locks.c        open.c         stat.c
exec.c         filesystems.c  minix/         pipe.c         super.c
ext/           inode.c        msdos/         proc/
$ ls -F net
Makefile      socket.c      tcp/
kern_sock.h   socketcall.h  unix.c
```

At the top level are the directories boot, fs, include, init, kernel, lib, mm, net and tools. There are also the files **Makefile** (to make the kernel), and Configure, a shell script that configures the **Makefile**. Other files are either generated by Configure (.depend. .config) or are documentation files (COPYING, which contains the Gnu Public License, README which is almost always out of date).

4.1 boot

The **boot** subdirectory contains code used while booting LINUX, including the boot sector itself. The boot sector is loaded at 0x7c00 by the boot loader. It then relocates itself to 0x90000 and uses BIOS routines to load the code in **setup**.S at 0x90200, and the LINUX kernel at 0x10000. **Setup**.S then gets information about the system from the BIOS and overwrites the boot-block with it. This information includes the kind of discs attached, the kind of VGA card available, the amount of memory, etc. It proceeds to set up the interrupt control system, and the initial memory map for kernel mode: it maps the virtual addresses so that the loaded kernel is at location 0, then branches to it.

The kernel starts with the code in **head**.S, which does yet more setup (determining whether the CPU is 386 or 486, determining whether the kernel needs to emulate a FP coprocessor, etc), and eventually invokes **main()**.

4.2 kernel

The **kernel** directory has subdirectories containing the block and character device drivers.

4.3 fs

Within the **fs** directory are subdirectories for each of the supported filesystems, and files containing glue for all the filesystems.

Filesystem-related system calls come into the generic code, and are passed off through a filesystem switch. Each part of the filesystem (superblock, inodes, open files) has a pointer to an array of function pointers for the various operations (see figure). Rather than each filesystem declaring an inode structure for itself, that is kept as part of a vnode (as is done in BSD and SVr4), data is translated to a canonical form when it is read from the disc, and translated back again when written. This leads to less code duplication than the alternate approach, but does add some runtime overhead. As you can see from this, the kernel is designed to be compiled with an Ansi C compiler (gcc 2.3.3, to be precise).

The most interesting filesystem is **proc**. This is a plan-9-style /proc filesystem, which looks like this:

```
$ ls -F /proc
1/       28/      4548/    53/      5576/    5728/    loadavg  version
1972/    3/       4549/    54/      5577/    5761/    meminfo
1973/    43/      4550/    55/      56/      5911/    self/
2195/    45/      4560/    5575/    5727/    kmsg     uptime
```

The directories that have numeric names correspond to processes; other 'files' give information about the kernel. For example, if **/proc/uptime** is catted

```
$ cat /proc/uptime
612891
```

it yields the seconds since last boot; if **/proc/meminfo** is examined, the display produced is:

```
$ cat /proc/meminfo
        total:    used:    free:   shared:  buffers:
Mem:  15785984 12840960  2945024  3080192  6291456
Swap: 16773120   262144 16510976
```

The directory **self** refers to the process that opens it.

Within each process directory are subdirectories and magic files:

```
$ ls -l /proc/1
total 3
-r--r--r--   1 root     root            0 Feb  9 15:44 cmdline
lrwx------   1 root     root           64 Feb  9 15:44 cwd -> [0802]:1
-r--r--r--   1 root     root            0 Feb  9 15:44 environ
lrwx------   1 root     root           64 Feb  9 15:44 exe -> [0802]:375
dr-x------   2 root     root            0 Feb  9 15:44 fd
dr-x------   2 root     root            0 Feb  9 15:44 lib
crw-------   1 root     root        1,  1 Feb  9 15:44 mem
```

```
struct file_operations {
        int (*lseek) (struct inode *, struct file *, off_t, int);
        int (*read) (struct inode *, struct file *, char *, int);
        int (*write) (struct inode *, struct file *, char *, int);
        int (*readdir) (struct inode *, struct file *, struct dirent *, int);
        int (*select) (struct inode *, struct file *, int, select_table *);
        int (*ioctl) (struct inode *, struct file *, unsigned int, unsigned int)
        int (*mmap) (void);
        int (*open) (struct inode *, struct file *);
        void (*release) (struct inode *, struct file *);
};

struct inode_operations {
        struct file_operations * default_file_ops;
        int (*create) (struct inode *,const char *,int,int,struct inode **);
        int (*lookup) (struct inode *,const char *,int,struct inode **);
        int (*link) (struct inode *,struct inode *,const char *,int);
        int (*unlink) (struct inode *,const char *,int);
        int (*symlink) (struct inode *,const char *,int,const char *);
        int (*mkdir) (struct inode *,const char *,int,int);
        int (*rmdir) (struct inode *,const char *,int);
        int (*mknod) (struct inode *,const char *,int,int,int);
        int (*rename) (struct inode *,const char *,int,struct inode *,const char *
        int (*readlink) (struct inode *,char *,int);
        int (*follow_link) (struct inode *,struct inode *,int,int,struct inode **
        int (*bmap) (struct inode *,int);
        void (*truncate) (struct inode *);
        int (*permission) (struct inode *, int);
};

struct super_operations {
        void (*read_inode) (struct inode *);
        int (*notify_change) (struct inode *);
        void (*write_inode) (struct inode *);
        void (*put_inode) (struct inode *);
        void (*put_super) (struct super_block *);
        void (*write_super) (struct super_block *);
        void (*statfs) (struct super_block *, struct statfs *);
};
```

```
$ free
              total       used      cache       free     shared
memory:       15416       4980       6436       4000       1528
swap0:        16380        300                 16080          0
```

```
lrwx------    1 root     root             64 Feb  9 15:44 root -> [0802]:1
-r--r--r--    1 root     root              0 Feb  9 15:44 stat
-r--r--r--    1 root     root              0 Feb  9 15:44 statm
```

When examined, they yield:

cmdline gives the commandline that the process was invoked with.

cwd is a magic symlink to the current working directory of the process. A readlink on the symlink gives the device and inode pair in the form shown ([0802]:1 means inode 1 on device major 8, minor 2). Likewise for **root** and **exe** which give the current root directory, and executable file respectively.

lib contains a similar magic symlink for each mapped file (usually shared libraries, thus the name).

fd contains more magic symlinks, one for each open file the process has.

mem when opened, gives access to the memory image of the process.

stat and **statm** give process state, and

env gives all of the process's environment.

4.4 net

In the net subdirectory are the socket interface and a subdirectory for each network type. My system has only tcp; others may have AX25, etc., as well.

The tcp code is based loosely on that from the BSD-NET release: likewise, the utilities are from that release (modified to run in a POSIX environment)

4.5 mm

Mm contains the memory management code. There are only three files (around 2000 lines) altogether. These provide swapping, the mmap system call and support for shared libraries.

The memory code relies heavily on the facilities of the Intel 386 memory-management system. It is probably the least portable part of the system as a whole.

As far as possible pages are shared between processes, marked copy-on-write where necessary. A typical output from the 'free' command is in the figure. Starting 15 copies of /bin/sh changed this to: Pages migrate between the buffer pool and other uses as necessary — there is no fixed size buffer pool.

Processes are fully demand-paged. However, paging activity only occurs when there is no free space: this leads to bumpy performance when memory gets tight, but is generally fine.

