

Australian UNIX systems User Group Newsletter

AUUGN

Volume 11, Number 4

December 1990

AUUG 1990 AGM Reports

The Australian UNIX* systems User Group Newsletter

Volume 11 Number 4

December 1990

CONTENTS

AUUG General Information	3
Editorial	4
AUUG Institutional Members	6
AUUG Incorporated 1990 Annual General Meeting.	9
AUUG President's Report.	11
Secretary's Report 1989/90	12
Treasurer's/Auditor's Report 1990	15
AUUG Election Report 1990	24
From the <i>;login:</i> Newsletter - Volume 15 Number 3	25
Book Review: Life With UNIX - A Guide For Everyone	25
Report on ISO/IEEE JTC1/SC22/WG15 Rapporteur Group on Internationalization Meeting	26
International Standardization	30
An Update on UNIX and C Standards Activity	34
From the <i>EUUG Newsletter</i> - Volume 10 Number 2	64
Call Doc Strange	64
AT&T Column.	69
!%@:: A Directory to Electronic Mail Addressing and Networks, Second Edition, 1990	72
Puzzle Corner	73
A New Data Encoding Scheme	75
Rules of AUUG Incorporated	77
AUUGN Back Issues.	86
AUUG Membership Categories	88

Copyright © 1990 AUUG Incorporated. All rights reserved.

AUUGN is the journal of the Australian UNIX* systems User Group (AUUG Incorporated).

Copying without fee is permitted provided that copies are made without modification, and are not made or distributed for commercial advantage. Credit to AUUGN and the author must be given. Abstracting with credit is permitted. No other reproduction is permitted without prior written consent of AUUG Incorporated.

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

AUUG General Information

Memberships and Subscriptions

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

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

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

Phone: (02) 361 5994
Fax: (02) 332 4066

General Correspondence

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

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

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

AUUG Executive

President **Greg Rose**

greg@softway.sw.oz.au
Softway Pty Ltd
New South Wales

Vice President

Pat Duffy

pzd30@juts.ccc.amdahl.com
Amdahl Australia Pty Ltd
New South Wales

Secretary

Peter Barnes

pdb@uqcspe.cs.uq.oz.au
Computer Science
University of Queensland

Treasurer

Michael Tuke

mjt@anl.oz.au
ANL Limited
Victoria

Committee

Frank Crawford

frank@teti.qhtours.oz.au
Q.H. Tours Pty Ltd
New South Wales

Andrew Gollan

adjg@softway.sw.oz.au
Softway Pty Ltd
New South Wales

Chris Maltby

chris@softway.sw.oz.au
Softway Pty Ltd
New South Wales

Scott Merrilees

sm@bhpese.oz.au
BHP Information Technology
New South Wales

Stephen Prince

sp@labtam.labtam.oz.au
Chancery Lane Computer Services Pty Ltd
Victoria

Next AUUG Meeting

The AUUG'91 Conference and Exhibition will be held from the 24th to the 27th of September, 1991, at Darling Harbour, Sydney. The AGM of AUUG Inc. will be held during the conference.

The AUUG'92 Conference and Exhibition will be held from the 8th to the 11th of September, 1992, at the World Congress Centre, Melbourne.

Editorial

Due to the fact that I have been unable to devote as much time to the job of editing AUUG as I would have liked, and also due to the fact that I have not received many articles, this will be the last issue of AUUGN for 1990.

This is the administrative issue of the year. In this issue you will find the minutes of the AUUG Incorporated Annual General Meeting, as well as reports from the previous year's office bearers that were presented at the AGM. These minutes will be confirmed at the next AGM, to be held during the AUUG'91 at Darling Harbour, Sydney, from September 24th to 27th, 1991.

The AGM is the best opportunity there is for the membership to influence the course of the committee. The current committee is committed to improving the benefits of membership of AUUG, but it needs to know what it is that the membership wants. The AGM gives you the chance to tell the committee directly, although there are other means available: have you considered writing a letter to AUUGN?

Also in this issue you will find the up-to-date Rules of AUUG Incorporated. These rules were changed by the referendum held last May, and they are published here for the information of members.

One final thing; I would like to extend thanks on behalf of myself and AUUG Incorporated to Michael Lawrence and Webster Computer Corporation. The previous editor of AUUGN, John Carey, was working at Webster, and after he left Webster and I took over editorship it has taken nearly two years to correct the address of the AUUGN Editor in various international mailing lists. In this time Michael has forwarded mail onto me, for which I thank him greatly.

AUUGN Correspondence

All correspondence regarding the AUUGN should be addressed to:-

David Purdue
AUUGN Editor
PO Box 366
Kensington, NSW, 2033
AUSTRALIA

Email: auugn@munnari.oz.au
Phone: +61 3 353 3913 (w)
 +61 3 813 1258 (h)
Fax: +61 3 353 2987

Contributions

This Newsletter is published approximately every two months. The deadline for contributions for the next issue is Friday the 1st of March 1991.

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

I prefer documents to be e-mailed to me, or mailed to me on a floppy disk (IBM-PC 5-1/4 inch or 720K 3-1/2 inch; or Macintosh 3-1/2 inch), and in plain text format. Hardcopy submissions should be on A4 with 30 mm left at the top and bottom so that the AUUGN footers can be pasted on to the page. Small page numbers printed in the footer area would help.

Advertising

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

Mailing Lists

For the purchase of the AUUG 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, details are printed at the end of this issue.

Acknowledgements

This Newsletter was produced with the kind assistance of and on equipment provided by the Advanced Imaging Systems department of Kodak (Australasia) Pty Ltd. I would also like to thank Labtam Australia for providing me with a network connection.

Disclaimer

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

AUUG Institutional Members

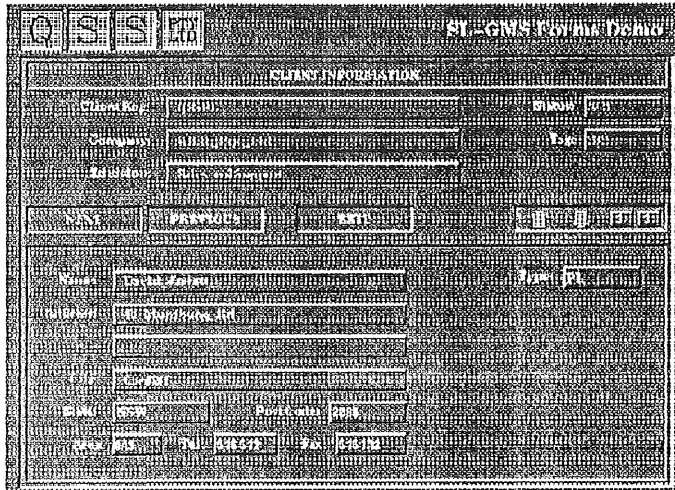
Amdahl Corporation	Department of Industrial Relations & Employment
Australian Nuclear Science and Technology Organisation	Department of Transport, Queensland
B.H.P. Information Technology Newcastle Region	Department of Treasury and Finance, Tasmania
BHP Melbourne Research Labs	Digital Equipment Corporation; (Australia) Pty. Limited
Basser Department of Computer Science	ERIN, Bureau of Flora and Fauna
Bureau of Meteorology	Epson Australia Pty Ltd
Bureau of Vocational, Further Education and Training	Exicom Australia Pty Ltd
Burns Philp Plumbing Supplies Group	Fremantle Port Authority
Centre for Information Tech & Comms	Geelong and District Water Board
Classified Computers Pty Ltd	Genasys II Pty Ltd
Co-Cam Computer Group	Hamersley Iron Pty Ltd
Commonwealth Department of Primary Industries and Energy	Harris & Sutherland Pty Ltd
Comperex (NSW) Pty Ltd	Honeywell Software Centre
Computer Software Packages	IBM Australia Limited
Crane Enfield Metals Pty Ltd	IPS Radio and Space Services
Data General Australia Pty Ltd	Information Systems Branch
Deakin University	Kodak (Australasia) Pty Ltd
	L. M. Ericsson Pty Ltd
	Macquarie University

AUUG Institutional Members

OPSM	Telecom Business Services
Pact International	Telecom Information Technology Group
Port of Melbourne Authority	The Far North Queensland Electricity Board
Queensland Education Department	The Logic Group
Queensland Justice Department	The Mathematics and Computing Department - BCAE (Kelvin Grove Campus)
SEQEB	The University of Melbourne (Information Technology Services)
Shire of Eltham	The University of New South Wales
Silicon Graphics Computer Systems	The University of Wollongong
Softway Pty Ltd	Tusc Computer Systems
South Australian Institute of Technology	University College of Central Queensland
Sphere Systems Pty Ltd	University Computing Services
Stallion Technologies Pty Ltd	University of New England
Stamp Duties Office Victoria	Vicomp
Steedman Science and Engineering	Wacher Pty Ltd
Sugar Research Institute	WordPerfect Pacific P/L
Tandem Computers Pty Ltd	Yartout Pty Ltd
Tasmania Bank	
Tattersall Sweep Consultation	
Tech Pacific	
Technical Software Services P/L	

NOW

Beyond Widgets



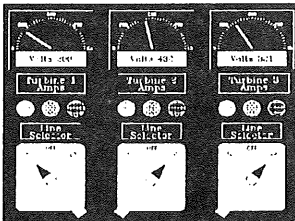
SL-GMS...the complete tool for application graphics.

The Graphical Modelling System from SL Corporation is the only complete application graphics tool for UNIX and VMS workstations, including 386-based UNIX platforms.

Draw new graphical objects · Connect easily to data sources
Animate to visualise real-time data · Use to control the application

Standard widgets are not enough:

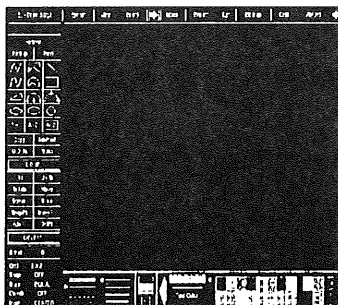
With the rise of graphics workstations has come a demand for tools that speed the development of graphics screens for applications. A bewildering array of tools has appeared to aid developers with X Windows and primary GUI styles: MOTIF, Open Look and DECwindows. Many of these tools are WYSIWYG editors limited to the creation of standard widgets such as menus, scroll boxes, sliders and buttons. Standard widgets, however, are not enough for application visualisation. Inevitably, the need arises for custom screen objects (graphs, maps, icons and other pictures) which are beyond such tools, and which are too time-consuming to create with Xlib. Developers also need a way to visualise changing data in real time.



A complete graphics tool must provide:

A Powerful Drawing Tool - The ability to create custom screen objects. Not limited to canned graph types, this flexible tool with many CAD features and an interface like familiar PC drawing tools allows you to draw whatever you need, attach dynamic behaviours and test those behaviours-all within the drawing tool.

Dynamics - The ability of screen objects to instantly reflect changes in data values, receive user input or execute callback functions. Over forty attributes such as colour or text changes, visibility on/off, percent fill, line width, rotation, movement, scaling, font style or size, zooms and window creation can be triggered by changes in data values or user input.



Complete Xt Widget Integration - SL-GMS graphics can fully incorporate Xt widgets. Screen objects created with SL-GMS fully interact with MOTIF, Open Look, DECwindows or other toolkit widgets, whether in the same or different windows.

GISMOs - Graphical Interactive Screen Management Objects are called GISMOs to distinguish them from Xt widgets. Fully interactive with Xt widgets, GISMOs can take any appearance you wish and trigger any user-defined function or external program. Created with the drawing tool, GISMOs provide developers with tremendous design flexibility.

HyperCard-like Screen Management - After screens have been created, the user must be able to button from any screen to any other screen in the application. With the screen management System (SMS) included in SL-GMS, the developer can give the user this ability without writing a line of code. SMS can bring up new screens with data sources attached and dynamics up and running.

Data Source Management - The ability to connect screen objects to data sources such as files, databases, expert systems and real-time feeds. With the Data Source Manager of SL-GMS, these connections are easily made.

Runtime Editors and Configurators - The ability of the enduser to customise or reconfigure screens to match the current environment.

Cross-Platform Portability - The ability to develop screens on any major workstation and run them on any other through simple ASCII file transfer. SL-GMS also supports PixWin, Iris GL, GKS and other non-X graphics environments. Versions such as Iris GL take advantage of special accelerator hardware and double buffering capability.

SL-GMS is widely used:

For real-time or highly interactive applications in fields such as manufacturing, process control, network management, cockpit display and financial trading.

Licensees of SL-GMS include:

ABB/Combustion Engineering
Applied Automation, Inc.
The Boeing Company
Bell Canada
Chrysler Motor Corporation
Cummins Engine Company
Eurotherm Ltd.
General Electric Company
Hewlett-Packard
Hughes Aircraft Company
Johns Hopkins University, APL
Lawrence Livermore National Labs
Lockheed
McDonnell-Douglas
Martin Marietta
Nothop Corporation
Peugeot
Westinghouse Nuclear Division
In Australia:
BHP Steel
Toshiba Power Division
Digital Equipment Corporation

Supported Workstations:

Sun (PixWin or X), DEC (VMS or ULTRIX), Silicon Graphics (X or GL), HP, Apollo, IBM, and MIPS as well as 386-based workstations running UNIX and the X windowing System.

Q.S.S. Pty. Ltd.

40 Munibung Rd., Cardiff NSW 2285.
PO Box 269, New Lambton 2305.
Tel. (049) 546524 Fax. (049) 545132



AUUG Incorporated

1990 Annual General Meeting

27th September 1990, World Congress Centre, Melbourne

These minutes are subject to amendment at the next General Meeting.

The meeting opened at 17:45 with the entire committee and a quorum of members present. The President (Greg Rose) took the Chair.

1 Apologies

None.

2 Minutes of the last meeting (10th August, 1989)

A copy of the minutes of the previous General Meeting, the 1989 AGM, was displayed for all members.

Moved Lawrie Brown, seconded Ken McDonell: That the minutes be accepted. Carried.

3 Business arising from the Minutes

None.

6 President's Report

The President, Greg Rose, presented the report as printed in AUUGN Vol 11 No 4. He thanked outgoing committee members Tim Roper and John Carey for their efforts. The meeting responded with acclamation.

Moved John Lions, seconded Ken McDonell: That the President's report be accepted. Carried.

7 Secretary's Report

The Secretary, Peter Barnes, presented the report as printed in AUUGN Vol 11 No 4.

Moved Lawrie Brown, seconded Robert Elz: That the Secretary's report be accepted. Carried.

8 Treasurer's Report

The Treasurer, Michael Tuke, presented the report as printed in AUUGN Vol 11 No 4.

Ken McDonell asked that expenditure on the secretariat be estimated on an annual basis.

Moved David Purdue, seconded Ken McDonell: That the Treasurer's report be accepted. Carried.

4 Returning Officer's Report

The Returning Officer, John O'Brien, presented the report as printed in AUUGN Vol 11 No 4.

Moved Ken McDonell, seconded Tim Segall: That the Returning Officer's report be accepted. Carried.

5 Approval of Appointments

The President explained that following the Constitutional amendments passed by the referendum, the committee had appointed Pat Duffy Vice-President (under Rule 23.(4)), Stephen Prince committee member (under Rule 23.(4)), and Scott Merrilees committee member (under Rule 23.(3)). These appointments now had to be approved by the General Meeting (under Rule 23.(5)).

Moved Robert Elz, seconded John O'Brien: That the appointment of Pat Duffy as Vice-President be approved. Carried.

Moved David Purdue, seconded Mark Andrews: That the appointment of Stephen Prince and Scott Merrilees as committee members be approved. Carried.

11 Other Business

11.1 The President reported that there had been suggestions that AUUG change its name, and invited comment.

All speakers spoke against the name change. Lawrie Brown suggested that we better publicise our charter. Andrew McCrae suggested that we raise our profile by making more press releases. Greg Kable suggested that we change our emphasis from UNIX to Open Systems. Tim Roper pointed out that several very successful Conferences and Exhibitions and associated press coverage had established the name AUUG, and that it would be counter-productive to change it.

A straw poll indicated overwhelming opposition to the change.

11.2 Scott Merrilees noted that ACSnet (or MHSnet) connection was a topic often raised by members. A survey of the meeting revealed that about 15% of members were interested in gaining access to the network.

Robert Elz gave a brief history of the Usenix/uunet public access system, and pointed out that a Melbourne company had already started a similar service.

Bob Kummerfeld noted that MHS was providing a similar service in Sydney for its customers.

A straw poll indicated general interest in the topic, and Scott Colwell and Greg Kable asked that the committee investigate means of obtaining network access for members.

The President closed the meeting at approximately 18:55.

AUUG President's Report

In the last year...

AUUG has held another successful conference. The AUUG'90 Conference and Exhibition is the first that we have held at the World Congress Centre, and I think that this is the best venue we have ever used. AUUG is sure to be back here in two year's time. AUUG'90 has also been the largest conference we have held to date, with 423 registrations and over 1100 people attending the exhibition.

1989/90 has been another year of growth for AUUG, especially in the area of institutional memberships, although we have not grown as much as figures suggest the UNIX market has grown.

In this year we established the AUUG Book Club, a working relationship with Prentice-Hall Australia that allows AUUG members to get a discount on Prentice-Hall books. PH also provide books for review in AUUGN.

In 1990 the Summer Technical Meeting program was finally got off the ground after false starts in previous years. Meetings were held in Perth, Hobart, Adelaide, Canberra and Sydney. It was originally intended to bring in a visiting speaker from overseas, but when this could not be arranged the meetings went ahead anyway with some local speakers being interchanged with several of the meetings, and all were very successful.

The Management Committee has been working to establish closer connections between AUUG and other international bodies with similar aims. In particular we have been seeking closer ties with Uniforum.

All this could not be achieved without the hard work, all volunteer labour, of many people. In particular I would like to thank two outgoing members of the Management Committee, Tim Roper and John Carey. Tim has been the AUUG Secretary for the past two years and has worked tirelessly to get the AUUG administration in order, especially in the areas of getting AUUG incorporated and trying to get secretarial assistance for the committee. John was AUUGN Editor for a number of years before taking a seat on the committee, and has been the Programme Committee Chair for AUUG'90. His efforts are one of the main reasons this conference has been so successful.

That was the good news.

On the down side, I must admit that the administration of AUUG is still a mess, despite Tim Roper's best efforts. Also, we have failed to add significant new services for members, despite the best of intentions.

Hopefully getting better.

After the recent amendments to the rules, AUUG has a larger Management Committee. This hopefully means we have a larger pool of volunteer labour to draw upon.

In the area of secretarial assistance, we are in the process of passing a lot of the day to day running of AUUG over to ACMS, who already provide management and support services for the conferences. It is hoped that AUUG can soon find a full-time employee to manage AUUG's business.

In summary, AUUG is growing and improving, but to continue in this way we need more involvement from the rank and file members, whether it be serving on the Management Committee, helping to organise a Summer Meeting, volunteering to help provide a member benefit, or just writing articles for AUUGN.

Greg Rose
AUUG President

Secretary's Report 1989/90

Peter Barnes

Secretary

AUUG Incorporated

1. The Present

1.1. Memberships

Memberships continue to grow rapidly, even though times are austere throughout the Industry. When you consider that each Institutional membership represents two members, we now have over 700 members.

As can be seen from the figures, Institutional members are growing faster than ordinary members, no doubt reflecting the continuing growth of the UNIX operating system as a commercial platform, and a greater awareness of the importance of Open Systems (and independent representation) by industry.

Another pleasing change to our membership has been the election of our first Honorary Life Member, Professor John Lions, a fitting tribute which was bestowed as soon as it was constitutionally possible.

At 21/9/90, membership numbers and approximate growth rates since the last report are:

Class	Number	% Change
Student	13	+30%
Ordinary	442	+50%
Institutional	131	+87%
Honorary Life	1	+100%

1.2. Conferences, Tutorials and Exhibition

1.2.1. AUUG '89

Following the success of AUUG '88, we optimistically predicted a growth of over 30% for AUUG '89. Growth was in fact about 25%, with about 400 registrants in total. Nevertheless we are pleased with the result in lean times. With a probable similar growth this year, we faced a watershed decision in our choice of venue - quite simply, we have outgrown everything except purpose-built centres like this.

The AUUG '89 pre-conference tutorials were very well attended, although our inexperience in running such an event was evident in places. The tutorials seem certain to be a regular event, and we have aimed for much greater polish this year.

1.2.2. Summer meetings

Following an extended gestation, the first Summer meetings were held this year right across the country, in Perth, Melbourne, Hobart, Canberra and Sydney. These meetings fill two needs: one created when we moved from bi-annual to annual conferences, and the other as a forum for more informal, technical papers. We believe these meetings were successful on both counts, and hope to extend them next year.

1.3. AUUGN

The newsletter has had an uneven time recently, as the editor has struggled with job changes and lack of access to tools. This highlights the difficulties we face attempting to service a rapidly growing membership with purely voluntary labour and loaned resources, and is a problem we must face squarely in the next year.

1.4. Membership benefits

AUUG is pleased to have been able to negotiate two discount schemes for its members: the first with Prentice-Hall, offering a 20% discount on selected titles, and the second with Addison-Wesley, offering a 20% discount on the O'Reilly's Nutshell series.

Bulk purchases of Usenix Proceedings have continued, making these important documents available to members quickly and at low cost.

This year sees the Conference Proceedings published as a serial in their own right, reflecting their quality and importance, and providing appropriate status for the papers presented.

For Institutional members, we have phased out subscriptions to Computing Systems, and instead are providing copies of the UniForum Product Directory, easily the most comprehensive and up-to-date document of its kind.

Earlier this year AUUG joined forces with the DMR Group Australia to co-sponsor a comprehensive program, titled "Strategies for Open Systems in Australia". This program is exploring the Australian marketplace, and investigating issues related to markets and implementation strategies for Open System. Summaries of results from this program will be made available to AUUG members.

AUUG is also investigating cooperation with Spectrum '90, and co-hosted an ACS Professional Development Seminar in Brisbane this year.

1.5. AUUG Chapters

We were pleased to see three local chapters created in the last year (WAUG, Sesspoole and SWiGS), although some of these bodies had existed informally before then.

Overall, growth in the local chapters has been slower than in AUUG itself, although the success of the Summer meetings will perhaps spur the chapters on to greater activity.

1.6. Growing pains

As well as electing a new committee, the last elections also passed several amendments to our Constitution, all changes which the Committee felt were necessary as AUUG grows and matures.

Other evidence of this growth is the upgrading of our office system with equipment kindly provided by Fujitsu Australia.

The last, and perhaps most important change is the establishment of a paid secretariat. This is currently underway, and we apologize for the very slow processing of membership details as we attempt to coordinate between Brisbane, Melbourne and Sydney. We believe that this marks the start of a new era for AUUG, as it had become obvious that a voluntary committee, no matter how enthusiastic, could not cope with some of the daily chores of running a large national organization. The establishment of a secretariat should allow the committee much more time to improve and extend member services.