```
$ free
                total       used      cache        free     shared
memory:         15416       7328       6436        1652       5788
swap0:          16380        300                  16080          0
```

4.6 tools and lib

The remaining directories (**tools and lib**) contain tools for building the kernel image from its boot block and the rest of the code (by concatenating and patching) and a small subset of the C library for use when building and testing the kernel.

5 Overall

Overall LINUX takes up around 4000 lines of assembler, 66 000 lines of C, and 12 000 lines of header code (including all device drivers, all filesystems and the floating point emulator). This leads to a kernel around 400k big (stripped), that takes 12 minutes to build on my 486/50. Around 4 minutes of this is spent in disc I/O, as the ultrastor driver is not yet very efficient, and only works at around 250K/sec instead of a theoretical 700K/sec — the Makefiles invoke sync between each major section, which slows the process down.

6 Conclusion

I find the system quite usable for most purposes. It seems quite fast, and is small enough to understand. It has the utilities and programs I need, from Spice through emacs and TEX to X-Windows (although, because my card is not supported, I either have to write a driver for it or live with monochrome X).

I'm typing this document using Epoch on an X-terminal attached by ethernet to the LINUX box, using a dos emulator and TEX in other windows. I'm using the machine for reading mail, (sendmail compiled almost straight out of the box), but I haven't installed Cnews or an NNTP newsreader yet, although both are available. Using FTP and Epoch with ange-ftp, I can use the LINUX box to edit files anywhere that tcp/ip is available from our system.

So overall — it's free and it works, and is open. By whatever definition you care to choose!

7 Where can I get it?

This document would not be complete without letting you know how to get LINUX for yourself. The easiest way is to use anonymous FTP to fetch the Soft-Landing-System (SLS). It's available at:

- kirk.bu.oz.au:/pub/OS/Linux/packages/SLS/*
- tsx-11.mit.edu:/pub/linux/packages/SLS/*
- sunsite.unc.edu:/pub/Linux/SLS/*

It contains 30 discs:

1. core (disks a1-a4) What you need to get going.
2. base (disks b1-b7) Extras — more libraries, man pages, networking, text processing, etc
3. Compiler (disks c1-c3) Gcc 2.3.3 — including C++.
4. documentation (disks d1-d2) Info and man pages for all the utilities.
5. sources (disk s1) Sources for things that are kernel dependent, likely to have to be changed, or hard to find elsewhere.
6. text processing (disks t1-t3) TeX, LaTeX, etc.
7. X-Windows (disks x1-x6) The XFree 386 release.

Distributed Object Management

HARALD REISS

REISS DYNAMICS ENTERPRISE

69 Le Souef Drive
Kardinya, W.A. 6163
AUSTRALIA
hr@rdeint.DIALix.oz.au

ABSTRACT

In this paper we would like to highlight some technical aspects of object management in a distributed environment, its implementation and its influence in software development.

Object oriented design is seen as the most promising way to get better software and to reuse its components. The management and storage of networks of objects and methods are a new development in computer science. Thus a consortium called Object Management Group (OMG) is formed, which has members all over the world.

The Object Management Group defines an architecture to transparently manage objects in heterogeneous distributed environments.

An object system as a collection of objects provides services to clients by a well-defined interface. The OMG (concrete) object model specifies the structure of this interface both to client requests and the object implementation. An Interface Definition Language (IDL) allows to completely describe the object interface with all its methods.

The Object Request Broker (ORB) provides the mechanism to locate the appropriate object implementation and sets up the communication path for client requests. The ORB itself is defined by its interfaces. An implementation can be directly linked into the client and object code, run as a server or be embedded into the operating system.

The architecture is also open to integrate different types of implementations of object systems such as object libraries or object oriented database systems (OODB).

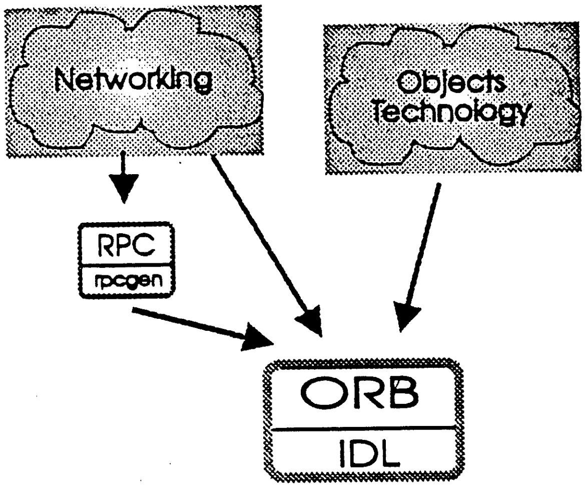
Computer Networking has matured during the last years. We have got facilities to do efficient and standardised communication between heterogeneous systems.

The socket transport interface appeared in 4.2 BSD first. This simple but powerful network interface opened the software world to a whole class of new applications independent of the underlaying physical network today. Sockets have been superseded by the X/Open Transport Interface (XTI).

Different hardware architectures are using incompatible data type representations. The presentation layer service *External Data Representation* (ONC-XDR) solved this problem by definition of a common network data type representation. Remote Procedure Calls (ONC-RPC) prepare the easy way into client server applications. One of the most popular packages is the Network File System NFS.

RPC/XDR is bundled with a definition language to specify data types and procedure calls at a higher level. A protocol compiler (*rpcgen*) creates the appropriate C code including all stubs required for the client and the server process will be created.

Object oriented design, programming and development is nothing new. SmallTalk as a pure object oriented language has been developed by Xerox PARC since the early 1970s and is available since 1980. Today the preferred language for object oriented programming is C++. Many C users see C++ as an enhanced C or just the next revision. This is an easy step into the object oriented world, even when C++ is considered more a class oriented than a true object oriented language.



Modern "object oriented" network concepts have changed the view of a network. It is no longer seen as a number of computers providing different services, but a set of objects.

Today it is generally impossible to have direct access to another applications data. Systems are open up to the networking level only. Most of the applications are still closed to their own environment.

None of the RPC derivates does give any support for this level. New mechanisms are required to open distributed applications and to allow interaction between applications. Several companies have recognised this problem.

OMG

The Object Management Group (OMG) was formed in April 1989 by a group of vendors and user organisations as a non profit organisation. The organisation "*is dedicated to maximising the portability, reusability and interoperability of software. The OMG is the leading world-wide organisation dedicated to producing a framework and specifications for commercially available object environments.*"

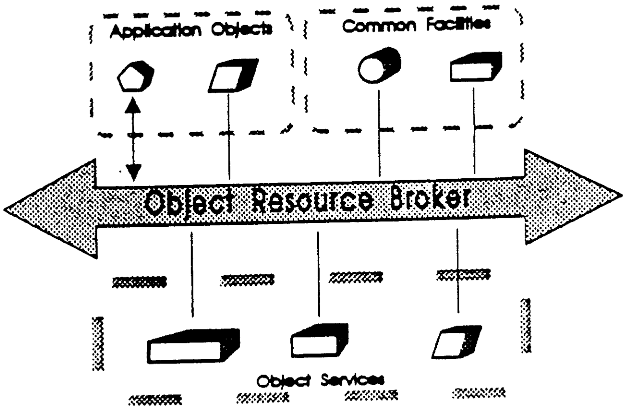
The OMG has more than 300 members today including names like Sun Microsystems, Digital Equipment Corporation, Hewlett-Packard, OSF, Microsoft. OMG itself employs very few people. Most of the work is carried out by members.

OMA

The first major result of the OMG has been the Object Management Architecture Guide (OMA). It describes an abstract object model, an architecture to implement distributed applications. The object management architecture is structured into four components:

- Object Request Broker
- Object Services
- Common Facilities
- Application Objects

Object Management Architecture



♦ The Object Resource Broker

Communication between objects is done by requests and results. The application object is able to use services of the Common Facilities by sending a request to these objects. Requests are independent of the executing machine. The underlaying network and machine architecture are fully transparent to the application. This is guaranteed by the Object Request Broker (ORB), the central component of this architecture. Services provided by the ORB are

- the *Name Service* to identify an object within the network,
- the *Request Dispatch Service* to decide which method of an object has to be called,
- the *Parameter Encoding* to have a unique data representation over the network,
- the *Delivery Services* to allow correct transport of requests and responses,
- the *Activation* to implement persistent objects,
- the *Exception Handling* to deal with error conditions and other unexpected situations,
- and the *Security Services* to identify communication partners.

♦ Object Services

Object Services provide basic functions to manage objects within a network environment. These are functions to create and manage classes and instances and to implement persistent objects. On a local level this type of functionality is provided partially by object oriented programming languages.

♦ Common Facilities

Common Facilities provide all functions of general characteristics required by many applications. They can be seen as a common class library. The application developer can use them to create his own classes tailored to the specific requirements.

♦ Application Objects

Application Objects summarises all types of applications based upon the OMA providing or using services of the system. Some types of applications are

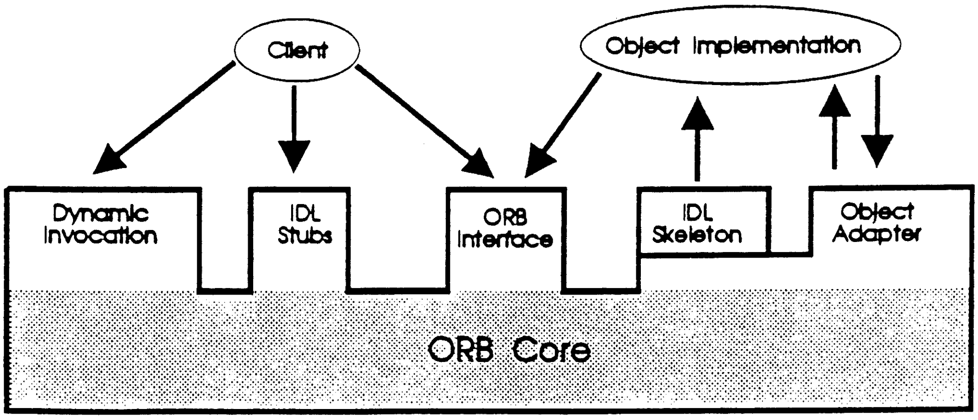
- CAD applications,
- CASE systems,
- network management applications.

OSF DME (Distributed Management Environment) and USL Atlas are concepts based on OMA.

Common Object Resource Broker Architecture

Two groups with HP, NCR, Object Design, Sun in one and DEC, HyperDesk in the other developed their own definitions of an ORB like implementation. Within the *Object Request Broker Task Force* they agreed to a common strategy in 1991, the Common Object Resource Broker Architecture (CORBA). It is a

Object Request Broker



concrete object model based on the Object Management Architecture. The document describes a binding definition of an ORB.

Two different types of interfaces are provided for the client. Both types of interface have the same capabilities.

The **Dynamic Interface** provides an interface repository to retrieve an interface definition at run time to call a method of an object. No static calls have to be created. The caller does not even need to know where the object is located (whereas ONC-RPC users have to know where the server is located!).

The **Stub Interface** is usually created by an Interface Description Language (IDL). The IDL compiler creates the interface for both the client and server side. This mechanism is very similar to the *rpcgen* program creating RPC stubs and header files.

♦ IDL

The object interface is specified as a formal description of a calling structure. The IDL syntax is very close to C++. A few extensions have been made to describe an interface in a distributed environment. IDL distinguishes between input, output and input/output parameters. An IDL *interface* definition is similar to a C++ *class*. Specifications of one interface can be inherited from other interfaces.

A compiler translates the definition into the target language. The C language mapping is specified in the current CORBA document. It shows how to translate each IDL construct into C language.

A public domain IDL compiler is available. It creates an intermediate language.

♦ Object Adapter

A generated server interface is used to call object implementations from the ORB. The object itself is able to request services from the ORB via the Object Adapter. The Object Adapter is the real interface to the ORB for an object implementation.

Object adapters have the following responsibilities:

- generation and interpretation of object references
- method invocation
- security of interactions
- object and implementation activation and deactivation
- mapping object references to the corresponding object implementations
- registration of implementations

Services not provided by a specific ORB core implementation have to be implemented in the object adapter to guarantee the same interface functionality to the object implementation.

Any special requirement needed by an object implementation have to be implemented in the object adapter (for example registration of multiple objects of an OO database in one call). This allows also performance tuning.

Examples of object adapters are:

- Basic Object Adapter
  This adapter type can be used for most object types.
- Library Object Adapter
  It is used for library objects linked to the client's program.
- Object-Oriented Database Adapter
  A special adapter tuned for OODBs.

♦ Object Resource Broker

The CORBA document allows different implementation strategies for the ORB. Objects have very different requirements how to interface the ORB depending on the object model used.

The ORB is defined by its interface only. It can be implemented as a single component, but there is no requirement for it.

Some examples how to implement an ORB are:

- Client- and implementation-resident ORB
  The ORB code is linked to the client. Interprocess communication (IPC) is used to access the object implementation.
- Server-based ORB
  The ORB is implemented as one or more server programs. IPC is used to communicate with clients and object implementations. The ORB is responsible for routing of requests between client and object implementation.
- System-based ORB
  The ORB is integrated into the operating system as a part of it.
- Library-based ORB
  "Light-weight" objects can be implemented as a library and linked into the client code. In this case the stubs are the object implementation itself. The object implementation has to trust the client not to damage its data as they share the same address space.

A client is not necessary restricted to access one ORB only. If different ORBs are able to work together, the client may use them simultaneously. Different styles of object references have to be resolved by the ORB.

The interface is organised into three categories:

- operations that are the same for all ORB implementations
- operations that are specific to particular types of objects
- operations that are specific to particular styles of object implementations

A request consists of

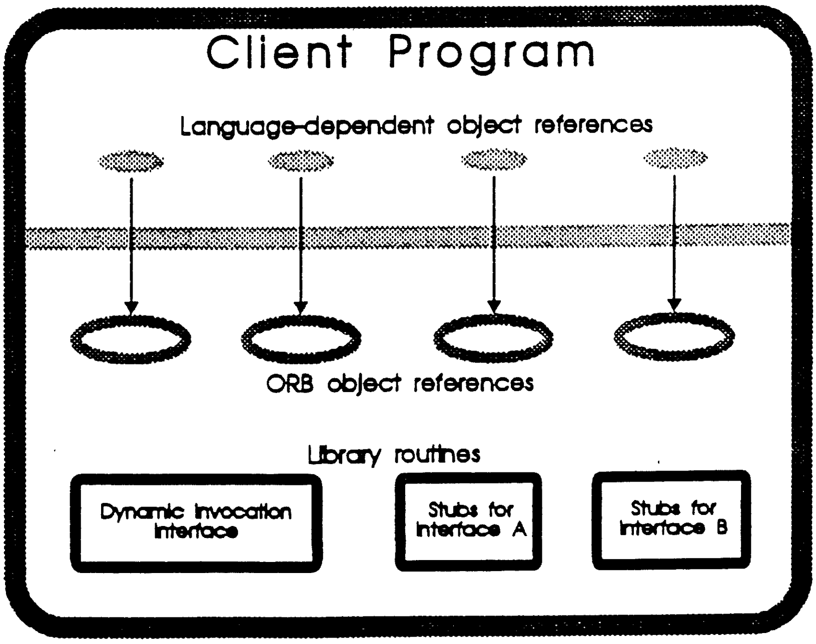
- an *operation* (name),
- a *target object*,
- zero or more *parameters* (data for target object),
- an optional *request context* with additional information about the request.

An exception is returned, if any abnormal condition occurred during the execution of a request. A request may return a single result value.

♦ Client

Objects are invoked by usage of the object reference, implemented as a language specific data type. The stubs convert this type, usually an opaque pointer, to the representation required by the ORB.

Object references can be received as an output parameter from another object invocation or an input parameter if the client is an object implementation itself. A string representation of an

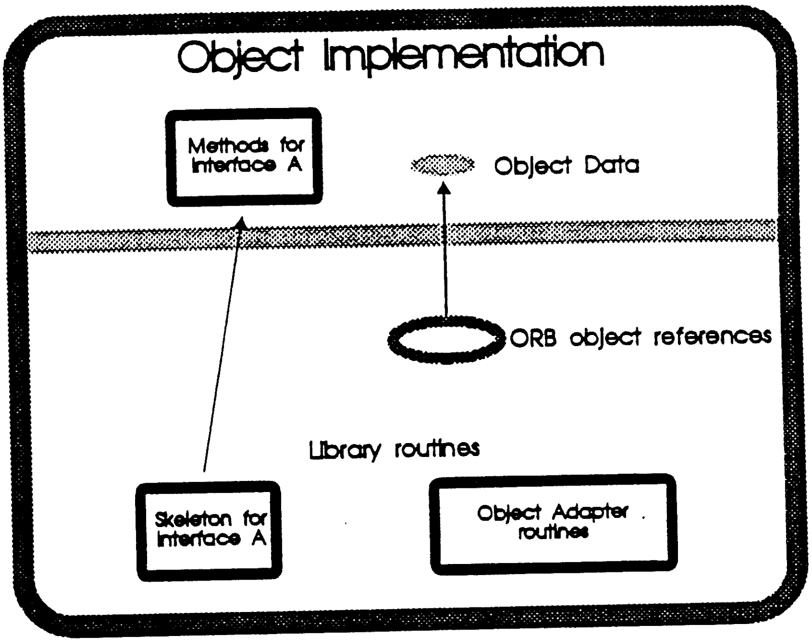


object reference can be used for permanent storage.

♦ Object Implementation

A set of data and code is required to implement the object itself. The implementation can use other object to implement its functionality. Some ways of implementation are

- (OO) libraries,
- servers,
- a program per method,
- an encapsulated application,
- an object oriented database.



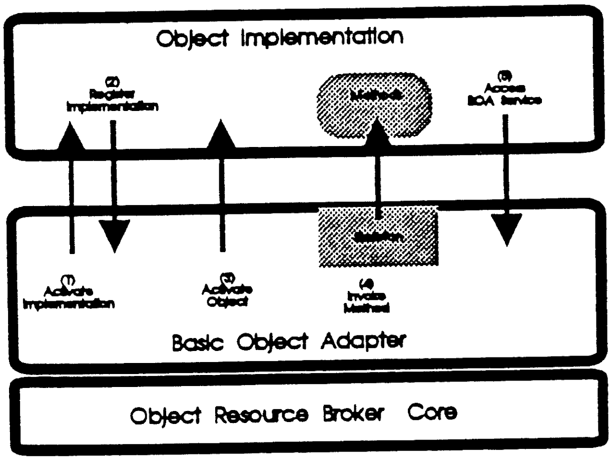
Encapsulating of existing applications allow their integration into an object oriented environment and their usage together with newly developed OO applications. This approach will especially enable the further usage of the major mainframe applications within the object oriented world.

♦ Basic Object Adapter

A standard object interface is specified in CORBA, the Basic Object Adapter (BOA). It should be available in every ORB implementation.

The main services provided by the BOA are

- Generation and interpretation of object references
  Unique references are used to identify an object.
- Identification and authentication of the calling client



- Activation and deactivation of an object implementation

The BOA has to implement persistent objects. This means permanent storage of objects (hard disk, etc.), write back of modified data.

- Calls of methods by usage of generated stubs
  There are different ways to activate a method:
  - Each method is an own program
  - Each object with all its methods is implemented as a program
  - A program implements an object and all its instances