Our address now has a phone and fax number:

AUUG Incorporated
P.O. Box 366
Kensington NSW 2033
Phone +61 2 361 5994
Fax +61 2 332 4066

2. The Future

Last year my predecessor, Tim Roper, posed a number of questions, only some of which have been answered. Perhaps the most important, that of a paid secretariat,

has been, so now is the time for members to consider what directions AUUG should be taking, and what services it should provide.

The rapid growth of commercial UNIX has meant an equally rapid change in AUUG's membership structure, and provides challenges to you and the committee.

Should we expand and improve the regional programme? How can we be both "Usenix" and "UniForum" in Australia? Do you, the members, want a greater voice in the current standards processes, or more information

about them? What additional member services and benefits would you like? Should we sink our cash reserves into a full-time secretariat?

These, and more, are issues on which we would welcome feedback and suggestions.

3. Acknowledgements

Most of the successes in this report are the result of much hard work by the previous Committee and other volunteers. I would like in particular to thank Tim Roper, who served ably as Secretary during a period of very rapid change, and whose place on the Committee will be greatly missed. I hope I can do the job half as well as he.

Another person we will miss is John Carey, whom I would like to thank for working hard and unselfishly for AUUG for more years than he probably cares to remember, both on the Committee and as AUUGN editor.

Thanks must also go to Glenn Huxtable, who masterminded the Summer meetings, a nightmare in tele-organization and long-distance cajolery.

Finally,

John Carey also deserves our thanks as Chair of the AUUG '90 Programme Committee, to whom also our collective thanks for what promises to be an excellent conference and exhibition.

Peter Barnes
AUUG Secretary

Treasurer's/Auditor's Report 1990

A.U.U.G INCORPORATION

PROFIT & LOSS STATEMENT

FOR THE PERIOD 1ST JUNE 1989

TO 31ST MAY 1990

<u>CONFERENCE AUUG 1989</u>	1990	1989
INCOME	46018.87	12550.41
<u>LESS EXPENSES</u>		
Advertising	8805.48	3579.44
Art Work		247.10
Badges		1320.00
Bank Charges		13.78
Insurance		877.60
Lecturer/Tutor Fees	3944.42	
Parking		14.00
Photocopying		177.60
Postage	696.40	96.82
Printing & Stationery/ Photocopying	2924.90	49.00
Press Release		475.00
Travelling / Accomodation / Meals	8257.90	5968.88
	-----	-----
	24629.10	12819.22
	-----	-----
NET PROFIT /(LOSS)	<u>21389.77</u>	<u>(268.81)</u>

A.U.U.G. INCORPORATION

PROFIT & LOSS STATEMENT

FOR THE PERIOD ENDED 1ST JUNE, 1989 TO 31ST MAY 1990.

CONFERENCE A.U.U.G. 1990.

Income		NIL
<u>LESS EXPENSES</u>		
Advertising	737.92	
Photocopying & Printing	<u>626.80</u>	
		<u>1364.72</u>
NET LOSS		<u>\$1364.72</u> =====

A.U.U.G. INCORPORATED

PROFIT & LOSS STATEMENT

FOR THE PERIOD 1ST JUNE 1989 TO 31ST MAY 1990

<u>INCOME</u>	1990	1989
Membership	53190.60	41145.00
Nutshell Handbooks	19930.15	3907.22
Open Look Specification		409.35
<u>Usenix Proceedings</u>		
- San Francisco		1288.00
- San Diego	492.00	657.00
- Baltimore	330.00	
AUUGN / Back Issues	4147.11	
Subscriptions	1461.00	685.00
Mailing List	5959.50	2776.00
Interest Received	5431.13	4092.17
 <u>SUMMER 90</u>		
- Melbourne	3873.00	
- Sydney	3149.95	
- Canberra	250.00	
Security Video	360.00	
Security Pacific National Bank	509.24	
	-----	-----
	99083.68	54959.74
 <u>LESS EXPENSES</u>		
<u>BANK CHARGES</u>		
- Credit Card	440.76	291.53
- Government	127.26	58.72
- General	88.39	96.54
	-----	-----
	656.41	446.79
	-----	-----
 Donations		200.00
 <u>MANAGEMENT COMMITTEE / MEETING EXPENSES</u>		
- Airfares	4325.50	3331.00
- Accomodation / Meals	485.70	355.50
- Parking	28.00	63.55
- Taxis etc.	205.15	203.10
- Registration	44.00	

- Editors Float	200.00	
	-----	-----
	5288.35	3953.15
<u>MEMBERSHIP</u>		
Freight / Postage	856.96	
Photocopying	16.80	326.16
Printing	2396.93	104.22
Product Directory	4117.34	
	-----	-----
	7388.03	430.38
	=====	=====
<u>NUTSHELL</u>		
Freight / Postage	2043.68	1502.71
Purchase Handbooks	6903.18	5347.00
	-----	-----
	8946.86	6849.71
	=====	=====
<u>USENIX</u>		
Baltimore Conference	734.63	
	=====	=====
<u>USENIX PROCEEDINGS</u>		
<u>SAN FRANCISCO</u>		
- Postage		277.05
- Purchase Proceedings		956.32
	-----	-----
	-	1233.37
	=====	=====
<u>USENIX PROCEEDINGS</u>		
<u>SAN DIEGO</u>		
- Postage		192.49
- Purchase Proceedings		1735.00
	-----	-----
	-	1927.49
	=====	=====
<u>OPEN LOOK EXPENSES</u>		
Photocopying		383.60
Postage / Freight		41.15
	-----	-----
	-	424.75
	=====	=====
<u>A.U.U.G.N.</u>		
Postage / Freight	3444.88	338.85
Printing	33348.88	29921.99
	-----	-----
	36793.76	30260.84
	=====	=====

<u>SUMMER 90</u>		
Administration	2898.70	
Melbourne	3576.62	
Sydney	3337.20	
Tasmania	358.00	
	-----	-----
	10170.52	
	=====	=====

<u>MAILING LIST</u>		
Photocopying / Printing	667.20	127.00
Postage / Freight		1382.57
	-----	-----
	667.20	1509.57
	=====	=====

<u>A.U.U.G. 89</u>		
Freight		73.21
Postage of CFP		577.94
	-----	-----
	-	651.15
	=====	=====

<u>OFFICE</u>		
Accounting	1850.00	
Freight / postage	1284.51	870.75
Petty Cash	101.20	
Printing / Stationery	2379.92	405.55
Purchases - Security Video	468.00	1644.46
Trademark Registration	25.26	450.00
	-----	-----
TOTAL OPERATING COSTS	76754.65	51257.96
	-----	-----
GENERAL A/C NET PROFIT	22329.03	3701.78
A.U.U.G. 89 NET PROFIT / (LOSS)	21389.77	(268.81)
A.U.U.G. 90 (LOSS)	(1364.72)	
	-----	-----
NET PROFIT	42354.08	3432.97
	=====	=====

AUUG INCORPORATED

BALANCE SHEET

AS AT 31ST MAY, 1990.

	<u>NOTE</u>	<u>1990</u>	<u>1989</u>
<u>CURRENT ASSETS</u>			
Cash		25151.48	27000.40
Receivables	(3)	6168.50	1922.55
Investments	(2)(4)	<u>58552.93</u>	<u>28832.57</u>
TOTAL CURRENT ASSETS		89872.91	57755.52
		=====	=====
<u>NON - CURRENT ASSETS</u>			
Intangibles		<u>988.10</u>	<u>988.10</u>
TOTAL NON - CURRENT ASSETS		<u>988.10</u>	<u>988.10</u>
TOTAL ASSETS		90861.01	58743.62
		=====	=====
<u>CURRENT LIABILITIES</u>			
Creditors and borrowings	(5)	<u>0.00</u>	<u>10236.69</u>
TOTAL CURRENT LIABILITIES		<u>0.00</u>	<u>10236.69</u>
<u>ASSOCIATION FUNDS</u>			
Accumulated profits		<u>90861.01</u>	<u>48506.93</u>
TOTAL LIABILITIES & CAPITAL		90861.01	58743.62
		=====	=====

A.U.U.G. INCORPORATED

NOTES TO AND FORMING PART OF THE ACCOUNTS

FOR THE YEAR ENDED 31ST MAY, 1990.

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 cost. Capital Gains Tax is not taken into account in determining the investments unless a definite 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 a receivable basis.

3. CURRENT RECEIVABLES

	1990	1989
<u>TRADE DEBTORS</u>	\$	\$
Membership	1328.00	1080.00
Mailing List	1133.50	709.00
Video	160.00	-
Nut Shell Books	3547.00	-
San Diego Proceedings	-	90.00
Openlook	-	43.55
	<u>6168.50</u>	<u>1922.55</u>
	=====	=====

4. CURRENT INVESTMENTS

	<u>1990</u>	<u>1989</u>
<u>QUOTED INVESTMENTS</u>		
- at cost		
Chase AMP	31000.00	12000.00
C.B.A.	<u>27552.93</u>	<u>16832.57</u>
	<u>58552.93</u>	<u>28832.57</u>

5. CURRENT CREDITORS & BORROWINGS

UNSECURED:

Bank Overdraft	164.09	164.09
Sundry Creditors	-	<u>10072.60</u>
	<u>164.09</u>	<u>10236.69</u>

6. MEMBERSHIP FEES

- Individual	27072.00	-
- Institution	<u>25619.60</u>	<u>-</u>
	<u>52691.60</u>	<u>41145.00</u>

No breakdown of figures were supplied for the year ended 31st May, 1989.

AUDITOR'S REPORT

MEMBERS OF A.U.U.G. INCOPORATED

We have audited the financial accounts, namely the Profit & Loss Statement, Balance Sheet and accompanying notes in accordance with Australian auditing standards.

In our opinion :

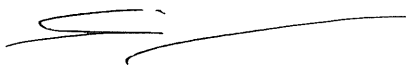
(i) the financial statements present fairly the state of the associations affairs at 31st May 1990 and the result of its operation for the year then ended: and

(ii) the financial statements have been drawn up in accordance with Australian accounting standards.

Date : 26th September 1990

Firm : Nicol & Nicol
Certified Practising
Accountants

Address : 230 York Street, South Melbourne.


.....
STUART C. NICOL
PARTNER

AUUG Election Report 1990

The annual election of office bearers was conducted in June. The Returning Officer was instructed by the Secretary, Mr. Tim Roper, that in the course of conducting the annual election several other ballots were to be conducted. These ballots concerned changes to the constitution rules and a nomination for Life Membership. There were 473 ballot papers mailed out including a report on suggested changes to the constitution.

Apart from a couple of spelling errors which slipped through, the election was carried out with little difficulty. The Assistant Returning Officer, Mr. David Purdue, came to Sydney the weekend following the closing of ballots and we counted all the formal votes.

There was an average of 87 formal votes for each ballot. I wonder why people insist on posting informal votes. Several people failed to turn the paper over and complete the reverse side.

The results for officer bears were as follows:

President

Greg Rose

Secretary

Peter Barnes

Treasurer

Michael Tuke

General Committee

Frank Crawford

Pat Duffy

Andrew Gollan

Chris Maltby

Returning Officer

John O'Brien

Assistant Returning Officer

David Purdue

The management committee proposed a number of changes to the rules of incorporation. These changes were grouped according to their functionality. To assist with the description of the changes, a copy of the proposed rules were enclosed with the ballot paper. All the proposed changes were passed by more than the required majority. The changes included: alteration to the AIMS, setting of fees, calling of meetings, expansion of the Management Committee, the introduction of a new Office Bearer and changes to the voting procedure.