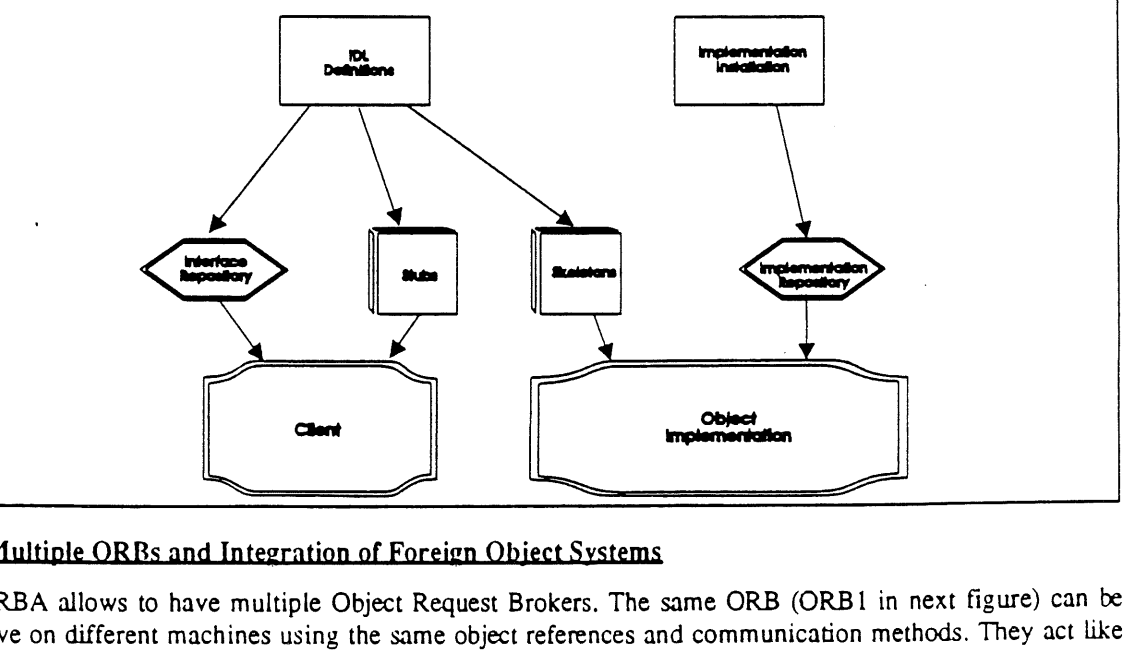
The BOA interface is mainly specified in IDL notation.

There are four different ways to activate an object implementation, also called *activation policies*:

- *Shared server* policy
  The BOA starts the server process for a number of object implementations when the first request is performed. As soon as the process is ready, it will register to the BOA.
- *Unshared server* policy
  The server process implements only one object.
- *Server-per-method* policy
  For each request, a new server is started.
- *Persistent server* policy
  The server process is started outside the BOA. Otherwise it acts like a shared server.

The way of communication between the BOA and the object implementation is not specified in CORBA. They are very dependant on the system and language environment used. The information required to connect to the object implementations is stored in the *Implementation Repository*. CORBA does not contain any further specification about the interface to it.

Interface- and Implementation Repositories



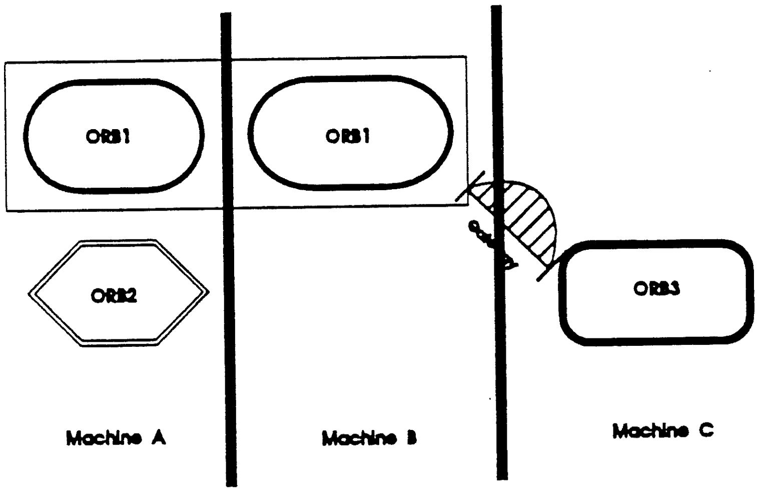
♦ Multiple ORBs and Integration of Foreign Object Systems

CORBA allows to have multiple Object Request Brokers. The same ORB (ORB1 in next figure) can be active on different machines using the same object references and communication methods. They act like a single ORB. Object references can be transferred between the different machines.

One client can use objects from more than one ORB (ORB1 and ORB2 in figure). Object references can be passed as parameter between the different ORBs. The ORB has to be able to distinguish between his own and others object references.

If two machines do not have any common ORB, a gateway is required to translate the object references and requests.

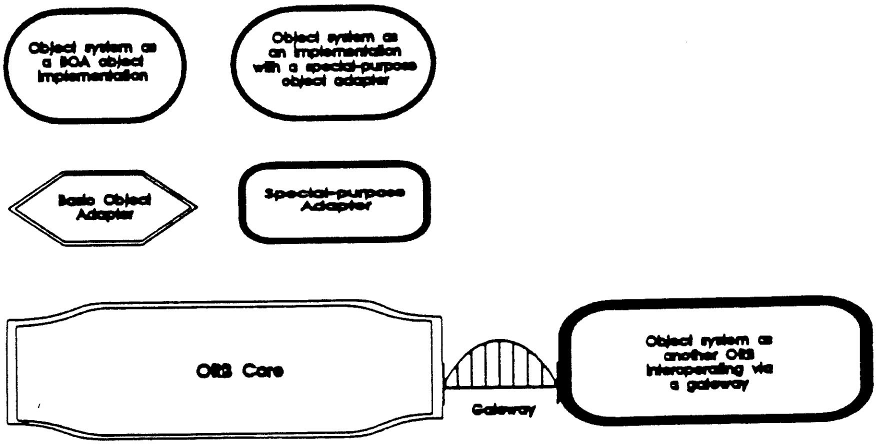
Multiple ORBs



Today there are many "non CORBA" object systems available. To integrate them into a CORBA environment, three different approaches are possible:

- Mapping of foreign system into ORB objects (BOA objects)
  The foreign object system is seen as one or more objects interfaced by a basic object adapter.
- Usage of special object adapter
  If the BOA is not suitable as an interface, a special object adapter can be used as interface.
- ORB Gateway
  The foreign object system is seen as ORB. A gateway translates between a CORBA ORB and the foreign ORB.

Different Ways to Integrate Foreign Object Systems


Object system as a BOA object implementation
Object system as an implementation with a special-purpose object adapter
Basic Object Adapter
Special-purpose Adapter
ORB Core
Gateway
Object system as another ORB interoperating via a gateway

Future Development RFI / RFP

The CORBA 1.1 document has not covered a number of areas. To fill these gaps the OMG issues Requests For Information (RFI). The response is used to prepare a Request For Proposal (RFP).

In January 1993 the OMG released the RFP *C++ Language mapping request for proposal.*

The C++ mapping will be the first object-oriented language mapping for the CORBA standard.

Special requirements are

- *reliability* to make sure incorrect IDL code does not corrupt the IDL language mapping,
- *performance*,
- *portability* to be independent of specific C++ implementation.

The final specification is scheduled for end of 1993.

A new CORBA 2.0 version is planned. Together with the C++ language mapping a RFI *ORB 2.0 Extensions* has been released. It is due end of April. 1993. Responses will be reviewed by Technical Committee (TC) "Object Request Broker 2.0 Task Force" (ORB2TF). The next step will be a RFP, scheduled for July 1993.

The addressed areas are:

- Areas in CORBA deliberately left incomplete (repository APIs)
- Additions to CORBA (additional language binding)
- Extensions to core ORB (transaction, concurrency generation)

Suggested response topics are:

- Multi-media (images, animation, video, speech) as streams of data
- Interoperability (high priority)
  Definition of interoperability
  Requirements and expectations for interoperability
  Expositions on general approaches to interoperability
  Discussion of specific mechanisms and protocols for interoperability
- Other language bindings
- Object adaptors
- Repositories
  Interface repositories (current CORBA: extract operations only, no insert, modify)
  Implementations repositories (no operations specified currently)
- Transactions
- Asynchronous messaging
- Multi-endpoint interaction and replication
- Concurrency
- Relationship to standards (for example ODP effort in ISO)
- Compliance and testability

Further topics for RFIs/RFPs are language bindings for ADA, COBOL and SmallTalk, interoperability between ORBs, asynchronous messaging.

References

1. The Common Object Request Broker: Architecture and Specification. OMG Document Number 91.12.1, Revision 1.1.
2. Object Management Architecture Guide, Revision 2.0, September 1, 1992. OMG TC Document 92.11.1.
3. C++ Language Binding Request For Proposals, December, 1992. OMG TC Document 92.12.11
4. Object Request Broker 2.0 Request for Information, December 1992. OMG TC Document 92.12.10

After the Fact [†]

A Small Guide to Intrusion Handling

by Bjorn Satdeva
<bjorn@sysadmin.com>

Somebody broke into your site over the weekend, so what are you going to do now? This is an awful question to face, especially if you are not prepared. In this article I will explore some of the issues involved handling an incident. However, since I am not an attorney, I would urge you to contact your company's legal department to discuss these issues in more detail.

All organizations should have a plan for what to do if an intrusion incident occurs. An old Chinese proverb says "It is a foolish man who first starts to dig the well when he gets thirsty." Of course, while it is impossible to plan in advance for the exact circumstances, it is easy to determine who the players are and who should be involved in untangling the situation. Of course, the question "How do you know you have been hit?" is, by itself, a very large issue. For now I will presume that you have either wired your gateway with bells and drums, or that you had been notified by a system administrator from a neighboring site that your site had been compromised.

Once a break-in has occurred, the subsequent actions that need to be taken can be split into the following steps:

First, *don't panic*. Much more damage can be done by sysadmins who have lost their heads than by an intruder. So before doing anything else, determine whether you really have an intrusion, keeping in mind that files and accounts can be created silently when you install an application.

When you have verified that your site actually has been hit, coordination will need to be set up within your site. If your site has set up a local Computer Emergency Response Team (CERT), this will be the group to contact. If not, you can contact the National CERT team at Carnegie Mellon U. (Tel: 714/268-7090). You will also need to decide which law enforcement agency should be contacted to catch the cracker. (See below.)

Don't panic.

You should then decide if you want simply to lock out the cracker or if you intend to track him down. Consult your legal department before deciding on the latter course, as your company might be liable in a civil suit filed by a site assaulted when the cracker was operating from your site. There has, to my knowledge been no law suits originating from downstream liability. CERT has been telling sites under attack to call their downstream sites, inform them of the circumstances, and let them decide their course of action.

Sooner or later, the day comes where either you have had enough, or when enough evidence has been collected, and the cracker will be locked out of your site. After the holes have been closed, you will need to do a complete recovery. At a minimum, all executables will need to be verified as being the correct ones, and all configuration files in */etc* are correct. Don't just do a reinstallation and then dump all the old files in */etc* onto the newly installed machine, as there could easily be hidden changes to, for example, *rc boot* files or the *passwd* file.

And finally, everyone should be notified that it is over and things are back to normal. This last step is often forgotten. Not doing so is impolite to your neighboring sites, which have been forced to 'alert' status during the incident, as well as anyone else who might have been involved.

Make sure you have set up a firewall.

The very best scheme, of course, is to avoid letting the cracker inside your systems in the first place. If you are on the Internet, make sure you have set up a firewall. This will remove most threats, while still allowing the users to maintain a reasonable working environment. If this is not possible, be aware of the biggest threat, which is users who have bad passwords. You can obtain *Crack*, a password cracker, from the CERT machine *<ftp.cert.org>*. This tool will enable you to find "bad" passwords in your password file. A better method is to install a new version of the *passwd* command that checks the quality of any new password before encrypting and storing it in the passwd file. One example of this is Matt Bishop's *passwd+* (available at *nic.funet.fi*). It also important to realize that many (but not all) crackers do not know a lot about UNIX, but rather follow a cookbook approach when breaking into a site. This establishes a pattern, which makes the cracker identifiable.

Legal Aspects of Intrusion Handling

Computer crime has become a problem which is now being taken seriously by the federal government. Computer Crimes were originally handled by the Department of Justice Fraud Unit, where they received little attention as that unit also han-

[†] This is a re-print from ;login, the USENIX Association Newsletter, Volume 18 Number 5

dles insurance fraud, Medicare fraud, and defense procurement fraud. A few years ago, it was moved into General Litigation, where a Computer Crime Unit, headed by Scott Charney, was created.

One of the biggest problems the Computer Crime Unit faces is educating attorneys. From a prosecuter's viewpoint, most attorneys attended law school before PCs became common, and most do not yet understand the technology. The Computer Crime Unit intends to educate the legal profession by encouraging them to learn the basics of DOS, UNIX, and networking. It is hoped that they will then be able to communicate knowledgeably to a judge to get a warrant or to a jury to obtain a conviction. The Computer Crime Unit is also working on establishing connections to the technical community, so they can call in experts in specific fields, when needed.

Another problem facing the Computer Crime Unit is the assessment of the scope of the problems. Many crimes are not reported. For example, banks do not like the publicity associated with a computer crime and will often not report such incidents. In many cases it seems to be cheaper to handle the problem administratively. The press has reported that the yearly losses are sometimes around 500 million dollars, at other times as much as 500 billion. Discrepancies this big in the amounts reported means that nobody knows what the true amounts are. The Computer Crime Unit has been in contact with the national and local CERT teams, and since last fall, US District Attorneys are required to report any activity related to computer crimes to the Computer Crime Unit.

Use a common sense approach to the law.

Another new step in fighting computer crimes is that the federal law enforcement agencies are starting to coordinate their efforts. Previously, any reported computer crime would be investigated by the local agency. This could lead to many teams investigating the same cracker, e.g., one in San Francisco, one in Chicago, and a third team in Boston. Nowadays if the three cases seem to be caused by the same cracker, the agents from the three cities will get together in order to solve the crime. Also, the two federal agencies investigating computer crimes, FBI and Secret Service, now share their case lists.

Is computer crime a new kind of crime or is it just the old kind of crime done in a new way? According to Charney, computer crime can be classified in one of the following three categories:

Computers as the target of the offense. Examples of this include PC viruses and UNIX worms, denial of service attacks, etc.

Computers as tools of the offense. This category includes embezzlement, e.g., rounding errors are accumulated into an account of the embezzler.

Computers incidental to the crime. In the old days, when the drug enforcement agency would kick in doors with a search warrant, they would find powder on the table, a stack of cash, and a little black book, with names, addresses, amounts and date of purchase. Today, the DEA is kicking in doors, and there is the powder, the stack of cash, and a stand-alone PC. In such cases, the computer is incidental to the crime.

Of these, only the first is treated by Department of Justice as Computer Crime. The other two are old type of crimes, which just are using new tools.

The Computer Crime Law

The United States has a computer crime law, *The Computer Fraud and Abuse Act of 1986* (which was originally introduced in 1984, but amended in 1986), also known as 18-USC 1030. I will not reprint the law here, but rather attempt to illustrate the practical aspects of the law, as it has been explained to me. It is important that system administrators don't get lost in the legal language, and use a common sense approach to the law.

The 1030 act comprises six parts. According to Charney, the first, 1030a1, has never been used and will never be in its current form. The statute prohibits someone from knowingly accessing a classified computer without proper authorization. The main reason for why the law will not be used is that it requires the prosecutor to prove the intent of the suspect, one of hardest thing to prove. Other existing espionage laws contain similar provisions, but do not require the proof of intent.

The second, 1030a2 is intended to protect financial institutions and has little interest for most other computer sites.

The third and fifth parts, 1030a3 and 1030a5, are the most used statutes in this law, and perhaps the most interesting for a system administrator. These statutes deal with trespassing, and protecting the computer from unauthorized access.

However, 1030a3 only applies to computers which are used (wholly or in part) for the federal government.

The fourth part, 1030a4 covers the use of federal interest computers in committing fraud. The definition of a federal interest computer is interesting: A computer becomes a federal interest computer if it is accessed by modem or through the Internet from another state. This definition allows the law to be used for computers owned and operated outside of the federal government, as long as it is in another state than the cracker. With today's connectivity, this is the usual situation.

The fifth part, 1030a5 covers intentional access to a federal interest computer (which can be any computer) without authorization, and either altering, damaging, or destroying information, and thereby causing a loss of one thousand dollars or more during a period of one year (except for medical records, where there is no minimum dollar requirement). The one thousand dollars requirement is important, as the law allows lost computer time and cost of reprogramming to be included in the loss.

The sixth part, 1030a6 applies to the use of bulletin board computers.

Besides the Computer Fraud and Abuse Act, a number of existing laws are also used for prosecution. Two examples of this are the wirefraud act and 18 USC 1014. These statutes prohibit interstate transfer of stolen property. The latter is an old law, which was passed to prevent automobile theft, but is used here when a file is transferred illegally from a computer in one state to one in another. The wirefraud act has been used when someone uses interstate wires to access a computer without authorization. Here, the cracker has been connecting and misrepresenting his/her identity.

Still, the laws are very confusing to lay people. The task of law endorsements agencies and attorneys is to determine which statutes should be used for prosecution. The tasks of the System administrators is to be able to explain to them what happened and possibly who the intruder might be.

The International Aspect of Intrusion Incidents