This was the first time a member has been nominated for the position of Honorary Life Member. Dr. John Lions, who has been an ordinary member for more than five years, was nominated for the position of Honorary Life Member following a petition to the management committee. Dr. Lions was elected as AUUG's first Honorary Life Member as a result of receiving the required majority of ballots cast.

John O'Brien
Returning Officer

Book Review

Life with UNIX – A Guide For Everyone

Don Libes & Sandy Ressler

(Prentice-Hall, 1989, 346 pages,
ISBN 0-13-536657-7)

Reviewed by Douglas A. Gwyn

Gwyn@BRL.Mil

This book should answer the vast majority of questions that a thoughtful user of the UNIX operating system is bound to have concerning aspects of UNIX that are not addressed by reference manuals and “how to” tutorials. This is important information, since UNIX is not only an operating system but also a philosophy of computing, a set of traditions, an active subculture, and a major market force. Most of the material in *Life with UNIX* is not readily available from other sources, making this an essential reference volume for a well-rounded UNIX library.

Written in an informal style, *Life with UNIX* tells you everything you ever wanted to know about UNIX and even things you didn't know you should ask about, among them: UNIX evolution and politics of its development; versions of UNIX and portability issues; UNIX licensing and the UNIX-based systems market; standards, changing technologies, and the future of UNIX; sources of printed information; tools and the use of the shell environment; C, system programming, and programming support tools; system administration and system (in)security; Usenet, public-domain software, and games; benchmarking, consulting, mailing lists, validation, and typesetting services; UNIX applications; and databases, emulators, internationalization, networks, parallel processing, real-time processing, and workstations. Also useful are the listings of conferences, workshops, courses, and user groups.

This guide includes a “Who's Who” listing of significant names in the UNIX community, brief reviews of UNIX-related books and periodicals, addresses for numerous organizations, and an excellent comprehensive index that helps locate answers to such vexing questions as “What is the NUXI problem, anyway?” There are also many items that readers may find entertaining. These include quotations, anecdotes, and descriptions of some of the ways the UNIX community has fun, such as the P1003 WeirdNIX competition and the annual International Obfuscated C Code Contest.

Life with UNIX is quite an accomplishment. I noticed only two nontrivial flaws: It is riddled with minor inaccuracies, roughly one per page. While these do not detract from its use as a significant source of information about the UNIX phenomenon, they do render it unsuitable as a primary reference source and for settling “bar bets.” I also feel that its attempts to foretell the future, especially its pronouncements about “the way things should be,” do not measure up to the quality of the rest of the book. For instance, the authors speak approvingly of interfaces like the Macintosh Finder replacing the traditional UNIX shell, presumably as one step toward turning computers into appliances. Improved user interfaces are undoubtedly possible, but the Finder is not sufficiently “programmer friendly.” What made UNIX great was that it was designed by skilled programmers for use by them, e.g., enabling programs to support further programs, thereby obtaining tremendous leverage.

This book should be a prerequisite for posting questions to the Usenet newsgroup *comp.unix.questions*, because it answers nearly all the obvious questions about UNIX.

;login: 15:3

Report on ISO/IEEE JTC1/SC22/WG15 Rapporteur Group on Internationalization Meeting

March 5-7, 1990, Copenhagen, Denmark

Dominic Dunlop, The Standard Answer Ltd., domo@tsa.co.uk

Denmark. A small country which has tax rates so high that its five million inhabitants complain that, when they buy themselves a car, they have to buy one and a half cars for the government. Some part of that tax goes to fund Dansk Standardiseringraad (DS), the national standards body, which works hard to ensure that the needs of Danes are not overlooked when larger nations get together to write standards. DS has got its teeth into international standards for computers, and with good reason: we've been doing things wrong all along. We'll have to mend our ways if we are to produce standards which really fill international needs, even if we don't go as far as building in a framework which can easily accommodate Danish taxation.

Metropolitan Chicago today has a population larger than that of Denmark. Imagine that you've just rebuilt the downtown area after the fire of 1871, only to have Alexander Graham Bell come along with the telephone, Edison deciding to generate electricity, and railroad companies starting to promote inter-urban lines. All these innovations need new infrastructure - cables and conduits and tunnels which you just hadn't known you'd need when you laid the roads, put up the buildings, and connected them to gas, water and drainage. As a result, competing telephone and electric companies string a tangle of wires from poles with little regard to safety and no regard for aesthetics or standardization, while elevated railways appear above existing roads, cutting off light at street level and filling upper floor rooms with smoke.¹ Only after many

1. In 1887, the West Chicago Protective League complained "... the proposed elevated road would materially and irreparably depreciate the value of real estate upon said streets... and render the dwellinghouses thereon unfit for private residences..."^[1] but amid the kind of political maneuverings for which the city is justly famous, the "El" got built anyway.

years of disruption, digging up streets and making holes in the walls of existing buildings would telephones, electricity and public transportation be safely hidden beneath the ground,² unseen, but playing an essential part in supporting the life of the city.

A descendant of Alexander Graham Bell's telephone company now supports the UNIX operating system out of Chicago. UNIX is a lot like the Chicago of the last century. We're at the stage of unifying the major variants in the POSIX standards and the commercial System V, release 4, only to find that there is an increasing clamor for whole new infrastructures to support international needs, to improve security, and to show that the system is performing as billed. Suddenly, we've got to add features to handle these requirements, and we've got to try to do it while observing the three conflicting maxims of standardization: do it once, do it right, and do it now. What's more, we have to try to do it in a way which remains hidden: existing programs should not break, nor should they get noticeably bigger or slower.

POSIX is not alone: those responsible for computer language standards face the same problems, and have also been the subject of constructive Danish criticism.^[2,3] The Danes' long-standing interest makes it particularly appropriate that the first meeting of the ISO POSIX working group's special interest group on internationalization should be hosted by DS in Copenhagen. Internationalization is the process of removing cultural bias from a system, and then providing tools to allow system administrators to localize the system by adding a cultural bias of their own choosing. No wonder Dansk Standardiseringraad is interested in this technology: its employees

2. Well, in the case of Chicago, some of the public transportation. You can still ride the El.

court a syntax error every time they type its name at the UNIX shell.³ Internationalization will allow Danes to mold systems to their requirements, rather than having to rub along with implementation assumptions based on American practice.

The Japanese are interested too: their cultural differences make Denmark look close enough to the U.S.A. to be a fifty-first state! And the U.S.A. is interested because it has been charged by ISO with the production of ANSI standards base documents for the international POSIX standards, and wants them to reflect international needs. Denmark, Japan, and the U.S.A. sent representatives to the internationalization meeting. There were also observers from EUUG/USENIX (myself), the IEEE's 1003.0 working group, and from an ISO study group which is grappling with the issues of character set use in computer languages.

The official title of the POSIX internationalization group is the ISO/IEEE JTC1/SC22/WG15 Rapporteur Group on Internationalization. Just to explore some of the jargon, a rapporteur is a technical expert nominated by a member body – a national standards organization such as ANSI or DS – to take an interest in a specialized aspect of a particular standards effort. WG15, the ISO POSIX working group, has rapporteur groups on security, conformance testing, and internationalization. The security group met in January, in conjunction with the New Orleans meeting of the IEEE 1003.4 working group; the conformance test group, which corresponds to the IEEE 1003.3 effort, met in Copenhagen along with the internationalization group (although this report does not cover its meeting).

Internationalization is peculiar in that, although the IEEE's POSIX standards are drafted with international needs in mind, there is no internationalization working group within the POSIX project. There is a study

group which, as part of the 1003.0 "POSIX Guide" work, is trying to decide how to bring internationalization into the official structure, so that it can be given officers, schedules, terms of reference, and all those other good things which make us standards people feel safer. It's a big problem, because the issue really affects every aspect of POSIX – it just took a while to realize that it was an issue at all. Unlike realtime extensions, security extensions, or transparent remote file access for POSIX, internationalization doesn't really make sense as an add-on to a basic operating system interface standard. Rather, the operating system and all its extensions need to be internationalized as a matter of course. Every other working group in the IEEE POSIX is charged with producing a distinct standard, but it is difficult to see how a new group dealing with internationalization could be given such a goal.

ISO has a similar problem, but it's worse because the organization has so many balls to keep in the air. If it is to apply the "do it once" and "do it right" maxims to internationalization, it seems clear that the issue must be handled near the top of Joint Technical Committee 1, the information technology standards group. After all, as well as computer languages and operating systems, internationalization affects communications, document standards, database, and much more. ISO recently bit a similar bullet, establishing a new subcommittee (SC27) immediately below JTC1 to handle the security issues which are beginning to affect so much of its work. It may yet do the same with internationalization.

The "do it now" criterion, on the other hand, argues in favor of addressing internationalization at a lower level – doing the work in a new department, rather than going to the trouble of establishing a whole new division. SC22, which is responsible for language and operating system standards, is currently considering the formation of a new working group at the same level as WG15 (C language), WG15 (POSIX), and the rest. This proposal has run into opposition, both from those who say that the issue should be handled at a higher level, and from those who feel that there isn't an

3. ISO 646,^[4] the earliest ISO standard for information technology, is the international derivative of ASCII. Its Danish variant replaces ASCII's } with aa. Around the world, # \$ @ [] ^ _ { | } ~ , all of which have a special meaning to the shell, are replaced by other characters in standards derived from ISO 646. See ^[5] for much more information.

;login: 15:3

issue: after all, aren't ISO's standards supposed to be international anyway?

Meanwhile, WG15 has established a subordinate group to handle internationalization at the lowest level possible. As somebody said at the meeting, "You can't get much lower than us." We spent our time discussing what we were supposed to be doing – and, equally important, what we could leave to others. In the end we came up with a little list:

Terms of Reference

The rapporteur group on internationalization (RIN) will study the aspects of internationalization related to POSIX and report its findings to SC22/WG15.

(Bland, imposing no needless restrictions on what we can do.)

Program of Work

1. Carry out survey to capture most of the requirements relevant to internationalization.

(A job and a half. We have to search out users around the world, and persuade them to tell us what features they really want, rather than what they can put up with, or program their way around.⁴)

2. Identify and forward requirements with recommendations to WG15.

(So WG15 gets to carry the can for us...)

3. Capture and collect national body profiles for reference.

(Denmark and Japan have already done some work on "profiles" that customize POSIX to suit local needs. Their work suggests that current internationalization features are inadequate.)

4. Perform investigations as needed to advance the internationalization work of WG15.

(We can poke our noses into anything that takes our fancy...)

5. Review, from an internationalization perspective, documents submitted to WG15 for review and comment from an internationalization perspective.

(We definitely get to poke our noses into anything that comes past WG15...)

4. But we need to be a lot more diplomatic than asking "What ticks you off most about these dumb American machines?" – although appeals to chauvinism have been known to achieve results...

6. Review, and evaluate impact on work of WG15 of, other documents relevant to internationalization circulated in JTC1 or its subcommittees.

(And we'll try to get our hands on information from further afield.)

That's a lot of work. It defines the function of our particular mill, but that mill still needs grist. That feedstock has to come from outside our group, and, because of our lowly position, we have to ask WG15 to ask others to supply it. WG15, in turn, may have to refer some requests to higher authority: we want to be aware of anything which happens in SC22 which is relevant to POSIX internationalization – for example, what the C language people in WG14 are up to. That involves going up another level in JTC1's hierarchy. Getting in touch with other subcommittees, such as SC2, which looks after character sets, potentially involves going right to the top of the bureaucracy. (Luckily, in this particular case, SC22's study group on character sets can stand in for SC2.⁵) Consequently, when WG15 next meets in Paris in June, it will have to deal with several resolutions concerned with turning on the taps and starting the information flow to the rapporteur group.

One of these taps is a little sticky: WG15 doesn't officially have a relationship with the IEEE's 1003.0 group, although it can, via ANSI, talk to 1003.1, 1003.2, and 1003.4 through 1003.9. The problem is that 1003.0 deals with profiles, baskets of standards which, when brought together, solve particular classes of problems – for example, those of transaction processing, realtime, or batch-oriented systems. Profiles are outside the scope of the ISO POSIX effort, so we can't officially talk to 1003.0, even though its study group is currently holding the baton on internationalization. Never mind. We'll do things unofficially until some official pathway is sorted out.

5. SC2's answer to life, the universe and everything is DP (draft proposal) 10646, which defines a 32-bit wide character set with 8- and 16-bit wide canonical versions for storage and transmission, and a 24-bit wide processing version for those who can get by with only eight million characters or so. As it's still at the DP level, it'll be a long time before it hits the streets, and, even when it does, there's the little matter of getting people to use it...

Apart from all this organizational stuff, we did review some existing documents. For example, DTR (draft technical report) 10176, a product of SC14, discusses the treatment of characters appearing in language constructs, variable names, literals, and comments, and turns out to have implications for *sh*, *awk*, *yacc*, and the other “little languages” defined in DP 9945-2, the forthcoming international standard for the shell and tools. And a document from SC22’s study group on character sets suggests that source files should have some means of announcing the character set that they’re using. Could this mean typed files or resource forks for POSIX?⁶ How would we hide that?

The group next meets in Paris in June, just before the WG15 meeting. If you want to come along, you have to persuade your national standards body firstly that you’re a technical expert on POSIX, and then that they should appoint you as internationalization rapporteur. This may be surprisingly easy – considerably simpler, for example, than getting somebody to fund your trip. To quote from ^[8], “...standards committees would be hard-pressed to find people who participate on their voluntary committees with purely rational-economic expectations. Standards committees seem bent on justifying their existences by using hard data to prove that standards are good, yet they persist in using altruistic appeals to attract committee members.” If you feel like responding to the altruistic appeal of this article, contact me by electronic mail.

Alternatively, if you’re a European, you can remain seated in front of your terminal and participate in a news forum on ISO 646 and all that: Keld Simonsen of the Danish UNIX Users’ Group has volunteered to initiate a discussion of the European perspective on character sets for POSIX. Denmark may be small, but it’s certainly making its voice heard on this issue!

References

1. Brian J. Cudhay, *Destination Loop*, Stephen Green Press/Viking Penguin (1982).
2. P. J. Plauger, *Quiet Changes, Part I*, *The C Users Journal*, vol. 8, no. 2 (February, 1990), pp 9-16.
3. Keld Simonsen, *A European Representation for ISO C*, *European UNIX systems User Group Newsletter*, vol. 9, no. 2 (Summer 1989), pp 15-18.
4. ISO 646:1983, *Information processing – ISO 7-bit code character set for information interchange*.
5. Keld Simonsen, *An extension to the troff character set for Europe*, *European UNIX systems User Group Newsletter*, vol. 9, no. 2 (Summer 1989), pp 2-14.
6. ANSI X3.159, 1989, *Programming Language C*.
7. P. J. Plauger, *Evolution of the C I/O Model*, *The C Users Journal*, vol. 7, no. 6 (August, 1989), pp 17-25.
8. Carl F. Cargill, *Information Technology Standardization: Theory, Process and Organizations*, Digital Press (1989).

6. UNIX’s elegant and flavorless files have already taken a beating from X3.159, the ANSI C standard^[6], since other operating systems tend to support filing schemes which are merely tasteless.^[7]

International Standardization

An Informal View of the Formal Structures as they Apply to POSIX Internationalization

Dominic Dunlop, domo@tsa.co.uk

January, 1990

This article provides an overview of the way in which the international standards community works, insofar as it affects POSIX and the incorporation into POSIX of internationalization features. I'm not going to describe the technology underlying internationalization other than to say that its aim is to make the operating system and applications software independent of the user's spoken language and its representation (character sets, collation, text direction, and so on). This done, localizations specific to each group of users can tailor programs to their requirements without the need for expensive and legally-problematic hacking of source code. (If you want to know more, let me know, and I'll either expand on the topic, or give a few pointers.)

Figure 1 shows the relationship of standards bodies as far as POSIX is concerned. (The picture may look very different for other standards efforts, such as Open Systems Interconnection, but that need not concern us here.)

All standards must originate somewhere, whether in industry, in a professional association, in a national standards body, or in an international standards body. In the case of the POSIX family of standards, the Institute (IEEE) has assumed responsibility for the initial production of the documents. The IEEE is a professional association which is open to qualified engineers, no matter what their nationality. It has been involved for many years in the production of consensus standards - that is, standards arrived at through a formal process which gives ample opportunity for any interested party to comment and vote on proposals.

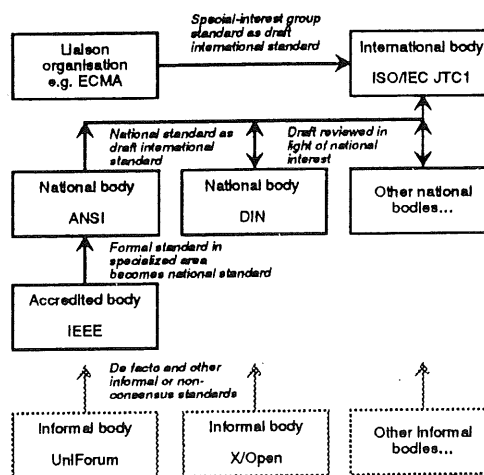


Figure 1

According to the standards procedures of the IEEE, the main group of interested parties is its membership, although non-members are also allowed to participate. Unusually among standards bodies, voting on IEEE standards is nominally "one member, one vote." (More typical standards bodies vote by corporation or by country.) The exception to the IEEE's individual voting scheme is that institutions can also participate, provided that they represent a broad constituency, rather than a single narrow commercial interest. Currently represented on the POSIX effort are the Open Software Foundation, UniForum, UNIX International, USENIX, and X/Open. None of these is an official standards body, although all are involved in the production of materials on which future standards may be based. In some cases, the organizations produce documents which look and smell like standards but which, because they are not produced by an open (and slow, and legalistic) consensus process,

may well show some bias towards the interests of the originating organization. Known broadly as industry standards, these documents appear before consensus standards, and must subsequently be brought into line if a consensus standard is to succeed.

As Figure 1 shows, in the hierarchy of standards organizations, the IEEE is near the bottom. Above it is firstly the national level, then the international. As the IEEE is based in the U.S.A., it has gained accreditation from the U.S. national standards body, ANSI (the American National Standards Institute). This means that ANSI considers the IEEE competent to produce national standards on behalf of ANSI. Of course, accreditation by ANSI gives rise to an anomaly: the IEEE, through a democratic process potentially involving an international membership, is creating national standards for the U.S.A. I shall return to this issue later.

ANSI, in turn, is a "member body" of Joint Technical Committee 1 (JTC1), an international standards body formed jointly by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) to handle the standardization of information technology. ANSI's role in JTC1 is nominally to represent U.S. interests in the "one nation, one vote" process by which international standards are ratified. Other member bodies such as DIN (West Germany), JISC (Japan), and IRISI (Iran), play a similar part, making sure that no standard conflicts with their own national interests.

Member bodies may sponsor draft standards at the JTC1 level. In the case of POSIX, ANSI is the sponsor. It is the expectation of the international standards community that a draft standard sponsored by a national member body in this way is likely to show a bias towards the needs and culture of that member body, and so may require amendment and perhaps extension before it is suitable for adoption as an international standard. Certainly, both POSIX and the C language have come in for criticism at the international level for their lack of support for non-Roman alphabets.

In order to root out and correct any bias or omission in a draft standard sponsored by a particular member body, other member bodies are expected to pore over the proposal, and feed in changes which reflect their national needs. Obviously, this could take forever: approaching a hundred countries are represented on JTC1. Typically, the number of member bodies participating in a particular standards effort is limited, and of these few play a very active role. In the case of the POSIX effort, around a dozen member bodies are circulated with the working group's paperwork, and of these, perhaps half are regularly represented at its meetings. Even so, by the time a national standard has progressed to the level of becoming a JTC1 draft, it is rather late to begin making changes – particularly if, as is the case for POSIX and C, there is a pressing need for an international standard.

As presented so far, the standards world is strictly hierarchical: a standard such as POSIX progresses from an accredited special interest group within a country, firstly to national level, and finally to international status. Officially, it is not until the final stage that interests outside the originating country get to comment on it. The process could be made more efficient if interest groups outside the originating country had a means of commenting at an earlier stage, but the hierarchy seems to preclude such comment.

Interestingly, there is a "side door" at the international level which can be used to short-circuit the normal time-consuming process. The top level of Figure 1 shows an organization in liaison with JTC1, the European Computer Manufacturers' Association (ECMA), which has gained the privilege of being allowed to propose and comment on standards at the international level. The process of obtaining liaison status is both difficult and lengthy, and is open only to international organizations with a valid claim to representing a specific broad area of interest. (Besides ECMA, the World Health Organization and Mastercard International are among the sixty or so bodies in liaison with JTC1.) If the members of a liaison body can formulate a standard which is useful to them, liaison status allows that standard to be proposed for

adoption as a formal international standard. Since all bodies with such status are themselves international (or at least regional), such proposals are likely to satisfy international needs without much need for amendment. (ECMA has sponsored several standards for magnetic media in JTC1; banking interests have been active in the standardization of credit cards.) Indeed, JTC1 has developed a "fast track" approvals mechanism for use when member bodies agree that little review is necessary – although it has to be said that not every use of the fast track has resulted in a standard being approved.

The strict hierarchy imposed by ISO makes for easy and obvious management control, but is under some strain. Firstly, where emerging standards seek to accommodate international needs from their first drafting, the late review by national member bodies provided by ISO makes for unnecessary delay – delay which could be avoided if national bodies had an official means of providing input at an earlier stage. Secondly, regional standards organizations – most notably CEN, the European Standards Centre – are growing in importance, and do not fit well into a scheme which is set up according to strictly national guidelines.

These two problems combine to foster provincial attitudes on the part of standards makers – and politicians – involved with POSIX both inside and outside the U.S.A. Those inside reason that, since they are creating a U.S. national standard, international considerations are relatively unimportant, and can be left for later. Outside the U.S., standardizers reckon that it will be so long before they can mold a U.S.-produced standard to their own requirements that they might as well develop their own, probably incompatible, standards to fill their immediate needs. In Europe, a proposal to adopt issue 3 of X/Open's Portability Guide (XPG3) as a standard was strongly backed for a while, even though XPG3 is not wholly aligned with POSIX. (On the reasonable grounds that the 1003.1 standard had not been approved at the time of publication. XPG4 will be aligned with POSIX.) Interestingly, just as the IEEE is seen in Europe as representing U.S. interests,

X/Open is seen by many U.S.-based observers as a European outfit, despite its many U.S. members.