Often the cracker is not from the United States. Obviously, it is much easier to make a domestic investigation vs. an overseas one. For instance, it is not possible to go into another country and seize evidence or suspects. In order to gain assistance in investigating such an incident, the government policy's need to deem the crime as severe and important. In many countries, computers are not as common as they are in the United States, and thus they may may not deem computer crime to be an important problem. However, countries like England, Australia, Canada, and the Netherlands are working together to solve these crimes, requiring liaisons between the law enforcement and treaties which stipulate mutual legal assistance.

Computer crime is a growing and complex phenomenon.

Another problem is, that even when the law enforcement agencies are working together across borders, there are big differences in the laws. Scott Charney has given as one example a foreign law enforcement agency that did not have any existing statutes like the 18 USC 1030, but decided to treat the login to a computer using another person's account as forgery (the intruder was signing on as another person).

The Law Enforcement Agencies

On the federal level, there are two agencies, the FBI and Secret Service, that are involved in solving computer crime. The FBI actually comprises two parts, the Foreign Counter-Intelligence (to be contacted when classified computers are involved), and the White Collar Crime Division. Historically, the FBI has not been very active in solving computer crime. However, much has changed since Clifford Stohl was chasing the wily hacker, and today the FBI has been reorganized to better deal with this area. Agents from the FBI and Secret Service now work closely together on cases which involve both agencies. This eliminates any confusion for the system administrators about which agency to call, e.g., if you get hit, contact the agency which is easiest to get in touch with. In case of FBI, ask for the supervisor of the White Collar Crime Division.

Conclusion

Computer crime is a growing and complex phenomena. Learning more about it, its ramifications, and the legal issues surrounding it can make us more effective in our jobs.

4.4BSD Release [†]

by Keith Bostic
<bostic@vangogh.CS.Berkeley.EDU>

We are happy to send you information about our June 1993 release of 4.4BSD. This distribution is the final release that will be done by the Computer Systems Research Group (CSRG). For details on the demise of CSRG, see "The End of BSD from Berkeley" below.

This distribution is intended to be used on production systems; it has been run extensively at several test sites and has proven to be stable and reliable. However, because of the shutdown of the CSRG, there will not be anyone available at Berkeley to assist with problems, so it should not be used by sites without enough local expertise to find and fix any problems that are encountered.

The code in this distribution may be redistributed and used in released products provided that you abide by the due credit requirements listed in your license agreement. We have attempted to make the system as compliant with the POSIX 1003.1 and 1003.2 standards as was possible at the time of its release. We have not been able to run it through any of the verification test suites, thus, you should not claim conformance with either standard without first validating the code.

We had planned on releasing two versions of the software, 4.4BSD-Encumbered and 4.4BSD-Lite. Currently, we are releasing only 4.4BSD-Encumbered. The 4.4BSD-Encumbered distribution is available only to sites with UNIX/32V, System III, or System V source licenses with Western Electric, American Telephone and Telegraph (AT&T), or UNIX Systems Laboratories (USL). The 4.4BSD-Encumbered distribution is a complete distribution in the style of 4.3BSD and contains the complete source for the Berkeley Distribution.

The 4.4BSD-Lite distribution was to have been a distribution that was copyrighted by the University of California and others, but could be freely redistributed. It was to have been available to anyone and require no previous license, either from USL/Novell or The Regents of the University of California. Its license agreement and content would have been similar to that of the two BSD Networking Releases. However, USL has brought a lawsuit against the University and the University has voluntarily decided to withhold the distribution of 4.4BSD-Lite until the lawsuit is resolved.

The information below is designed to acquaint you with the details of our distribution so you can decide whether you would like to receive it. Order information is at the end of this article.

What is 4.4BSD?

This software distribution is provided on one 6250 bpi 1/2" 9-track tape or one 8 mm Exabyte cassette only. The 4.4BSD-Encumbered distribution contains complete source as well as binaries for one of the following three architectures:

- HP 9000/300 68000-based workstations
- DECstation 3100 and 5000 MIPS-based workstations
- SPARCstation I & II SPARC-based workstations (Please note that the SPARC kernel will not run on the Sparcstation 10.)

If you wish to obtain binaries for more than one architecture, they may be purchased at the same time for an additional $500.00 fee.

The distribution supports a somewhat wider set of machines than those for which we have built binaries. The architectures that are supported in source form include:

- HP 9000/300 68000-based workstations
- Intel 386/486-based machines (ISA/AT or EISA bus only)
- Sony News MIPS-based workstations
- Omron Luna 68000-based workstations
- DECstation 3100 and 5000 MIPS-based workstations
- Sparcstation I & II SPARC-based workstations

The distribution does not include the machine support for the Tahoe and VAX architectures found in previous BSD distributions. Our primary development environment is the HP9000/300 series machines. The other architectures are developed and supported by people outside the University. Consequently, we are not able to directly test or maintain these other architectures, so cannot comment on their robustness, reliability, or completeness.

The major new facilities available in the 4.4BSD release are a new virtual memory system, the addition of ISO/OSI networking support, a new virtual filesystem interface supporting filesystem stacking, a freely redistributable implementation

[†] This is a re-print from ;login, the USENIX Association Newsletter, Volume 18 Number 5

of NFS, a log-structured filesystem, enhancement of the local filesystems to support files and filesystems that are up to 2^{63} bytes in size, enhanced security and system management support, and the conversion to and addition of the IEEE Std1003.1 ("POSIX") facilities and many of the IEEE Std1003.2 facilities. In addition, many new utilities and additions to the C library are present as well. The kernel sources have been reorganized to collect all machine-dependent files for each architecture under one directory, and most of the machine-independent code is now free of code conditional on specific machines. The user structure and process structure have been reorganized to eliminate the statically-mapped user structure and to make most of the process resources shareable by multiple processes. The system and include files have been converted to be compatible with ANSI C, including function prototypes for most of the exported functions. There are numerous other changes throughout the system.

The new virtual memory implementation is derived from the MACH operating system developed at Carnegie-Mellon, and was ported to the BSD kernel at the University of Utah. The MACH virtual memory system call interface has been replaced with the *mmap*-based interface described in the "Berkeley Software Architecture Manual" (see *UNIX Programmer's Manual, Supplementary Documents, PSD:5*). The interface is similar to the interfaces shipped by several commercial vendors such as Sun, USL, and Convex Computer Corp. The integration of the new virtual memory is functionally complete, but still has serious performance problems under heavy memory load. The internal kernel interfaces have not yet been completed and the memory pool and buffer cache have not been merged.

The ISO/OSI Networking consists of a kernel implementation of transport class 4 (TP-4), connectionless networking protocol (CLNP), and 802.3-based link-level support (hardware-compatible with Ethernet®). We also include support for ISO Connection-Oriented Network Service, X.25, TP-0. The session and presentation layers are provided outside the kernel by the ISO development environment (ISODE). Included in this development environment are file transfer and management (FTAM), virtual terminals (VT), a directory services implementation (X.500), and miscellaneous other utilities.

A new virtual filesystem interface has been added to the kernel to support multiple filesystems. In comparison with other interfaces, the Berkeley interface has been structured for more efficient support of filesystems that maintain state (such as the local filesystem). The interface has been extended to include UCLA's support for stackable filesystems. These extensions allow for filesystems to be layered on top of each other and allow new vnode operations to be added without requiring changes to existing filesystem implementations.

In addition to the local "fast filesystem," we have added an implementation of the network filesystem (NFS) that fully interoperates with the NFS shipped by Sun and its licensees. Because our NFS implementation was implemented using only the publicly available NFS specification, it does not require a license from Sun to use in source or binary form. By default, it runs over UDP to be compatible with Sun's implementation. However, it can be configured on a per-mount basis to run over TCP. Using TCP allows it to be used quickly and efficiently through gateways and over long-haul networks. Using an extended protocol, it supports Leases to allow a limited callback mechanism that greatly reduces the network traffic necessary to maintain cache consistency between the server and its clients.

A new log-structured filesystem has been added that provides near disk-speed output and fast crash recovery. It is still experimental in the 4.4BSD release, so we do not recommend it for production use. We have also added a memory-based filesystem that runs in pageable memory, allowing large temporary filesystems without requiring dedicated physical memory.

The local "fast filesystem" has been enhanced to do clustering which allows large pieces of files to be allocated contiguously resulting in near doubling of filesystem throughput. The filesystem interface has been extended to allow files and filesystems to grow to 2^{63} bytes in size. The quota system has been rewritten to support both user and group quotas (simultaneously if desired). Quota expiration is based on time rather than the previous metric of number of logins over quota. This change makes quotas more useful on fileservers onto which users seldom log in.

The system security has been greatly enhanced by the addition of additional file flags that permit a file to be marked as immutable or append only. Once set, these flags can only be cleared by the super-user when the system is running single user. To protect against indiscriminate reading or writing of kernel memory, all writing and most reading of kernel data structures must be done using a new *sysctl* interface. The information to be access is described through an extensible "Management Information Base" (MIB).

The 4.4BSD distribution contains most of the interfaces specified in the IEEE Std1003.1 system interface standard. The biggest area of change is a new terminal driver. The terminal driver is similar to the System V terminal driver with the addition of the necessary extensions to get the functionality previously available in the 4.3BSD terminal driver. 4.4BSD also adds the IEEE Std1003.1 job control interface, which is similar to the 4.3BSD job control interface, but adds a security model that was missing in the 4.3BSD job control implementation. Other additions include IEEE Std1003.1 signals, FIFOs, byte-range file locking, and saved user and group identifiers.

There are several new tools and utilities included in this release. A new version of *make* allows much-simplified *Makefiles* for the system software and allows compilation for multiple architectures from the same source tree (which may be mounted read-only). Notable additions to the libraries include functions to traverse a filesystem hierarchy, database interfaces to *btree* and hashing functions, a new, fast implementation of *stdio* and a radix sort function. The additions to the utility suite include greatly enhanced versions of programs that display system status information, implementations of various traditional tools described in the IEEE Std1003.2 standard, and many others.

We have been tracking the IEEE Std1003.2 shell and utility work and have included prototypes of many of the proposed utilities. Because most of the traditional utilities have been replaced with implementations conformant to the POSIX standards, you should realize that the utility software may not be as stable, reliable or well-documented as in traditional Berkeley releases. In particular, almost the entire manual suite has been rewritten to reflect the POSIX defined interfaces, and in some instances it does not correctly reflect the current state of the software. It is also worth noting that, in rewriting this software, we have generally been rewarded with significant performance improvements. Most of the libraries and header files have been converted to be compliant with ANSI C. The default compiler (gcc) is a superset of ANSI C, but supports traditional C as a command-line option. The system libraries and utilities all compile with either ANSI or traditional C.

Work has also progressed in several other areas. Several important enhancements have been added to the TCP/IP protocols including TCP header prediction and serial line IP (SLIP) with header compression. The routing implementation has been completely rewritten to use a hierarchical routing tree with a mask per route to support the arbitrary levels of routing found in the ISO protocols. The routing table also stores and caches route characteristics to speed the adaptation of the throughput and congestion avoidance algorithms.

The Kerberos (version 4) authentication software has been integrated into much of the system (including NFS) to provide the first real network authentication on BSD.

This release includes several important structural kernel changes. The kernel uses a new internal system call convention; the use of global ("u-dot") variables for parameters and error returns has been eliminated, and interrupted system calls no longer abort using nonlocal goto's (longjmp's). A new *sleep* interface separates signal handling from scheduling priority, returning characteristic errors to abort or restart the current system call. This *sleep* call also passes a string describing the process state, which is used by the *ps*(1) program. The old *sleep* interface can be used only for non-interruptible *sleeps*. The *sleep* interface (*tsleep*) can be used at any priority, but is only interruptible if the PCATCH flag is set. When interrupted, t*sleep* returns EINTR or ERESTART.

Many data structures that were previously statically allocated are now allocated dynamically. These structures include mount entries, file entries, user open file descriptors, the process entries, the vnode table, the name cache, and the quota structures.

The End of BSD from Berkeley

For the following three reasons, the CSRG clearly could not continue in its present form.

1. Funding had become increasingly time-consuming and difficult. We were spending more and more of our time obtaining funding, time that we would have preferred to spend working on BSD. As many of you are intimately aware, computer corporations are actively seeking ways to reduce discretionary outlays. Also, as UNIX vendors have developed their own research groups, the work of the CSRG became less necessary to them. Finally, making BSD freely redistributable resulted in fewer distributions sold, as other organizations sold our releases for less money.

2. Support within the University of California declined as BSD became less widely used internally. Victims of our own success, many

of the features once found only in BSD are now available from every vendor.

3. The system has become too large and complex for a group of four to architect and maintain. In the last few years it became obvious to us that we had to expand the size of our group if we wanted to continue developing and distributing a complete UNIX system. Expansion was impossible given the external funding environment and the space constraints imposed by the university.

BSD has always been a community effort, and, as a community effort, does not rely on a small group of people in Berkeley to keep it going. BSD will not go away, but will live on through the free software and commercial efforts of many people.

We thank you for your support over the years, your funding, and, of course, the software you've contributed to make the BSD system what it is today!

Sincerely yours,

Marshall Kirk McKusick
Research Computer Scientist
Computer Systems Research Group

If you are interested in obtaining the distribution, please contact:

Pauline Schwartz, Distribution Coordinator
Computer Systems Research Group
Computer Science Division, EECS
University of California
Berkeley, California 94720

System Administration Tools...

Your Vendor Never Told You About: The Publication

by Elizabeth D. Zwicky
<zwicky@erg.sri.com>

System administrators in general live outside the academic "publish or perish" framework. On the other hand, there are few places where "publish and flourish" is not true. One system administrator of my acquaintance claims to have settled an argument with a particularly obstreperous user by dropping a stack of LISA proceedings onto the table and saying "My publications are bigger than yours." The user deferred to superior credentials.

Publications are relatively cheap credentials to get. It takes time and work to write a good paper and submit it, and there's an emotional cost to getting rejected, but that's nothing to the effort involved in getting a degree, for instance. Furthermore, everybody benefits: you get the fame, your employer gets its name in print, and everybody who reads it gets information.

You don't need to be able to write beautifully, and you don't need to have completely brilliant ideas. You need to be able to express an idea with reasonable clarity, and be willing to work with an editor, and to take reasonable precautions like running a spelling checker. You do need to have an idea to express, but it doesn't need to be earth-shaking; a good new idea or a good summary of other information is quite sufficient. For instance, if you write a program that you find useful, it's probably possible to get at least a short paper out of it. If you find yourself explaining a topic frequently on USENET, you can probably get a paper out of it. [*Editor's Note: Or at least a short article in ;login:.*]

Internet Hypertext is Here †

by Tony Sanders
<sanders@bsdi.com>

Have you wondered how to navigate the Internet to find specific data? Have you found yourself lost in a twisty maze of Gopher holes? The World-Wide Web may be just what you are looking for.

The World-Wide Web (aka Web, WWW and W3) is a wide-area, hypermedia information-retrieval project. Tim Berners-Lee began development on Web in March, 1989 to simplify access to the large body of documents generated at CERN. Web browsers provide a uniform interface for data available via a variety of Internet protocols (e.g., Gopher, WAIS, NNTP, FTP and HTTP).

> *The World-Wide Web is a wide-area, hypermedia information-retrieval project.*

WWW services are now widely available on the Internet. Users can wind their way into ever-twistier mazes of hypertext links where they can "warp" from geological data to Lewis Carroll's *Through the Looking Glass* before you can say "drink me."

Getting Started

To get started, you will need a machine with Internet access and a piece of software known as a Web browser. If a browser is available for your platform, it is worth the time spent installing it. If all else fails, you can probably use one of the following tty-based browsers:

- CERN Line-Mode
  ```
  telnet info.cern.ch
  ```
  (no login required)
  This connects you to the public access line-mode browser at CERN (in Switzerland, be gentle with overseas links).

- Lynx
  ```
  telnet ukanaix.cc.ukans.edu
  ```
  login: www
  This connects you to the public access Lynx browser at the University of Kansas.
  Lynx source is available via anonymous *ftp* from *acs220.cc.ukans.edu* in *pub/lynx/*.

- *Emacs* W3-mode
  WWW mode for *emacs* (works best with Lemacs or Epoch) was written by William Perry *<wmperry@indiana.edu>*. Source is available via anonymous *ftp* to *moose.cs.indiana.edu* in *pub/elisp/w3/*.

The following World-WideWeb browsers are available for X11 platforms:

- NCSA Mosaic
  NCSA Mosaic is available via anonymous *ftp* from *ftp.ncsa.uiuc.edu* in *Mosaic/xmosaic-binaries*. You can pick up binaries for Alpha, DEC, IBM, SGI, and Sun or get the NCSA Mosaic sources in *Mosaic/xmosaic-source*. This version requires Motif.

- TkWWW (Tcl/Tk based)
  This version is available via anonymous *ftp* from *export.lcs.mit.edu* in */contrib/tkWWW-**. Requires Tcl/Tk. Includes a WYSIWYG editing mode.

- ViolaWWW
  ViolaWWW is available via anonymous *ftp* from *info.cern.ch* in */pub/www/src/viola920730.tar.Z*. It's still in beta.

- MidasWWW
  MidasWWW is available via anonymous *ftp* from *info.cern.ch* in */pub/www/src/midaswww-1.0.tar.Z* or from *freehep.scri.fsu.edu* in *freehep/networking_news_email/midaswww/midaswww_1.0.tar.Z* This version requires Motif and the UIL utilities.

Browsers for other platforms include:

- MACWeb (Macintosh)
  This is available via anonymous *ftp* from *info.cern.ch in pub/www/bin/mac/MacWWW_V1.02.hqx*. Check out the file *README_V1.02* for details. Requires *Mac-TCP* 1.1.1 or later.

- Cello (PC/Windows)
  This is available via anonymous *ftp* from *fatty.law.cornel.edu* in *pub/LII/Cello.*. Currently requires the Distinct TCP/IP stack. They are working on a Winsock compliant version.

- NextWWW (NeXTStep 3.0)
  This is available via anonymous *ftp* from *info.cern.ch* in *pub/www/bin/next*.Includes a WYSIWYG editing mode.

† This is a re-print from ;login, the USENIX Association Newsletter, Volume 18 Number 5

Browsing the Web

The heart of the WWW hypertext system is a document-reference scheme called URLs (Uniform Resource Locators). When you select a hypertext link, the browser automatically extracts the URL from the document. However, sometimes you will need to input the URLs by hand (for example, you need to input by hand the pointers to hypertext documents contained in this article.) For these occasions, browsers usually provide a "Goto" or "Open" dialog.

Here are some basic URLs to give you a feel for what they look like and how they work:

- *ftp://info.cern.ch/pub/www/src*
  This means use anonymous *ftp* to get */pub/www/src* from *info.cern.ch*
- *gopher://gopher.ora.com/*
  This means use gopher to retrieve the root menu from *gopher.ora.com.*
- *http://www.vuw.ac.nz:80/non-local/gnat/www-primer.html*
  This means use HTTP on port 80 to get */non-local/gnat/www-primer.html* from *www.vuw.ac.nz.*

See *http://www.ncsa.uiuc.edu/demoweb/url-primer.html* for more detailed information about URLs.

All you have to do now is run your Web browser software. When it first starts up, it will provide you with what is know as a *Home Page.* The Home Page is a hypertext document just like any other, the distinction being it's the first document you see. It will typically contain hypertext links to documents that outline global and local services that are available.

> *The World-Wide Web is growing to become a delivery platform for on-line books and journals.*

Here are shortcuts to some interesting locations on the Web (the hypermedia features require NCSA Mosaic, other browsers will be supporting these features in the near future):

- Expo
  Hypermedia exhibits including "Vatican," "Soviet Archive," "1492: An Ongoing Voyage,"and "Dead Sea Scrolls"
  http://sunsite.unc.edu/expo/ticket_office.html
- Weather Maps/Reports
  Current weather map for the USA
  (You can select a location on the map and WWW will display a weather report for that area.)
  http://www.mit.edu:8001/usa.html
- Honolulu Community College Campus Tour
  Multimedia tour of the Honolulu Community College campus
  (See the "HCC Campus Map and Directory" under "Maps and Directories" to help find your way around campus. The tour includes MPEG movies for some locations. There is also an exhibit on the Berlin Wall Freedom Monument and you can explore the Dinosaur Exhibit.)
  http://pulua.hcc.hawaii.edu/

The World-Wide Web is growing to become a delivery platform for on-line books and journals. O'Reilly & Associates just announced they will be delivering a product called Global Network Navigator (GNN) via the Web. GNN will consist of a regular news service, an on-line magazine, *The Whole Internet Interactive Catalog,* and a global marketplace containing information about products and services. For information about a free subscription send email to <*info@gnn.com*>.

For more information, the newsgroup *comp.infosystems.www* is available for general discussions. There is also a mailing list oriented towards development issues.To subscribe send email to <*listserv@info.cern.ch*> with the body "SUBS www-talk." The mailing list itself is <*www-talk@info.cern.ch*>. If you have a Web browser you can get more information from the following locations:
http://info.cern.ch/hypertext/WWW/TheProject.html
http://info.cern.ch/hypertext/WWW/Bibliography.html
http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/whats-new.html

An Update on UNIX-Related Standards Activities †

by Nicholas M. Stoughton
USENIX Standards Report Editor
<nick@usenix,org>

IN MEMORIAM

On Thursday, the 15th of July, at 7:20 in the evening, the Language Independent Specification Requirement died peacefully, in its sleep. The LIS requirement left one child, the POSIX.1 Language Independent Specification, which has now filed to change its name. LIS will not be missed by any of the PASC members. No flowers should be sent, but contributions to the furtherance of non-LIS standards will be gratefully received.

Back To Work?

The LIS wars have raged for a year now, with the armies of the victorious righteous led principally by Stephe Walli, Pete Meier and Paul Rabin. The entire UNIX community owes them a resounding "thank you" for having finally removed this onerous burden from our shoulders. Since Language Independent Specifications were originally mandated, attendance at POSIX meetings has fallen steadily, by thirty to forty percent, and progress has been at a snail's pace. This lack of progress has wounded the Open Systems industry seriously. Homogeneous, nonstandardized, proprietary systems, such as Windows-NT (or Win-doesn't, to use its new nickname), have had the perfect argument in their favor there are no standards to conform to.

The standards that are published, such as POSIX.1 (ISO 9945-1:1990), and POSIX.2 (now in print), have often been forced to leave out widely accepted industry practice for reasons of expediency. The work to add, for example, symbolic links to POSIX.1, or sockets to POSIX.12, has been in hand for some time, but so hampered by the LIS and Test Assertions requirements that they are still some way away from completion.

Freed from these shackles, working groups can now get down to the business of making new standards. They need help, encouragement, and support. For example, the X/Open Transport Interface work of the POSIX.12 group was funded by X/Open. The socket version of the same standard has had to progress on an entirely volunteer basis. If you care about sockets, help them!

All sorts of problems arise from slow progress. The POSIX.4a thread API shows a classic case of this; currently at Draft 8, the standard has been so long coming and so needed, that many people have implemented POSIX threads (or pthreads) from much earlier drafts (such as Draft 4). Things have changed dramatically in the interface from Draft 4 to 8, and when the final standard emerges (at Draft 9 or 10) those on earlier versions of the interface will be faced with major problems upgrading. In the meantime, people mistakenly believing they are buying standards-based products, are developing more and more applications on these old interfaces.

So let us get back to work, shake off the torpor, and try to make some visible progress over the next few months. It is also essential that attendance at POSIX meetings climb again. Falling numbers mean that costs rise and revenues fall. If we are not careful, POSIX will just be forced into chapter 11 bankruptcy, and that will be the end of standards as we know them! Some would think that this would be good. However, don't forget the kind of standards that other bodies, who might well step in to fill the gap, produce: e.g., OSI, PCTE, and Standard BASIC.

User Requirements

Not all of you who read *;login:* also get to read a newsletter published by the IEEE, called *The Standards Bearer*, which discusses the entire IEEE standardization effort. Andrew Salem, Staff Director of IEEE Standards and Secretary of the IEEE Standards Board has an interesting article in the April 1993 edition entitled "Standards Users Are We Listening To Them?"

I reproduce this article here, since, while not aimed at in particular, it is of direct relevance to the POSIX effort.

"Standards Users Are We Listening To Them?"

> Standards activity is very expensive to industry, government, and the public interest groups that support it. It deserves a management system that is at least as good as that which we apply to our individual organizations.
>
> In such a management system, the first step should be to determine the purpose of the

† This is a re-print from ;login, the USENIX Association Newsletter, Volume 18 Number 5

> standard or the standards activity. This is where the user will be identified. There will most likely be a first, second, and possibly third level of user to identify. The primary user is that interest group that the standard or the standards activity is intended to serve. This is the most fundamental issue that a standards group must address, and once identified, every management decision that follows must relate back to assure that the standard does serve the user and user requirements. Such an approach would produce a standards activity quite different from some of those we see today.
>
> It wouldn't be so hard to meet user requirements if we could all agree on who the real users are. John Rankine, IEEE Standards Board member, has pointed out that users exist at varying levels of interest, knowledge, and need. And within the system of standards, user interest, knowledge and need will vary significantly. But in all cases, the user is unique to the standard. Yet I don't know of any definition of user in the context of standards activity. And so we need to do two things. First, we must clearly define the user in the context of standards activity. Second, we need to establish a management system for standards activity that goes beyond the procedures.
>
> In my opinion, user requirements are paramount for any standards activity; everything else is secondary. With this attitude, the cardinal principles of standardization balance, consensus and appeals must be considered and appropriately applied to achieve user requirements. As an example, the present American National Standards Institute (ANSI) process allows any single interest group to comprise 50% of the total membership of a balloting body. This right should be restricted to a user group. Further the general interest group can constitute more than 50% of the voting body. Yet the general interest group is the least affected group, having the least financial interest in the standard and the least public responsibility for the standard. Nevertheless, the process that we use today allows this group, in effect, to control the standard. These rules almost assure that user standards will not be met. The definition of balance has to be reconsidered who votes and at what level in the process must user requirements be considered?
>