Provincial attitudes among technical people and their managers outside the U.S.A. exacerbate the problems. Although the IEEE makes some effort to reach this constituency by holding one of the quarterly working group meetings outside U.S. every couple of years, the majority of attendees are always Americans. Europeans in particular seem, even if they have the inclination to attend, to find it difficult to justify the expense to their management. The interests of Arab countries and the Indian subcontinent are seldom represented at all. In contrast, delegates from Japan and other Pacific rim countries have been attending meetings in increasing numbers, even when lengthy and costly travel is involved.

Given the current structure of the international standardization community, is it possible to work within it and yet overcome the two problems which face the POSIX effort: that of obtaining useful international input at an early stage; and the parallel problem of preventing divergence between POSIX and emerging industry, national, and regional standards? Can the current structure accommodate formal mechanisms which provide for solutions, or will the problems remain unless the structure itself is changed?

Until now, practical international input to POSIX has come from two sources which are not a part of the formal hierarchy of international standardization: UniForum and X/Open. As I have already mentioned, X/Open is an international grouping seen by some as primarily European; its active membership has to date consisted of computer suppliers. UniForum, which was known as /usr/group until 1989, is a grouping of hardware suppliers, software authors, value-added resellers, and users. As with X/Open and other groupings, it is the suppliers which have played the largest part in the organization – users have seldom made their voice heard. UniForum is U.S.-based, but has affiliates around the world. These affiliates are largely autonomous, and, despite efforts to involve them, have played almost no part in UniForum's standards activities – even when

these are involved with internationalization. (While UniForum's Technical Subcommittee on Internationalization has active participation from outside the U.S.A., the people concerned became involved directly, rather than through their local UniForum affiliates.) USENIX, the other user grouping with institutional representation to the IEEE POSIX project, has a better claim to providing a forum for users, but is almost exclusively North American, and, unlike UniForum, has no internal structures concerned with standardization. The European UNIX systems User Group (EUUG) has a truly pan-European membership made up, like that of USENIX, primarily of computer programmers and technical users, but has not participated officially in any standards effort. Its involvement to date has been confined to the co-sponsorship with USENIX of a standards monitor service, which provides members with information about progress on POSIX and in related areas.

It is my view that, if international interests are to play a greater part in the drafting of POSIX standards, they must be represented formally within the IEEE. This is not to minimize the importance of the work done by UniForum, but rather to say that an official stamp of some sort is necessary in order that its importance receives a wider recognition both inside and outside the IEEE. Unlike other topics handled in the past by UniForum, real-time and transaction processing among them, internationalization has never officially been incorporated into the POSIX effort because it cannot stand alone. There cannot usefully be such a thing as a standard for internationalization; rather, internationalization should be a consideration in the drafting of any standard for computer software.

The 1003.0 (POSIX Guide) working group is currently wrestling with the problem of handling internationalization issues within POSIX. It may be possible to borrow a useful concept from ISO: that of the rapporteur group. Rapporteur groups cut across normal boundaries, bringing together those who are interested in some problem or activity which is common to a number of standards projects.

It is over-optimistic to hope that bringing internationalization officially into the POSIX fold will result in immediate participation by those who currently wait until documents reach the ISO level before commenting through their national member bodies. One way to reach this audience might be to convince it that the IEEE is indeed an international, rather than strictly North American, grouping. A radical way of achieving this would be for the IEEE to seek liaison status with JTC1, so obtaining a means of submitting base documents directly, instead of through ANSI. To do this would involve the IEEE in the considerable expense and logistic complexity of sponsoring standards – a task for which resources are not currently in place in an organization which seldom gives the appearance of being over-endowed with resources.

In any event, even if the IEEE were to apply for liaison status tomorrow, it would be a long time before it was granted. Unless or until this happens, it seems to me that it is the duty of user groups around the world to encourage their members to play a part in the process through the IEEE. So that's what I've been doing in this article!

An Update on UNIX and C Standards Activity

Jeffrey S. Haemer

Report Editor, USENIX Standards Watchdog Committee

What the reports are about

Reports are done quarterly, for the USENIX association, by volunteers from the individual standards committees. The volunteers are familiarly known as "snitches" and the reports as "snitch reports." The band of snitches and I make up the working committee of the USENIX Standards Watchdog Committee. The group also has both a financial committee: Alan G. Nemeth, Ellie Young, and Kirk McKusick (chair); and a policy committee: the financial committee plus John S. Quarterman (chair). Our job is to let you know about things going on in the standards arena that might affect your professional life — either now or down the road a ways.

An official statement from John:

The basic USENIX policy regarding standards is:

to attempt to prevent standards
from prohibiting innovation.

To do that, we

- Collect and publish contextual and technical information such as the snitch reports that otherwise would be lost in committee minutes or rationale appendices or would not be written down at all.
- Encourage appropriate people to get involved in the standards process.
- Hold forums such as Birds of a Feather (BOF) sessions at conferences. We sponsored one workshop on standards.
- Write and present proposals to standards bodies in specific areas.
- Occasionally sponsor White Papers in particularly problematical areas, such as IEEE 1003.7 (in 1989).
- Very occasionally lobby organizations that oversee standards bodies regarding new committee, documents, or balloting procedures.

- Starting in mid-1989, USENIX and EUUG (the European UNIX systems Users Group) began sponsoring a joint representative to the ISO/IEC JTC1 SC22 WG15 (ISO POSIX) standards committee.

There are some things we do *not* do:

- Form standards committees. It's the USENIX Standards Watchdog Committee, not the POSIX Watchdog Committee, not part of POSIX, and not limited to POSIX.

- Promote standards.
- Endorse standards.

Occasionally we may ask snitches to present proposals or argue positions on behalf of USENIX. They are not required to do so and cannot do so unless asked by the USENIX Standards Watchdog Policy Committee.

Snitches mostly report. We also encourage them to recommend actions for USENIX to take.

We don't yet have active snitches for all the committees and sometimes have to beat the bushes for new snitches when old ones retire or can't make a meeting, but the number of groups with active snitches continues to grow (as, unfortunately, does the number of groups). This quarter, you've seen reports from .0, .1, .2, .3, .4, .7, .8, .11, and .12, as well as reports from 1201 and from X3J11 (not really a New Orleans report, but useful none the less).

If you have comments or suggestions, or are interested in snitching for any group, please contact me (jsh@usenix.org) or John Quarterman (jsq@usenix.org). If you want to make suggestions in person, both of us attend the POSIX meetings.

Reports on the October 1989 Meeting in Brussels

(continued from ;login: vol. 15, no. 2)

Report on IEEE 1003.2: Shell and tools

Randall Howard <rand@mks.com> reports on the October 16-20, 1989 meeting in Brussels, Belgium:

Background on POSIX.2

The POSIX.2 standard deals with the shell programming language and utilities. Currently, it is divided into two pieces:

- POSIX.2, the base standard, deals with the basic shell programming language and a set of utilities required for *application portability*. Application portability essentially means portability of shell scripts and thus excludes most features that might be considered interactive. In an analogy to the ANSI C standard, the POSIX.2 shell command language is the counterpart of the C programming language, while the utilities play, roughly, the role of the C library. POSIX.2 also standardizes command-line and function interfaces related to certain POSIX.2 utilities (e.g., popen, regular expressions, etc.). [Editor's note — This document is also known as "Dot 2 Classic."]

- POSIX.2a, the User Portability Extension or UPE, is a supplement to the base POSIX.2 standard; it will eventually be an optional chapter of a future draft of the base document. The UPE standardizes commands, such as screen editors, that might not appear in shell scripts but are important enough that users must learn them on any real system. It is essentially an interactive standard that attempts to reduce retraining costs incurred by system-to-system variation.

Some utilities have interactive as well as non-interactive features. In such cases, the UPE defines extensions from the base POSIX.2 command. An example is the shell, for which the UPE defines job control, history, and aliases.

Features used both interactively and in scripts tend to be defined in the base standard.

In my opinion, the biggest current problem with the UPE is that it lacks a coherent view: it's becoming a repository for features that didn't make it into the base standard. For example, *compress* is in the current UPE draft. It's hard to rationalize classifying file formats as an "interactive" or "user portability" issue, yet the one used by *compress* is specified in the UPE. It certainly doesn't fit in with a view of the UPE as a standard that merely adds utility syntax information (e.g., information that would allow users to type the same command line to compress a file on any system). This highlights the schizophrenic nature of the UPE: it addresses a range of different needs that, taken together, do not appear to define a whole. Dot 2 Classic, to my taste, appears to have far more unified scope and execution.

A second, related, problem with the UPE is that there appears to be less enthusiasm for it than for the base standard. A number of people, including me, understand the need for it, but it doesn't appear to have the strategic importance of the base. [Editor's note — The UPE is, frankly, controversial. Like 1201, the committee undertook the UPE out of a fear that if they didn't, NIST would do the job without them. Supporters note that although its utilities are probably not necessary for portability of most software, it would be unpleasant for programmers to do the porting work without them. Detractors counter that POSIX was never intended to cover software development and that the group is exceeding not only its charter, but that of the entire 1003 committee.]

Status of POSIX.2 Balloting

POSIX.2 is in its second round of balloting. The first ballot, on Draft 8, produced many objections that are only partially resolved by Draft 9. Although there were only fifty-four pages of unresolved objections remaining after Draft 9 was produced, the current balloting round is not restricted to existing objections, and there will almost certainly be many new

;login: 15:3

ones. Remaining objections range from the perennial war between David Korn and the UNIX Support Group over what features should be required in the POSIX shell, through the resolution of the incompatible versions (Berkeley and USG) of *echo*, to the treatment of octal and symbolic modes in *umask*.

A digression to illustrate the kind of issues being addressed:

In March of 1989, a study group from 1003.2 met at AT&T to resolve major objections to the shell specified in Draft 8 by the two warring parties. This was a good place to hold the meeting, since both parties are from AT&T: one led by David Korn of Bell Labs, the author of the popular Korn Shell (KSH) the other, a group led by Rob Pike of Bell Labs Research and the UNIX Support Organization, advocating more traditional shells, like the System V Bourne Shell and the Version 9 Research Shell. Korn's group contends that the shell should be augmented to make it possible to efficiently implement large scripts totally within the shell language. For example, while the more traditional camp views shell functions as little more than command-level macros and uses multiple scripts to modularize large shell applications, the Korn shell views functions as a tool for modularizing applications, and provides scoping rules to encourage this practice.

The two philosophies engender different opinions on issues such as the scoping of traps within functions and the use of local variables. Other contentious issues were the reservation of the brace ({}) characters as operators (rather than as the more tricky "reserved words"), the promotion of tilde expansion to a runtime expansion (like parameter expansion), and the issue of escape sequences within *echo*, *print*, and *printf*.

The meeting produced a false truce. I attended, and believe that both parties had different views of the agreement that came out of the meeting. As a result, Draft 9 produced balloting objections from both parties and the dispute continues unabated. Shades of POSIX.1 Tar Wars...

I suspect the next draft (Draft 10) will fail to achieve the consensus required for a full-use standard. This is a good thing. Useful features are still finding their way into the document. (Draft 9 introduces *hexdump*, *locale*, *localedef*, and more.) Also, the sheer size (almost 800 pages) of Draft 9 has prevented many balloters from thoroughly reviewing the entire document. Still, there is a stable core of utilities that is unlikely to change much more than editorially; I predict the standard will become final around Draft 12.

A mock ballot on Draft 4 of the UPE will probably start after the New Orleans meeting in January, and the resulting Draft 5 will probably go to a real ballot somewhere in summer to early fall of 1990. Although many sections remain unwritten or unreviewed, the UPE is a much smaller standard than POSIX.2 and should achieve consensus more quickly.