> In the context of meeting user requirements, what constitutes consensus? Is it a consensus of all interest groups or of the users? The word requirements implies something the user can't live without — a must. The present rules require that all interested parties have an opportunity to be participants. I agree with that. However, in practice, all interested parties vote, and that means that the users will be a minority in the process. There are some standards activities in which only the users vote on the final approval of a standard. Most of these activities involve public safety issues, and I believe it is a proper procedure in these cases. While consensus is sacred in the standards community, I think that in its present form it is overdone, misused, and serves as a weapon against achieving user requirements.
>
> The appeals process is too loosely applied and is successfully used to delay the standards process. I am in favor of putting a framework around the process to clearly define how this process can be included in the context of achieving user requirements. After all, due process is not endless process.[1]

"User requirements are paramount for any standards activity; everything else is secondary." My favorite forgotten user requirement is the production of standards in a timely fashion. Much of the recent debate on Language Independence and Test Assertions has been forced because we needed to break the log-jam and try to get some standards back on the road toward production.

Consensus by exhaustion, now a popular phrase at POSIX meetings, is not to be encouraged!

As Andrew Salem concludes, due process should not be endless process.

Report on ISO Working Group

John Hill <jhill@corp.bb.unisys.com> reports on the May meeting in Heidelberg, Germany:

On May 10–14, 1993 SC22/WG15, POSIX and its rapporteur groups met in Heidelberg, Germany. This article reports the key outcomes of that meeting. Note that the meeting approved several resolutions and assigned several action items. At the end of this report I have reproduced the text of those resolutions thought most important. The action items are not included as they're too boring.

[1] From "Standards Users Are We Listening to Them?", by Andrew Salem, reprinted with permission from the *IEEE Standards Bearer*, Vol.7 No.2, April 1993.

Readers should note that I acted as the head of the US delegation. As a result, this report shows a bias in favor of US participation and participants. This is not to denigrate the important contributions made by non-US participants in any way. I focus on the US perspective because that is how I see things. To quote the memorable former Houston Oilers football coach, Bum Philips, "You got to dance with them what brung you."

Attendance

The first thing to understand about this meeting is who, or really which countries, participated. The usual suspects were rounded up. There were delegations from seven countries. The numbers represent the corresponding number of delegates. Participants came from: Canada (3), France (1), Netherlands (2), Germany (1), Japan (2), UK (3), US (5, plus 4 other attendees, for various reasons).

Major Decisions Based on Contributed Documents

At the risk of pointing out the obvious, I note that the fundamental way by which business is conducted at international standards meetings is by formal document. If a country wants to accomplish something, the best means is to contribute a document stating their goal. If necessary, supporting documentation can additionally be contributed.

As has been its practice, the US contributed the most to the meeting, as measured by the number of formal documents submitted. The following are the most notable US contributions together with the corresponding meeting outcomes.

a. **IEEE/CS/PASC status report (WG15 SD-11)**
This states the current development status for all the PASC projects. The contribution tells JTC1/SC22/WG15 what they can expect, when it should arrive, and how large it will be.

b. **Presentation of language independent specifications (WG15 N383)**
This contribution identifies the situation the US is facing given its obligation to deliver language-independent specifications (LISs) for POSIX APIs. Outcome: The US initiated a discussion of the current situation about LIS. A presentation focused on both the US options for developing LISs, and requirements assumed to be fulfilled by LISs. WG15 provided guidance to PASC.

c. **Development synchronization plan (WG15 N352R)**
This contribution identifies how the US proposes JTC1/SC22/WG15 develop and approve interpretations to reflect recently approved JTC1/SC22 procedures for interpretations. Outcome: WG15 adopted the US-sponsored plan.

d. **Recommendations on management of rapporteur groups (WG15 N356)**

This contribution recommends JTC1/SC22/WG15 closely manage the work of its rapporteur groups. Outcome: WG15 adopted the US proposal to closely manage its rapporteur groups.

Other WG15 Decisions

Some of the decisions reached by WG15 were not based on contributed documents. The most important of those were:

1. Lead Rapporteur for Conformance Testing: WG15 named Gerry Powell, succeeding Roger Martin.
2. Ada Binding: Following continued resistance from France, WG15 decided not to register IEEE Std 1003.5 (Ada Binding) as a Committee Document (CD). The US advised WG15 of its likely intent to submit the standard for fast track approval.
3. Real Time Extensions: WG15 requested the US to submit the latest draft of P1003.4 for registration as a Draft Amendment (DAM) to 9945.
4. FORTRAN 77 Binding: The US initiated a discussion of the possible submission of IEEE Std 1003.9 for fast track approval as a technical report. WG15 supports this and instructed the convener to report it to SC22.
5. Test methods: WG15 approved a disposition of comments on CD 13210, test methods for 9945.
6. Model for Conformance Testing: The UK provided an early draft of a potential New Project (NP) to develop a generalized model for conformance testing. WG15 decided to distribute the draft for comments.
7. National Profiles: Japan and the internationalization rapporteur group provided a draft NP to develop a technical report for guidelines to developers of national profiles. Discussion will take place at the next WG15 meeting.
8. NP for LIS Guidelines: WG15 decided to not pursue developing a standard for these guidelines. Instead, it encourages WG11 to develop them.

In summary, this was a successful meeting due mainly to the effectiveness of each of the delegates (not just those from the US but from all the other countries, too.) The next meeting is to be hosted by the United States. It will be held 25-29 October 25–29, 1993 in Annapolis, Maryland, immediately following the October PASC meeting.

Resolutions

RESOLUTION 93-233
Requirements on Specification Methods

SC22/WG15 reaffirms the following requirements relating to the choice of specification methods for projects JTC1.22.21, JTC1.22.39, and JTC1.22.42:

1. *To provide a clear and authoritative functional specification for POSIX services, and*
2. *To provide coordination and consistency between multiple language bindings to the same services, and reaffirms the following related objectives:*
3. *To support easier and faster development of language bindings,*
4. *To support the development of language bindings that are appropriate for each programming language,*
5. *To support the development of language bindings by independent Standard Development Organizations, and*
6. *To support the development of POSIX standards in a timely and efficient manner.*

RESOLUTION 93-234 PASC Information

Whereas SC22/WG15 has a requirement to develop both language Independent Specifications (LIS) and Language Dependent Specifications (LDS), and

Whereas SC22/WG15 recognizes that, due to resource constraints, the current approach to the development of PASC LIS standards is having a severe negative impact on the ability of the US member body to develop and forward LDS standards in a timely manner, and that this has an equal effect on the work of SC22/WG15, and

Whereas SC22/WG15 notes that the current US development group may discontinue the current requirement for the parallel development of LIS and LDS, for the reasons given above, and

Whereas SC22/WG15 recognizes that the current work in the area of LIS is not yet mature, and is the subject of work in other groups, such as WG11, and

Whereas SC22/WG15 recognizes that the requirement that LIS was intended to satisfy may be met in other ways, and

Whereas SC22/WG15 wishes to ensure that both LIS and LDS are brought forward into the international arena in as timely a manner as possible,

Therefore SC22/WG15 resolves: To investigate the most appropriate means of satisfying the requirements and objectives contained in Resolution 93-233, for which the current LIS methods were originally formulated, taking account of the work of other groups, such as WG11.

1. *That it will accept LDS documents for progression to DIS, without an accompanying LIS, on an individual basis where Member Bodies support the action, until such time as it determines the precise form in which the requirements and objectives, for which LIS was originally formulated, are to be met.*
2. *To request the US to preserve and progress in a timely manner the LIS work that is currently underway, taking fully into account the requirements and objectives set out in SC22/WG15 Resolution 93-233, and to ensure that US standards developed for progression within SC22/WG15 satisfy those requirements and objectives.*

RESOLUTION 93-235 Disposition of IEEE 1003.9

Whereas there exists a completed IEEE FORTRAN 77 binding to 9945-1:1990 (P1003.9),

Therefore SC22/WG15 invites the US to forward the IEEE 1003.9 document to JTC1 with the recommendation that it be fast-tracked as a Technical Report Type 2.

RESOLUTION 93-243 Appreciation of Mr. Rabin

SC22/WG15 thanks Mr. Paul Rabin for the presentation he gave on the status of the LIS activities within the IEEE PASC *and the various options available to continue this work.*

RESOLUTION 93-244 NP for LIS Guidelines

Whereas the general question on Language Independence is clearly the focus of SC22/WG11, and

Whereas WG11 has drafted an NP for a Type 3 Technical Report on Guidelines for the Preparation of Language Independent Service Specifications;

Therefore WG15 encourages WG11 to proceed with the NP for a Type 3 Technical Report on Guidelines for the Preparation of Language Independent Service Specifications, taking into account the requirements and objectives for specification methods listed in Resolution 93-233.

(This Resolution supersedes WG15 Resolution 92-197.)

Report on POSIX.0: Guide to OSE

Kevin Lewis <klewis@gucci.ent.dec.com> reports on the July 16–20. 1993 in Denver, CO:

For the first time in a long while (in fact possibly since the first meeting in March 1988), we had a very loose meeting schedule, essentially because the document is in the hands of our production editor, Hal Jespersen, who is producing the next draft of the guide. This is to be an interim draft, which will be numbered 15A, and will serve strictly as an accountability draft. The section leaders will use this interim draft to ensure that all of the changes they submitted as a result of ballot resolution have been made. Once this is turned-around, Hal will produce draft 16, which will be sent to the IEEE for recirculation and forwarded to SC22 through WG15 for Committee Document (CD) registration. The goal is to commence the recirculation by late August.

The group met with POSIX.22, formerly known as the Distributed Security Study Group, to discuss how its work should relate to the work of POSIX.0 and to discuss the possible future integration in the guide. What is meant by "integration" is yet to be defined. We may actually decide to insert the final product into a future revision of the POSIX.0 Guide or simply point to it. One key concern that arose out of the joint session is the apparent decomposition of the application platform as proposed by POSIX.22 in order to expose other API's for security purposes. This conflicts with the model developed by POSIX.22. Given that the last comment on this issue from a Double Deuce member was preceded by his pounding his fist on the table, it became apparent that no consensus was going to be reached. The groups agreed to meet again jointly during the October 1993 meeting, in Bethesda.

The group held another joint meeting with POSIX.18, which is developing the Interactive System Application Environment Profile (AEP). This work is quite important to POSIX.0 because it serves as a real live profile that can be submitted to ISO as an ISP. It can also serve as a model for future profiling work and makes the concept of profiles a real one. The chair of POSIX.18 started by informing us that the profile would be held in limbo until other base standards work that served as normative references within the profile was completed. We argued (in an encouraging tone, of course) that these standards should be placed in an annex in order to allow this work to go to ballot. Later in the week, the POSIX.18 working group agreed to take this approach.

The remainder of the week was spent discussing the issue of future of our own work. This started in our normal "organized chaos" manner. Being creative during a period when we've had to be quite ballot- focused was a challenge. The result of the discussion yielded three areas that appear at present to be the priority for future work:

- distributed processing,
- system and fault management, and
- user profiles.

Individuals with interests in each of these areas agreed to write up draft Project Authorization Requests (PARs) for each topic with the objective of discussing them during the October meeting. These PARs are for discussion only and not for submussion to the PASC SEC just yet. The group also agreed to invite some folks from outside the effort to come in and give us some ideas on what we should be considering for future work. These will be from the OSE Implementors Workshop (OIW) and SC21.

As the meeting concluded, the co-chair reminded the group not to forget that we still had the goal of getting over the hump of our first recirculation to which a note of consensus was reached with a collective sigh.

Report on POSIX.7: Systems Management

Matt Wicks <wicks@fnal.gov> reports on the July 12–16, 1993 meeting in Denver, CO:

[*Editor's note: Two reports on the work of the POSIX.7 system management activities are presented this time. With the formation of a new project to address User Administration, it is useful to have a special report on this new activity.*]

Introduction

Two of the three POSIX.7 System Administration small groups met during the week:

- POSIX.7.2 – Software Installation and Management
- POSIX.7.3 – User and Group Administration

The week also contained three plenary sessions that also included representation from the POSIX.7.1 Printing Administration working group.

Plenary Sessions

The one significant issue discussed at the plenary sessions was how much coordination and commonality is desired and should be required between the various POSIX.7 standards? This is an issue that has been wrestled with for nearly two years (in fact from about the time it was agreed to form small groups, each with tits own

Project Authorization Request (PAR) and document to produce). Now that one document has actually entered formal ballot, and another document is a couple of meetings away from entering ballot, these issues need a resolution.

One possible solution to this problem is to have a POSIX.7 document that is common to all POSIX.7 standards. This could include items such as common utility options, common non-normative sections such as task descriptions, common object model, etc. No resolution was reached on the above topic, but Martin Kirk, Chair of POSIX.7 has taken an action item to determine how all of POSIX.7 will fit together as well as addressing issues with respect to the formation of the ISO document that covers POSIX system administration, ISO 9945-3.

Independent of the above problems, POSIX.7 decided it was important to agree upon a standard syntax for options that will be common to utilities from all of the small groups. This specifically addressed the use of -x/-X.

Note the following description of these option letters indicates my bias as an active member of the POSIX.7.2 small group! The -x option is used to pass in extended options or attributes. This option letter is required because there are large number of possible options/attributes that could potentially be required of system administration utilities. It is impractical, if not impossible, to create a different option letter for each possible item.

An example of -x for printing would be something like