Status of the Brussels Meeting

The Brussels meeting focused on the UPE, with only a summary report on the status of balloting for the base standard. For most of the meeting, small groups reviewed and composed UPE utility descriptions. The changes generated at the meeting will appear in Draft 3.

The groups reviewed many utilities. The chapter on modifications to the shell language (for interactive features) is now filled in, and such utilities as *lint89* (the recently renamed version of *lint*), *more*, etc. are approaching completion. Still, much work remains.

[Editor's complaint — We think renaming common commands like *lint* ("lint89") and *cc* ("c89") is both cruel and unusual. We are not eager to re-write every makefile and shell script that refers to *cc* or *lint*, nor to retrain our fingers to find new keys each time the C compiler changes. The name seems to have been coined by either a hunt-and-peck typist, or someone who has longer and more accurate fingers than we do. (Was it, perhaps, the work of Stu Feldman, author of *f77*?) Moreover, replacing commands with newer versions is

commonplace and traditional in UNIX. Examples like *make*, *troff*, and *awk* spring to mind. If an older version is kept on for die-hards, it's renamed (e.g., *otroff*, *oawk*).

One Dot-Two member rebuffed our objections with the reply, "But, you see, this isn't UNIX: it's POSIX."

Because the meeting was in Europe, attendance at the working group meetings was lower than normal (20-25 rather than the normal 35-40 in POSIX.2). Nevertheless, the choice of location served a purpose. The meeting was held in Brussels to garner international support and participation, particularly from the European Economic Community. There were many EEC representatives at the background sessions on POSIX and two or three European working group members in the POSIX.2 meetings who wouldn't normally have attended. Though it remains to be seen what will come out of having met in Brussels, I am convinced that the extra effort will prove to have been justified.

Report on IEEE 1003.5: Ada-language Binding

Ted Baker <tbaker@ajpo.sei.cmu.edu> reports on the October 16-20, 1989 meeting in Brussels, Belgium:

The P1003.5 group is producing an Ada-language binding for 1003.1. The Brussels meeting had two objectives: to reach consensus on a draft document to be distributed for mock ballot, and to solicit input from the European community. We achieved the first but not the second; only one of the ten attendees was European (Olle Wikstrom, from Ericsson).

The technical editor (David Emery) and the chapter authors had worked very hard between meetings to produce version 3.2 of the document, and Dave brought copies to the meeting. The working group reviewed it to try to correct any serious errors or omissions before mock ballot.

There was a lengthy discussion about schedule and logistics for the mock ballot. The present plan is to send out copies of the next draft, in ISO format, to both the ISO and the entire 1003.5 mock ballot mailing list. [Editor's note: All committees are re-formatting their documents in ISO format to smooth the way for ISO acceptance (see Dominic Dunlop's report on WG15 for more details), and an IEEE copy editor appeared on the scene in Brussels to give P1003.5 guidance and help in this.] Since there is no way that enough input can be received before the next POSIX meeting, in January, the group has scheduled a special meeting for mock ballot resolution, between the January and April POSIX meetings, to be held in Tallahassee. The objective will be to produce a proposed standard to be reviewed at the April meeting.

Most technical issues discussed were minor, compared with previous meetings. The most significant, and complicated, was the treatment of system configuration limits. Here are three problem areas:

1. Tri-state configuration parameters (true, false, undefined) in the POSIX C binding need to be treated differently in the Ada binding, because Ada prohibits references to undefined symbols (i.e., Ada lacks an `#ifdef` facility).
2. For the same reason, it isn't clear how an Ada binding can accommodate future POSIX extensions. Suppose, for example, a future extension adds a new configuration constant. How does one write an Ada program that takes advantage of the new feature on implementations where it's available without preventing the same program from compiling on older implementations, where it's not?
3. Because Ada compilers can do optimizations, such as dead code elimination, based on static expressions (the nearest analog to some C preprocessor capabilities), it is important to provide compile-time constants, where safe. At the same time, to support "bubble pack" software that is usable on different system configurations, programs should also be able to defer binding such values until run time.

:login: 15:3

The group did achieve consensus on a treatment of configuration limits for the mock ballot. It includes a combination of functions, to allow software to defer resolution of system limits and characteristics until runtime, and implementation-defined constants and numeric ranges, to allow optimizers to take advantage of information available at compile time. This does not fully solve all the problems mentioned above. Perhaps the mock ballot process will turn up some suggestions for improvements.

The treatment of process arguments and environment variables, which must be provided as parameters when starting a new process or calling Exec produced another controversy.

Unlike C, Ada does not allow pointers to stack or statically allocated objects. An Ada POSIX interface implemented over a C-language binding must bridge this gap somehow. For example, an implementation might use a C-compatible data structure and hide the non-Ada details, or use an Ada data structure and translate between the two forms. Everyone agreed that the interface should avoid constraining the implementation, but the first interface solutions appeared to rule out desirable implementations. The present solution permits an application to ensure that if the Ada POSIX interface machinery allocates any "heap" storage this storage is be recovered, while allowing an implementation to impose restrictions that would permit stack allocation. A price paid for this compromise is that writing portable applications takes more care: an application that works OK with one implementation may lose storage or exceed size limits with another.

At the previous two meetings, we had substantial interaction both with other groups working on language-independence and with P1003.4 (real-time). There was much less this time, partly because the group was concentrating so hard on getting ready for mock ballot, partly because meetings were spread over several buildings, and partly because P1003.4 mostly skipped Brussels.

On the administrative side, Steve Deller was promoted from Vice Chairman to Chairman (in charge of external affairs and running meetings) and Jim Lonjers was chosen as Vice Chairman (in charge of administering ballot resolution). This change was required because the ex-Chairman (Maj. Terry Fong) has been unable to participate regularly in the working group recently, owing to conflicts with his professional duties.

Another issue that came up was whether working group members are at liberty to publish papers or present talks on the 1003.5 work. The answer is, "Yes." Until now, some members have been exercising self-censorship, based on an earlier agreement designed to discourage anyone (e.g., defense department personnel) from making commitments (e.g., requiring use of the POSIX Ada binding in contracts) based on erroneous (e.g., overly optimistic) progress reports. It did not take much discussion to agree that such censorship is now counterproductive, and may never have been wise. At this point, P1003.5 certainly wants public exposure of its draft document, and hopes that such exposure will generate more reviewers and active working group members.

Report on IEEE 1003.7: System Administration

Steven J. McDowall <sjm@mca.mn.org> reports on the October 16-20, 1989 meeting in Brussels, Belgium:

Background

Now, almost everyone agrees that 1003.7 should deal with networks, not just isolated systems. To wit, it would be nice if I could administer all the machines in a network from a single machine with simple commands. For example, to add a user to all machines in the domain *mn.org*, all I should need to do is issue a command like `adduser -d mn.org -options -parameters username`. The question is, without any *de facto* standard already in place to adopt, how can we achieve this?

The Approach

This is important, so pay attention. Because the major goal of 1003.7 is to create a standard way to manage a set of objects, the group has decided to take an object-oriented approach. Our idea is to begin by creating a list of objects to manage, then to follow that by defining the set of commands to manage each object. This approach is novel for both system administration and POSIX. It will probably require more work on the front end to define the objects, their attributes, and their relationships, than to define the actual command structure to support and manipulate them. Whether this approach will work remains to be seen.

The Meeting

The meeting was boring. To put it bluntly, the week was simply a work week. Objects (and sub-objects) were defined and discussed in detail, then put in the draft. Little got done on the first and last days, due to EEC formalities, which left us with three working days instead of the normal four and a half. Attendance was pretty dramatically reduced, too. About half the normal North Americans showed up, probably because of the location, and only one (yes one...) new European came even though we were meeting in Europe. Oh well, except for my having had my passport stolen, it was a good chance to see Belgium.

Concerns

1. The process is taking a long time to move ahead, both because of the difficulty involved and because we seem to attract less manpower than many other groups. Moreover, since we're taking a radical approach, it takes extra time to teach the ideas to anyone new that does come.

2. System administration doesn't have the glamour of some of the other areas being standardized. As the Rodney Dangerfield of POSIX, 1003.7 gets no respect.

3. The notation we're using to define our objects is ASN.1. "Why ASN.1?" you ask.

Simply because it's a standardized meta-language to describe abstract data types. The feeling was that this would help make the whole package more suitable for interoperability. I bring this up because there's some movement throughout 1003 to redo all data structures in a new meta-language created by some of the people working on language-independence. Not only would this require that we go back and redo our definitions, but I also think ISO will only allow the use of standardized data-languages in their standards. Does anyone out there know if there is such an ISO restriction? If so, it's important for 1003 as a whole, not just for dot seven.

4. Currently, almost all working-committee members are from vendors. IBM, DEC, HP, AT&T, and others are well-represented. A few interested parties, like OSF and /sys/admin are there as well, but as far as I can tell, there isn't one real user. By "real user" I mean someone who does nothing but administer a system. All of us are connected somehow with creating an administrable system or getting paid to do so. Of course, I should make clear that we all have to administer systems of our own, so we're not simply an ivory tower group with no real experience, but representation is still grossly unbalanced.

5. Finally, there's been a loss of focus on interoperability directly attributable to the loss of our X/Open representative, Jim Oldroyd. Jim was well respected and made many valuable contributions, but can no longer attend our meetings. As the X/Open representative, he was very concerned with multi-vendor environments, and was a major force in helping us focus on and ensure interoperability. I am not saying that no one else on the committee cares about the issue, but it does seem to be being pushed aside in a spirit of, "I think we shouldn't have any interoperability problems if we do this, so let's do it and worry about it later on." Jim had helped provide a more positive, direct approach of determining up front what would be needed for true interoperability. If X/Open is still interested in System Administration, and in making sure

;login: 15:3

the 1003.7 standard includes provisions for interoperability, we could still use their help.

Report on IEEE 1003.8/2: Networking (IPC)

Steve Head <smh@hpda.hp.com> reports on the October 16-20, 1989 meeting in Brussels, Belgium:

Overview

P1003.8 is the IEEE POSIX networking standards committee, working on network standard interface definitions for POSIX. The committee is currently divided into six subcommittees: transparent file access, network IPC, remote procedure call, OSI/MAP services, X.400 mail gateway, and directory services.

This report is a summary of the activity in the network IPC subcommittee, which is currently working on two potential interfaces, a "detailed" interface (DNI) and a "simple" interface (SNI). DNI is roughly (though not exclusively) at the transport level. SNI is intended to be somewhat simpler to use than DNI, but at roughly the same level.

At this meeting, presentations of DNI and SNI were made at the EEC (Common Market) headquarters in Brussels. Discussions on DNI (definitions) and SNI (routines) continued. The main topics of discussion were:

1. DNI, SNI presentation to EEC
2. DNI definitions
3. SNI routines
4. Schedule
5. Security
6. P1003.8/2 → full POSIX committee

Detail

1. DNI, SNI presentation to EEC

Keith Sklower and Steve Head gave presentations on DNI and SNI respectively to POSIX attendees at EEC (Common Market) headquarters. This meeting was scheduled in Brussels primarily to obtain European input. The

presentations went well, and attendees included X/Open and EEC representatives.

No significant differences of opinion or direction were noted between the committee and other attendees. This indicates some degree of success (?). (Other networking groups, such as directory services, were not so fortunate.)

This meeting "broke the ice" with international organizations in the area of networking, and we now expect increased interaction with those organizations.