```
-x "-media-used iso-a4-white"
```

whereas for software, a typical use is

```
-x enforce_dependencies=true
```

The -X option is used to specify a file that contains on each line individual options/attributes that could have been specified using multiple -x options.

The end result of all of this, is that after several meetings of discussion on these two options, a common syntax was developed for all POSIX.7 small groups. It will require each small group to make some modifications.

Print Administration

This small group did not meet because they were in the process of going through formal ballot. Formal ballot was scheduled to end on July 29. The balloting group consists of 72 people. About half of the members are from IBM, HP, DEC, USL, and Sun.

This is the first system administration standard to go to ballot, so many people (well at least myself) are quite interested to see how it proceeds. There has been much discussion over the past several years of the appropriateness of the POSIX.7 work. The reaction to this ballot should shed some light on this issue.

Software Installation and Management

The Software small group recently conducted a mock ballot and is working towards their goal on entering formal ballot towards the end of Q94. The purpose of this meeting was to try to finalize any major technical issues for incorporation into the next version (Draft 10). In addition, the entire document was reviewed for lesser technical issues as well as both major and minor editorial issues.

The first major issue discussed was whether or not the standard should include an Application Programming Interface (API) as well as a command line interface (CLI) for certain functions. The decision was made not to include an API, although it would be desirable to formulate an API for a future amendment to POSIX.7.2.

The second issue was a discussion on whether the standard should address error recovery and software installation rollback. The lack of such functionality was a source of a variety of objections from the mock ballot. After significant work, it was agreed to specify basic functionality in this area and a specification was worked out consistent with existing practice from some of the vendors in the group.

The next major issue was the discussion of the inclusion of managed objects in the POSIX.7.2 standard. Managed objects are a creation from some of the networking management standards organizations and are consistent with an object oriented programming paradigm. Proponents indicate that including them in the standard will promote interoperability. I have long been against including them in the standard, as I feel we should concentrate on more practical issues of specifying a CLI and/or API. My viewpoint prevailed at the meeting, at which it was agreed that we will not specify formal managed objects.

The last major issue was the discussion of including functionality in the standard to support patching and updating software, another item pointed out in the mock ballot. The decision was made that the standard currently allows for these features and no specific additional functionality is required.

User and Group Administration

A major landmark was reached for the User group in that their Project Authorization Request (PAR) was approved by the Sponsor Executive Committee (SEC). This allows them to begin work on their standard, and we hope it will encourage organizations to send people to participate in this group. A lack of interested people has been a major obstacle for this group.

The small group spent most of the meeting working on getting their document into official format and reviewing the existing object definitions and task descriptions. A major purpose of this review was to identify terms that need to be included in the definitions section.

The User group indicated that they have decided not to address the *su* or *login* commands, but hope to get POSIX.2 (Shell and Utilities) or POSIX.6 (Security) groups to address these commands. There is little hope that these groups will actually undertake this work. Another possibility is to submit another PAR to address these commands as an amendment to the POSIX.2 standard and ask that the work be assigned to POSIX.7. Personally, I think this would be a good idea, and would meet the idea of standardizing existing practice better than any other work currently in POSIX.7.

Report on POSIX.7.3: User Group Administration

Bernard Kinsler <kinsler@rsvl.unisys.com> reports on the July 12–16, 1993 meeting in Denver, CO:

[Disclaimer: This is a personal view of SysAdmin and has not been written in consultation with the other members of POSIX.7.]

Some Background & History

Some time ago, the Systems Administration group decided that it was neither feasible, nor desirable to try and produce a monolithic SysAdmin standard. It would weigh about 300 pounds, be balloted in about 2093 and not be approved. The area is just too broad to be covered in one standard and too broad to achieve any kind of consensus.

So it was decided to form small groups to look at particular aspects of SysAdmin. Each group could pick an area that needed standardization (because of industry demand), that could be standardized (because of sufficient existing practice) and that could be done in a realistic time scale (because of a narrow scope). The timescale is important because ISO insists on it.

Present State

One of the IEEE criteria for approving a working group is that there should be a critical mass of attendees representing a broad industry background. With the current level of attendance at POSIX.7, three small groups are possible.

- **POSIX.7.1 Printing Administration**
  Printing has just completed the formal ballot period. This is a large and complex standard, so many comments are expected, and it is expected to take quite a while to resolve them. The current draft is POSIX.7.1/D7. If you want to help in the ballot resolution process, you will be welcome at the next meeting.
- **POSIX.7.2 Software Installation**
  Formal ballot entry is planned for around April 1994. Many of the comments from the mock (informal) ballot related to the style and format of the document. In response, the group is doing some very heavy editing of the draft in preparing for the full ballot. One of the high priority tasks is to make the document more readable – a novel approach for a draft standard. Because of the large amount of restructuring being done, no significant functional changes are intended to be made. The current draft is POSIX.7.2/D8. But there is still a lot of work left, and again more volunteers can be used.
- **POSIX.7.3 User/Group Management**
  The User/Group Management small group started at the Chicago, July 1991 meeting. We are now on the second version of the first draft.

The New Project

At the Denver meeting, the Sponsor Executive Committee (SEC) approved the Project Authorization Request for POSIX.7.3 to address User and Group Management.

Why has it taken so long to get this short a distance?

The initial problem was in defining the scope of User Management. The concept of users is central to SysAdmin. Without users, there is very little that can usefully be managed. But users can do many things and touch many areas. For example, User Management must cover security aspects, but security can not be defined by User Management. There are many tasks that an administrator does in managing the needs of users. Do all of these tasks fall in the scope of User Management?

Some examples:

- Managing mail
- Quota administration
- Monitoring and auditing users
- Accounting (as in billing for resources used)
- Software license usage

Early on, we did get carried away in some of the things that we thought were possible, and we could rightly have been accused of invention. This freewheeling was exacerbated by the lack of good reference documents that would have given a focus to the group. We had to cover distributed user management, in which existing practice seemed to be sadly lacking.

What has changed?

Nearly everything! In the last two meetings, we have concentrated on two reference documents:

- **Santa Cruz Operation's User Management**
  Provides a CLI (Command Line Interface) for the management of system defaults, user templates, and extended security

- **USL SVID3**
  Provides a CLI for the management of the user database, home directories and the group database (This interface is in wide use and incorporates a large body of existing practice.)

Using these documents, we have been able to define our efforts narrowly and base them solidly on existing practice. We can formalize this to cover a distributed heterogeneous environment. With the help of the security group, we can cover the extended security requirements. As reassurance of our new narrow scope, none of the items listed above are included in User/Group Management.

Having defined what we want to do and armed with good reference documents, we rewrote the Project Authorization Request (PAR) at this last meeting. Having an approved PAR is a prerequisite to producing a ballotable standard.

The stated scope and purpose of the new group is this:

> User administration includes, but is not limited to, tasks such as the creation and maintenance of user accounts and groups in both single systems and heterogeneous distributed environments. POSIX.7 is committed in this standard to provide the distributed management for a POSIX.1 and POSIX.2 conformant system.
>
> Although POSIX.1 describes a user database, a group database, and the concept of a user's home directory, utilities to manage these entities are not described by any current standard. The User Administration working group takes as its mission the management of these entities, as described in POSIX.1 and POSIX.2, as well as the management of optional extensions such as an account password.

Current State of User/Group Management

There is an initial draft, which we spent all of the last meeting putting into the official format and in refining existing text. The following commands are defined,

useradd	*groupadd*	Add a user/group
userdel	*groupdel*	Remove a user/group
usermod	*groupmod*	Modify a user/group
userls	*groupls*	List a user/group
passwd		Change user/group password

Managed objects are defined using GDMO (*Guidelines for the Definition of Managed Objects*, ISO 10165-4). The intention here is to use GDMO as a convenient way to give a formal definition of the objects and attributes, but there is no intention to require an object oriented implementation. To aid readability, we have omitted all those parts of GDMO which are only needed for implementation. This includes NAME BINDINGS, REGISTERED AS, and so on. We have simplified the BEHAVIOUR syntax, again for readability. Don't try and compile these definitions! Equally, do not be put off by GDMO. These definitions should be quite readable even if you have never seen GDMO.

There are only four managed-object classes defined, which are organized as a collection of PACKAGES. This is a convenient way to separate mandatory and optional attributes. All mandatory attributes for a class are grouped together in a package. Optional attributes are grouped in a logical manner, for example, all the basic password attributes for the user class are in one package. An added advantage is in helping to define the optional extras. If an optional package is implemented, all of it must be implemented.

Support Needed

The User/Group Management group is actively seeking support. If you would like to join this group by attending meetings, you are welcome to. If you have an interest in this area and feel that you can contribute in any way, then this is your chance to influence the standard.

If you can not attend, but feel that standardization of this topic is worthwhile, then a letter of support from your company would be helpful. The next meeting will be at the Bethesda Marriott Hotel, in Bethesda Maryland, October 18 – 22, 1993. For more meeting details, contact:

Standards and Technical Activities Assistant
IEEE Computer Society
1730 Massachusetts Ave. NW
Washington, DC 20036-1903
+1 (202))371-0101
FAX (202) 728-9614

If you would like any more information on how to attend (or on anything else related to POSIX.7. I am the Secretary), contact me. I would prefer email.

Report on Automated Testing BOF

Kathy Liburdy <liburdy@hubcap.clemson.edu> reports on the July 12–16, 1993 in Denver, CO:

The Automated Testing (AT) BOF was created as a forum for the discussion of alternative and progressive approaches to conformance testing. While many members of POSIX are interested in testing issues, they may not be able to (or want to) devote the majority of their efforts to testing. By providing brief reports on current research in conformance testing, concise updates on related standards groups, and demonstrations of testing systems, the AT BOF extends the opportunity to all members to participate in this important aspect of the standards development process.

The AT BOF meets quarterly during meetings. To facilitate communication between meetings, a newsletter, called OATS, and a mailing list have been created for the group. There are three issues to date of the newsletter *Automated Testing in Open Systems.* Articles describing existing testing efforts as well as commentaries on testing related issues are welcome for future issues. Submissions, as well as requests for copies of existing issues, may be sent to *<liburdy@hubcap.clemson.-edu>*.The mailing list *oats* can be joined by sending a message to *<oats-request@stdsbbs.ieee.org>*. The first AT BOF was held in New Orleans, January 1993. Clemson University demonstrated their automated testing system (CATS), which was funded by the U.S. Navy. CATS is an interactive system that has proven useful in the design of interfaces as well as the testing of implementations for conformance. A single user version of the CATS facility is available as a demonstration copy by anonymous *ftp* from the site *ftp.eng.clemson.edu* in the directory *pub/cats*.

Shane McCarron introduced the Assertion Definition Language (ADL) Project sponsored by the Japanese Ministry of International Trade and Industry. This project is an effort to develop technology that will automate the process of test suite development, provide improvements in the availability of test suites for Open System standards, and improve the overall quality of these standards.

The second AT BOF was held at the April POSIX meeting in Irvine. DEC gave a presentation and demonstration of their testing technology, which is based on extended finite state machines (EFSMs). As part of an advanced development project at DEC, the concepts of EFSMs have been applied to a variety of test problems and can achieve both exhaustive and thorough testing through automatic test generation. Jim Leathrum of Clemson University introduced a formal, yet English-based language for writing test assertions in the CATS environment. This assertion language was designed to have intuitive, yet precise, semantics with a natural syntax. Examples of assertions developed in this assertion language were presented.

The group then discussed the potential impact of the recent decision to rescind the requirement that all standards be accompanied by POSIX.3 style test assertions prior to passing ballot. There was a general consensus that there are many challenging issues that remain to be addressed in the area of conformance testing and that the AT BOF may serve as an arena to explore potential solutions to existing and emerging problems.

At the Denver POSIX meeting in July, presentations were given by Lowell Johnson, Shane McCarron, and Rob Savoye. Lowell Johnson presented the status of the P2003 working group. The P2003 working group continued working on Draft 0.2, updating sections on profile testing, LIS, and general assertion writing. The group also determined that the scope of P2003 would be to target the standard for PASC, but write it using general terminology and be applicable for specifications from ISO to a private company's specification. Shane McCarron gave an update on the Chimera project and presented results of initial research into existing technologies for automated testing. Shane reported that the Chimera project had found a base technology at Sun Microsystems that was mature and changeable. During the next nine months, Sun will take the base research, which uses C++, and modify it to meet the remaining requirements. The research is expected to be completed by March 1994. A proof of concept is planned using the existing C++ technology to describe tests against 1003.1 and 1003.2. A complete test suite is planned for development in early 1994.

Rob Savoye presented developments made by Cygnus in the field of automated software test generation. DejaGnu is a framework used to set up automatic testing, particularly regression testing. Test suites built by DejaGnu are available at no cost via direct *ftp*. Cygnus is currently developing a 1003.1 test suite. A general description of DejaGnu and the DejaGnu 1.0 Release information was recently posted to the *oats* mailing list. The meeting concluded with a discussion of future directions for the AT BOF. The Steering Committee on Conformance Testing (SCCT) had expressed an interest in investigating alternative testing methods in a forum similar to that of the AT BOF. The issue of whether or not the AT BOF should formally become part of the SCCT was raised. Participants expressed their desire to continue the AT BOF but not officially link it to any standard working group.

The next meeting of the AT BOF is scheduled for October 1993. The date and time will be posted to the oats mailing list. The current agenda includes an update on the ADL Project by Shane McCarron and a presentation of the CATS approach to testing profiles by Kathy Liburdy. Suggestions for additional agenda items are welcome and may be posted to the *oats* mailing list.

Unix Tricks & Traps: form letters with MH

I cope with my email using MH, the Rand Message Handling system. MH is freeware, and I reckon it's God's own mailer, because if you get as much mail as God, you need an industrial-strength mailer.

One of the things I like about MH is that it stores each message as a separate file; mail folders are directories. So the third message in folder *File-AUUG* is most likely called *Mail/File-AUUG/3* (relative to my home directory). MH's commands are separate shell commands, such as `inc` to incorporate new mail, `repl` to reply, and `comp` to compose a message.

This means I can use all the facilities of Unix and the shell to manipulate my email.† Not only can I, for example, rewrite `inc` to automatically sort my incoming mail into several different mailboxes depending on sender and subject, but I can extend the power of my mail environment by adding new commands, like the following shell script, which I call `formletnum`:

```
#!/bin/sh
if [ $# != 1 ]; then
    echo "USAGE: $0 formletter-number"
    exit 1
fi
exec comp -form File-Formletters/$1
#
# end
```

So

```
formletnum 3
```

composes a letter using message 3 in folder *File-Formletters* as a template.

I can also choose my template by subject, using the following script, which I call `formlet`:

```
#!/bin/sh
if [ $# -lt 1 ]; then
    echo "USAGE: $0 subject"
    exit 1
fi
exec comp -form File-Formletters/'pick +File-Formletters \
    -subject "$*" | head -1'
```

If I say, for example

```
formlet new account
```

this uses MH's `pick` command, piped to `head(1)`, to find the message number of the first message in folder *File-Formletters* with "new account" in its subject line. The message number is then passed to MH's `comp` command, as in the previous script.

I also have one form letter template that is not in any folder. It is in the top level of my *Mail* directory, and the following C-shell alias lets me compose a message using it:

```
alias poem    comp -form poemtemplate
```

So when I feel like sending a newly-written poem to my friends in a certain international on-line poetry workshop, all I have to do is say `poem`, type in the lines, and MH and `csh`, working with several other Unix tools, take care of the rest.

By the way, I'm supposed to be the *editor* of this column. I'll gratefully accept your contributions, which will be replied to (not automatically!) and duly filed in the appropriate MH folder until next AUUGN-time.

Janet Jackson <jackson@cwr.uwa.edu.au>

† When I'm using X, I also use `xmh`, a wrapper that calls the MH commands, to graphically manage my labyrinth of mail folders. I also know of two visual interfaces to MH that run on ordinary terminals: `vmh`, a `vi`-like interface, and `plum`, Tom Christiansen's interface written in `perl`.

AUUG MANAGEMENT COMMITTEE
SUMMARY OF MINUTES OF MEETING 5th November 1993

Present: Phil McCrea, Chris Maltby, Glenn Huxtable, Frank Crawford, Michael Paddon, Greg Birnie.

Apologies: Chris Maltby, Chris Schoettle, Peter Wishart.

Guests: Liz Fraumann, Wael Foda, Lachie Hill, Piers Lauder.

1. AUUG93

86 individual and 12 institutional new members were directly attributed to the conference. The majority of these were a direct result of the membership form in the registration brochure.

Overall the conference & exhibition appears to have a profit of between $50K & $60K. Numbers to be confirmed.

LH recommended we promote tutorials separately as they are a large portion of income factor. GH suggested a tutorial road show outside of the summer conference series.

AUUG '93 PR was extremely good. Media showed interest and was keen to meet the speakers. Both the build up and follow up was good. Stoll provided good mechanism to gain interest. Strength of programme was exciting and offered good level of corporate participation.

PL (AUUG93 program chair) gave an overview of suggestions for '94. He will pass on directly to the '94 committee. He covered: Beverages at breaks, cross-over time shortened with additional time going to lunch and breaks to be longer. Hire staff for terminal room, booth. Managing things better, i.e. dedicated to task and managing it. Tweak dinner and continue with the change, also need to change expectations.

AUUG95 and forward venues were discussed including Brisbane as an option for 1995 or 1996.

2. President's Report

A new organisation, TAOSS, has been formed with an oversight board for open systems activities. AUUG should be involved. PM to discuss with PW.

3. Treasurer's Report

FC reported that we had a minor difficulty with cash flow. However, the bulk of expenses are over from the conference. We are expecting $16k from connect.com for AARNet connection fees. FC suggested we could use a dedicated resource to handle connection and network items. Current status is: $67k in cash management, $30k and $34k in other accounts. PM thanked FC for his excellent job with the conference.

4. Australian Articles

Articles for Australian - general requests were sent out to chapters and AUUG exec. It was suggested to put an ad in AUUGN requesting papers. EF to contact Brad Jelfs and request an article from him. MP suggested direct contact with the chapters would help and asked the executives to contact their chapters.

5. Summer Conferences

Summer Conference Series - Final dates:

tab(;); l r l. Feb.;2-5;Melbourne ;10-11;Darwin ;14-16;Canberra ;21-22;Perth ;28-1;Adelaide Mar.;3-

Kirk McKusick will be doing a roadshow tutorial with all of the summer conferences.

6. New Committee Member

Chris Schoettle has indicated he cannot continue on the committee due to unforeseen work commitments. A new committee member needs to be co-opted.

7. Other Business

The UniForum NZ conference is in May. EF will check on group rate to attend. EF to check on all Pacific-rim conferences and post.

EF to do advertisement for AUUGN of promotionals and documents left. T-shirts will sell @$10.00 each, Proceedings $50.00, '92 Proceedings sell for $25.00@ overseas, postage will be charged. In addition, one copy of all will be sent to Adrian Booth as AUUG historian.

Cheque signing - it was moved, seconded, and passed to raise the limit required for dual signatures to $1k. Signatories will be: GH, PM, FC, CM.

AUUG collateral - update of handbook and supplemental directory will be done.

Membership Drive - EF to do research on costing for direct mail campaign to MIS 2000 list etc. and provide budget. Implement in February time frame to work with summer conference series.

3 Year Business Plan - PM stated was logical and overall had no problems with it. Should be doing all that was mentioned. Time was running out, so asked committed to review it via e-mail.

8. Next Meetings

The next meeting will be on Fri 10th December 1993 at Softway.

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.

MEMBERSHIP APPLICATION

INDIVIDUAL

To apply for AUUG membership, complete this form and return it with payment in Australian Dollars to:

REPLY PAID 66, AUUG MEMBERSHIP SECRETARY,
P.O. BOX 366, KENSINGTON, NSW 2033, AUSTRALIA
Tel: +61 2 361-5994 Fax: +61 2 332-4066

Tick this box if you wish your name withheld from mailing lists made available to vendors. □

NOTE: Please do not send purchase orders - perhaps your purchasing department will consider this form to be an invoice. Foreign applicants please send a bank draft drawn on an Australian bank, and remember to select either surface or air mail.

INDIVIDUAL OR STUDENT MEMBERSHIP:

I, ______________________________ do hereby apply for:

Renewal/New membership of AUUG	□ $ 90.00
Renewal/New Student membership	□ $ 25.00 (please complete certification portion)
International surface mail	□ $ 20.00
International air mail	□ $ 60.00
UniForum affiliate membership	□ $ 20.00

TOTAL REMITTED: AUD$_______ (Cheque, or money order or credit card)

I agree that this membership will be subject to the rules and by-laws of AUUG as in force from time to time, and that this membership will run from time of joining/renewal until the end of the calendar or financial year.

____
Signature

____
Date

LOCAL CHAPTER DESIGNATE:

You can participate in the activities of a local AUUG Chapter. Part of your fee will be given to the chapter to support those activities. By default a regional chapter will be selected for you. If you would rather nominate a chapter, please specify here ______________ (indicate NONE for no chapter).

TO BETTER SERVE YOU, PLEASE PRINT YOUR CONTACT INFORMATION:

Name/Contact: ______________________________

Position/Title: ______________________________

Company: ______________________________

Address: ______________________________

______________________ Postcode__________

Tel: BH__________________ AH __________________

Fax: BH__________________ AH __________________

email address:______________________________

Over for Institutional Membership

STUDENT MEMBER CERTIFICATION: *(to be completed by a member of the academic staff)*

I, ______________________ certify that
(administrator)

______________________ is a full time student at
(name)

______________________ and is expected to
(institution)

graduate approximately ______________.
(date)

____
Signature

______________________ ______________
Title *Date*

Please charge $__________ to my

□ Bankcard, □ Visa, □ Mastercard

Account number:______________________________

Expiry date:______________________________

Name on card:______________________________

Signature:______________________________

AUUG Secretariat Use

*Chq: bank*______________ *bsb* ______________

a/c ______________ *#* ______________

Date: ______________ *$* ______________

Initial: ______________

Date processed: ______________________________

Membership # ______________________________

AUUG Inc. as a user group, exists to provide UNIX® and open systems users with relevant and practical information, services, and education through cooperation among users.

MEMBERSHIP APPLICATION

INSTITUTIONAL

To apply for AUUG membership, complete this form and return it with payment in Australian Dollars to:

REPLY PAID 66, AUUG MEMBERSHIP SECRETARY,
P.O. BOX 366, KENSINGTON, NSW 2033, AUSTRALIA
Tel: +61 2 361-5994 Fax: +61 2 332-4066

Tick this box if you wish your name withheld from mailing lists made available to vendors. □

NOTE: Please do not send purchase orders - perhaps your purchasing department will consider this form to be an invoice. Foreign applicants please send a bank draft drawn on an Australian bank, and remember to select either surface or air mail.

We, ______________________________
(Company Name)

do hereby apply for:

Renewal/New* Inst. membership of AUUG	□ $350.00
International surface mail	□ $ 40.00
International air mail	□ $120.00

TOTAL REMITTED AUD$________
(Cheque, money order, or credit card)

I/We agree that this membership will be subject to the rules and by-laws of AUUG as in force from time to time, and that this membership will run from time of joining/renewal until the end of the calendar or financial year.

I/We understand that I/we will receive two copies of the AUUG newsletter, and may send two representatives to AUUG sponsored events at member rates, though I/we will have only one vote in AUUG elections, and other ballots as required.

Signed______________________ Date ________

Title ______________________________

INSTITUTIONAL MEMBER DETAILS:
To be completed by institutional members only.

Following are our specified contacts. The primary contact holds the full member voting rights. The two designated reps will also be given membership rates to AUUG activities including chapter activities. By default a regional chapter will be selected for you. If you would rather nominate a chapter, please specify in space provided (indicate NONE for no chapter). *(Please print clearly or type)*

Primary Contact______________________

Position/Title ______________________

Address ______________________

______________________ Postcode__________

Bus. Tel: ______________ Bus. Fax:______________

e-mail address ______________________

Local Chapter Pref. ______________________

1st Rep.______________________

Position/Title ______________________

Address ______________________

____

Bus. Tel: ______________ Bus. Fax:______________

e-mail Address ______________________

Local Chapter Pref. ______________________

2nd Rep.______________________

Position/Title ______________________

Address ______________________

____

Bus. Tel: ______________ Bus. Fax:______________

e-mail address ______________________

Local Chapter Pref. ______________________

Please charge $______________ to my

□ Bankcard, □ Visa, □ Mastercard

Account number:______________________

Expiry date:______________________

Name on card:______________________

Signature:______________________

AUUG Secretariat Use

*Chq: bank*______________ *bsb* ______________
a/c ______________ *#* ______________
*Date:*______________ *$* ______________
*Initial:*______________
*Date processed:*______________________
*Membership #*______________________

AUUG Inc. as a user group, exists to provide UNIX® and open systems users with relevant and practical information, services, and education through cooperation among users.

AUUG Incorporated
Application for Newsletter Subscription
AUUG Inc.

Non members who wish to apply for a subscription to the Australian UNIX systems User Group Newsletter, or members who desire additional subscriptions, should complete this form and return it to:

AUUG Membership Secretary
PO Box 366
Kensington NSW 2033
Australia

- Please don't send purchase orders — perhaps your purchasing department will consider this form to be an invoice.
- Foreign applicants please send a bank draft drawn on an Australian bank, or credit card authorisation, and remember to select either surface or air mail.
- Use multiple copies of this form if copies of AUUGN are to be dispatched to differing addresses.

This form is valid only until 31st May, 1994

Please *enter / renew* my subscription for the Australian UNIX systems User Group Newsletter, as follows:

Name: .... Phone: .... (bh)

Address: .... (ah)

.... Net Address: ....

....

.... *Write "Unchanged" if address has not altered and this is a renewal.*

....

For each copy requested, I enclose:

□ Subscription to AUUGN	$ 90.00
□ International Surface Mail	$ 20.00
□ International Air Mail	$ 60.00

Copies requested (to above address) ________

Total remitted AUD$______
(cheque, money order, credit card)

□ Tick this box if you wish your name & address withheld from mailing lists made available to vendors.

Please charge $_____ to my □ Bankcard □ Visa □ Mastercard.

Account number: _. Expiry date: __/__.

Name on card: ______________________________ Signed: ____________________

Office use only:

Chq: bank ________ *bsb* ___-___ *a/c* ______________ # ______________

Date: __/__/__ *$* *CC type* ___ *V#* __________

Who: __________ *Subscr#* ________

AUUG

Notification of Change of Address

AUUG Inc.

If you have changed your mailing address, please complete this form, and return it to:

AUUG Membership Secretary Fax: (02) 332 4066
P O Box 366
Kensington NSW 2033
Australia

Please allow at least 4 weeks for the change of address to take effect.

Old address (or attach a mailing label)

Name: ....

Address: ....

....

....

....

....

Phone: .... (bh)

.... (ah)

Net Address: ....

New address (leave unaltered details blank)

Name: ....

Address: ....

....

....

....

....

Phone: .... (bh)

.... (ah)

Net Address: ....

Office use only:

Date: __/__/__

Who: ____________

Memb# _________

Calendar of Events †

1993

Oct	18-22	IEEE 1003, Bethesda, MD
	25-29	INTEROP '93 Europe, Paris, France
	26-28	UniForum Common Desktop Environment UNIX Developers Conf. San Jose, CA
Nov	1- 5 *	LISA VII, Monterey, CA
	3 -5 *	NOSSDAV Workshop, Lancaster, UK
	29 -	FedUNIX, Washington, DC
Dec	3	" "
	4-10	DECUS, San Francisco, CA
	7-9	Sun User Group, San Jose, CA
	8-10	JUS UNIX Fair, Japan

1994

Jan	10-14	IEEE 1003, Irvine, CA
	17-21 *	USENIX, San Francisco, CA
Mar	21-25	UniForum, San Francisco, CA
Apr	11-14 *	C++, Cambridge, MA
	25-28 *	UNIX Applications Development Toronto, Canada
Apr	5-9 *	SANS III, Arlington, VA
	18-22	IEEE 1003, Lake Tahoe, CA
May	2- 6	Networld+INTEROP 94, Las Vegas, NV
	7-13	DECUS, New Orleans, LA
Jun	6-10 *	USENIX, Boston, MA
	"	Networld+INTEROP 94, Berlin
Jul	11-15	IEEE 1003
Sept	12-14	Networld+INTEROP 94, Atlanta, GA
	19-23 *	LISA VIII, San Diego, CA
Oct	23-27	ACM OOPSLA, Portland, OR
	26-28 *	Very High Level Languages Santa Fe, NM
Nov	12-18	DECUS, Anaheim, CA

1995

Jan	16-20 *	USENIX, New Orleans, LA
Feb	21-23	UniForum, Dallas, TX
May	13-19	DECUS, Atlanta, GA
Jun	19-23 *	USENIX, San Francisco, CA
Nov	2- 8	DECUS, San Francisco, CA

1996

Jan	22-26 *	USENIX, San Diego, CA
Mar	12-14	UniForum, San Francisco, CA
May	18-24	DECUS, Orlando, FL
Nov	16-22	DECUS, Anaheim, CA

This is a combined calendar of planned conferences, symposia, and standards meetings related to the UNIX operating system. If you have a UNIX-related event that you wish to publicize, please contact <*login@usenix.org*>. Please provide your information in the same format as above.

* = events sponsored by the USENIX Association.

ACM: Association for Computing Machinery
AUUG: Australian UNIX Users Group
DECUS: Digital Equipment Computer Users Society
EurOpen: European Forum for Open Systems

FedUNIX: Council of Advanced Computing Systems Technologists in Government
FIRST: Forum of Incidence Response & Security Team
GURU: Roumanian UNIX User Group
IEEE: Institute of Electrical and Electronics Engineers

IETF: Internet Engineering Task Force
INET: Internet Society
Interex: Intl. Association of Hewlett-Packard Comp.Users
JUS: Japan UNIX Society

LISA: USENIX Systems Administration Conference
NOSSDAV: Network and Operating System Support for Digital Audio and Video
OOPSLA: Object - oriented Programming Systems, Languages, and Applications
SAGE: System Administrators' Guild

SANS: System Administration, Networking & Security
SEDMS: Symposium on Experiences with Distributed and Multiprocessor Systems
UKUUG: United Kingdom UNIX Systems Users Group
UniForum: International Association of UNIX and Open Systems Professionals

† This is a re-print from ;login, the USENIX Association Newsletter, Volume 18 Number 5

X/Open

Committed to Open Systems

New Publications

New Publications

New Publications

C206 Management Protocol Profiles (XMPP)
34p. $72.00
This Management Protocol Profiles CAE Specification addresses communications using the CMIP and SNMP management protocols. RPC-based mechanisms will be addressed in future documents. This document supersedes the XMPP Preliminary Specification, which was published in January 1992.

C303 ACSE/Presentation Services API (XAP) 254p $126.00
XAP is an Application Programming Interface to the connection-oriented services of the Presentation Layer of the OSI protocol stack, including access to the ACSE application service element from the Application Layer. X/Open has defined this API as an interface to support portable implementations of application-specific OSI services and non-OSI applications. This specification describes the XAP API and defines the functions and data structures which it provides for use by applications.

C307 Data Management: SQL Remote Database Access 76p $108.00
This specification relies on the ISO/IEC RDA SQL standard, which specifies a message format for remote communication of SQL database language statements (query and update) to a remote database. This specification defines uses of the message fields and other implementation information such as sequencing and optional features. It shows how SQL statements map to the Remote Database Access (RDA) protocol. This document is the result of a joint collaborative effort between the X/Open data Management Working Group and the SQL Access Group

C318 X /Open Transport Interface (XTI), Version 2 264p. $126.00
The X/Open Transport Interface (XTI) specifies a transport service interface that is independent of any specific transport provider. It is concerned primarily with the ISO Transport Service Definition (connection-oriented or connectionless). However, it may be adapted for use over other types of provider. In particular, this specification includes Appendices showing how XTI maps to TCP and UDP, NETBIOS, minimum OSI functionality (mOSI), and SNA. This Version 2 of the XTI Specification replaces the previous version published in January 1992. The main body of the specification remains unchanged from that previous version, the main differences being the addition of Appendices covering mOSI and SNA.

G207 Systems Management: Reference Model 68p $108.00
This guide is one of several documents within X/Open's Systems Management programme (XSM), the primary aim of which is to promote development of management software that allows an administrator of a distributed system to manage a network of heterogeneous systems as a single logical system. This Guide provides an architectural overview of the Systems Management model, and identifies the various components of the model. It employs object-oriented techniques. This is a high-level model which encompasses both OMG-compliant object-oriented technology and the OSI interaction, and indicates an approach for coexistence of both models.

G302 Systems Management: Managed Object Guide (XMOG) 82p $108.00
This guide is one of several documents within X/Open's Systems Management programme (XSM), the primary aim of which is to promote development of management software that allows an administrator of a distributed system to manage a network of heterogeneous systems as a single logical system. This Guide introduces the essential nature of managed objects, and the necessary framework for defining them. It discusses the managed object definition and development process, and explores the issues involved in registering them. It also discusses conformance testing.

G307 Distributed TP: Reference Model, Version 2 44p. $120.00
This document provides a functional description of the X/Open Distributed Transaction Processing (DTP) model, a software architecture that allows multiple application programs to share resources provided by multiple resource managers, and allows their work to be coordinated into global transactions. It describes the use of the DTP model within the X/Open Common Applications Environment (CAS) and is a prerequisite to other present and emerging X/Open documents that address DTP. This document supersedes the previous version of the Reference Model published In 1991 (Doc. No. G120). It has been updated to take account of DTP interfaces developed by X/Open, particularly with regard to how Communication Resource Managers (CRMS) fit within this X/Open DTP model.

P303 Data Management: SQL Call Level Interface (CLI) 210p. $126.00
This document defines the SQL Call Level interface (CLI). It provides application programmers with an application programming interface (API) for database access. Both C and COBOL bindings are given. Readers should be familiar with the X/Open CAE Specification, Structured Query Language (SOL), Doc. No. C201. This document is the result of a joint collaborative effort between the X/Open Data Management Working Group and the SQL Access Group.

P312 X/Open DCE: Remote Procedure Call 666p. $162.00
This document specifies Remote Procedure Call (RPC) services, interface, protocols, encoding rules and the Interface Definition Language (IDL). The purpose of this document is to provide a portability guide for RPC application programs and a conformance specification for RPC Implementations. This document includes text excerpted and/or derived from the Open Software Foundation Application Environment Specification for Distributed Computing (AES/DC) with the permission of OSF.

Q931 X/Open Technical Programme $270.00
The X/Open Technical Programme is a binder with seperately bound parts that describes the Technical Programmes to be undertaken by X/Open over the next three years. Part 1 provides a Technical Roadmap. Parts 2 through 12 each describe a major technical programme area. It is primarily intended for users interested in implementing an open systems strategy and in planning the development or procurement of open systems solutions; for vendors (both software and systems vendors) interested in pursuing an open systems product strategy; and for systems integrators interested in providing open systems integration services and facilities.

For the full publications list, or to place an order, contact Tony Blackmore:
Tel: 03 879 7412, Fax: 03 879 7570 or Email 100036.1124@compuserve.com