2. DNI definitions

The committee discussed DNI definitions. Steve Head presented a paper on the subject. Suggestions made at the meeting will be incorporated into a future version of the paper, which will be circulated via electronic mail. If no further significant issues are raised, it will be incorporated into the next DNI draft.

3. SNI routines

The committee discussed SNI routines, based on a paper from Keith Sklower. No conclusions were reached, however, this particular discussion was very useful since it brought a number of goals and requirements for SNI into clear focus.

SNI is adopting some characteristics of ISODE (the ISO Development Environment). This is probably beneficial since it means that SNI will be partially based on a working implementation instead of being entirely new. As such, it may gain importance as a migration strategy for transferring applications from TCP/IP to ISO. (ISODE stands for the ISO Development Environment, a publicly available collection of networking software that runs over either TCP/IP or ISO transport and allows higher level applications to be oblivious to the type of transport a given system provides.)

4. Schedule

The working schedule has been delayed by the need to make presentations at Brussels, instead of doing "real work." Originally, we had scheduled the topics of connection setup,

connection tear-down, and name resolution for this meeting. These topics were not discussed, and our schedule has been shifted back a quarter to reflect this. These topics will be discussed at the next meeting.

5. Security

We held another joint meeting with the POSIX security group, P1003.6. An electronic mailing list was created for the topic of network security. For more info or to be put on the list, please contact Mike Ressler (bellcore!mpr or mpr@bellcore.com). A list of topics on networking security to begin discussions on was initiated.

6. P1003.8/2 → full POSIX committee

The decision to make P1003.8/2 a full POSIX committee was postponed by the POSIX executive committee (SEC). This subject will be re-addressed at the next POSIX meeting in January.

Reports on the January 1990 Meeting in New Orleans

Report on IEEE 1003.0: POSIX Guide

Charles Severance <crs@convex.cl.msu.edu> reports on the January 8-12, 1990 meeting in New Orleans, LA:

Dot zero is producing a guide to the POSIX Open System Environment (OSE). The guide will bring existing and evolving standards together to provide specifications for all aspects of an OSE – everything from application programming interfaces to user interfaces and system management. It will give users an overview of the 1003, and other related standards, describe their interrelationships, and help them select the subset of available standards necessary for any particular application.

Draft Six Review

The group reviewed draft six, and points of special interest were:

- the formal definition of “open system”
- internationalization
- an editorial review of the entire document to ensure a consistent style
- a review of some high-level architecture diagrams, proposed to make Chapter 3 (“Overall Architecture”) easier to understand

The only one of these discussed by the entire group was the definition of “Open System.” To simplify the definition we created a definition for “Open Standard” which was used in the Open System definition. Here is the definition we finally agreed on:

Open System: A system that implements sufficient Open Specifications for interfaces, services, and supporting formats which enable properly engineered applications software: a) to be ported across a wide range of systems with minimal changes, b) to interoperate with other applications on local and remote systems, and c) to interact with users in a style which facilitates user portability.

Open Specification: A public specification which is maintained by an open, public, consensus process to accommodate new technologies over time and consistent with international standards.

The group won’t define “user portability” until next meeting, but the idea is that users should see a consistent interface from application to application, both within and across systems. Public user-interface standards should simplify both user training and vendor documentation.

The other issues were handled in small working groups.

1. Internationalization. This group identified parts of the document affected by internationalization and other “cross-component” issues, such as system management and security. They promise to present new draft text for the

;login: 15:3

internationalization sections by the next meeting.

2. Editorial review. This group tackled the no-fun jobs of reviewing the entire draft for style and identifying areas that had too much, or too little, detail. Along the way, they proposed a style guide and template for sections of Chapter 4.

3. Architectural overview. This group continued work on Chapter 3 to complete the text of the chapter, and worked to simplify it, and make it easier to understand. The CCTA (UK) presented a high-level classification scheme called "MUSIC" (Management, User Interface, System Interface, Information Interchange, and Communication) as a potential contribution to chapter 3. The chapter will have extensive modifications and additions for the next meeting.

Application profiles

Next meeting we'll discuss exactly what must be in a POSIX Application Environment Profile (AEP). Profiles will affect and generate procurement issues, so this will be a key discussion.

Profiles specify a set of standards for specific computing areas, such as supercomputing. Not all standards will be required for all areas; a profile lists the subset of the standards necessary for a particular area.

The biggest point of contention in this discussion will probably be whether 1003.1 [Editor: the system interfaces set out in the Ugly Green Book] will be required for all profiles. Should vendors be allowed to advertise compliance to, say, 1003.11 (transaction processing), if they've implemented that standard on an underlying system that doesn't support lower-level POSIX calls like *fopen()*? (There isn't a standard for 1003.11 yet, but you get the idea.)

One argument advanced for requiring 1003.1 is that it will force vendors to adopt it more quickly. I don't think that 1003.1 needs any help in that area. Another is that requiring compliance will ensure that vendors who

want to advertise POSIX-compliant systems are following the general POSIX direction and not just implementing the simplest standard so they can claim that their system implements "some POSIX."

An argument made against the requirement is that it may damage implementations. For example, real-time systems may lack even a file system, and may want a very limited subset of the POSIX interface to keep the implementation as small as possible. If all of 1003.1 is required, vendors may have to add costly and unnecessary features just to claim POSIX compatibility.

When the dust settles, I think 1003.1 will be strongly suggested but not required, because 1003.1 is a pretty arbitrary subset of any list of "required system interfaces."

[Editor: We disagree. 1003.1 is a set of applications programming interfaces carefully chosen to be necessary and sufficient to make an operating system UNIX-like for the C programmer. Providing standards for a UNIX-like operating system should be the goal of the POSIX standards, and attempts by vendors uncomfortable with UNIX to dilute the effort should be cut off at the pass.]

[Author: POSIX must evolve a set of independent standards that have UNIX as their heritage. POSIX standards are all evolving as UNIX-like standards. Why discourage a vendor from implementing some subset of UNIX-like standards just because the vendor is not ready to provide a complete 1003.1 implementation?]

Want to go to a POSIX meeting?

This was my first POSIX meeting. In case you haven't been and are thinking of going, here are a couple of things you'll want to know.

New people are welcomed. As a practical matter, it helps to stick with a group for the entire week. It's tough to understand much if you come into an advanced discussion cold. It would help if each group summarized its purpose and listed the big issues at the beginning of each meeting, to get everyone in the proper

frame of mind. Still, you'll be granted a sort of first-time armor to protect you when you ask naive questions or need clarification. For extra insurance, use the phrase "I will take an action item..." often.

Report on IEEE 1003.1: System services interface

Mark Doran <md@inset.co.uk> reports on the January 8-12, 1990 meeting in New Orleans, LA:

Most published standards inevitably require updating through corrective supplements. P1003.1 has now reached that stage. The first supplement, P1003.1a, is at an advanced stage and was the central issue at the New Orleans meeting.

Also on the agenda were:

- further talks with the group working on transparent file access;
- more language-independent-specification work; and
- a run-through of the material in the embryonic second corrective supplement, P1003.1b.

P1003.1a Ballot Resolution

The first corrective supplement to IEEE 1003.1-1988 (POSIX.1) is intended to correct errors and oversights in the first publication with a view to clarifying the intent. It is definitely not meant to introduce new functionality or behavior into the standard.

This work received its second recirculation ballot during the week preceding the New Orleans meeting. Donn Terry, chair of P1003.1, hopes that one, or at most two, more recirculations will bring the document to a publishable state. Accomplishing this will send it off to ISO, who will ballot it for six months. (That's right, six *months*; an IEEE recirculation ballot lasts ten days - does this seem a little lopsided to you?)

The details of the content of P1003.1a and its ballot resolution are long and complex, so I won't repeat them here. However, there is one issue worth raising which the ballot brought to light. On the subject of changes relating to the support of split baud rates, one balloter commented:

While we do not agree with the direction this issue is obviously taking, we will abide with the decision of POSIX insofar as split baud rates are concerned.

But we would be remiss in our responsibilities if we did not express our complete outrage with the provincial attitudes expressed by a number of the ballot comments we have had the pleasure to review during this recirculation period.

Split baud rates ARE NOT uncommon with a great number of the community of users of these standards. Obviously, many of those submitting ballots have not had the opportunity to consider the needs or requirements of users outside their own immediate view. We abhor such a limited, irresponsible scope, especially considering the nature of the tasks we are charged with resolving. It is our hope that we shall do better in the future.

Only rarely are standards meetings graced with such florid language, and the balloter clearly has at least the tip of his tongue in his cheek. However, there is, underneath this bonhomie, a serious point being made.

The IEEE is an ANSI-accredited standards-developing body, responsible for making standards pronouncements for use in the USA. All POSIX standards are being passed to ISO for potential adoption as international standards. The POSIX steering committee (SEC) has declared that POSIX would like to think of itself as an internationally accessible organization. If POSIX is indeed to be internationally accessible then the attitudes of some of those who attend will have to change. Take for instance, the split baud rate issue mentioned above.

Working group discussions revealed that split baud rate support, though a non-issue in the USA, is important in Europe. (The reasons for this stem from the way the PTTs in Europe structure their charges for communications lines - PTTs are Europe's little AT&T

;login: 15:3

at any problem. Delve deeply into POSIX and ANSI C internationalization issues, and you'll always discover topics that the committees have not yet dealt with. This is not a criticism of the internationalization standardization groups; much work is still needed and solutions to many problems remain elusive. In the *uuencode* example, we felt the output of *uuencode* should be code set invariant (i.e., *uuencode* on an EBCDIC system should produce the same results as *uuencode* on an ASCII or ISO 646 character system). To achieve this, ' ' through ' _ ' must be expressed as 0x20 through 0x5F and "begin" must be expressed as 0x62 0x65 0x67 0x69 0x6E (the hex equivalents of 'b' 'e' 'g' 'i' 'n' in ASCII). POSIX appears to offer no standard way to convert a file from one code set to another.

Attendance at the UPE working group was, again, relatively small – around a dozen people. One reason is PAR proliferation. Most companies cannot afford to send one committee member to each working group. (I, for example, also had to cover TFA, POSIX.1b, and the internationalization efforts.) [Editor: Readers should note that that being spread thin didn't stop Randall from turning out a clear, thoughtful report. Thanks, Randall.] Another reason is that there is less enthusiasm for the UPE than for Dot 2 Classic. Even Hal Jespersen has said that "...basically the NIST put our feet to the fire to do the UPE."

Some people want the UPE to include an EMACS editor description as well as one for *vi*. Unfortunately, although there was talk of an EMIN proposal, none was submitted to the working group. If you EMACS fans want it included in the ever-expanding UPE, then submit a proposal. [Editor: Listen up, folks. He's serious.] (Of course, some devotees feel that standardization would be inappropriate for an extensible environment like EMACS.)

"Revision/Source Code Control Software" is a much-shuffled area of future standardization within the overall POSIX.2 PAR. Fearing another *Tar Wars*-like clash between fanatic supporters of of SCCS and RCS, the topic was removed from Dot 2 Classic and deferred to the UPE. The Source Code Control System

(SCCS) is the original UNIX source code control system which was implemented in the mid 1970's, modeled after mainframe systems of the time. The more modern (no bias here...) Revision Control System (RCS), by Walter Tichy of Purdue University, claims to have improved on SCCS. Each has its proponents; SCCS appears to have a stronger following because of commercial support by vendors, but RCS appears to have a more devoted *underground* following. The working group is divided between those who want either SCCS or RCS and those who want neither, arguing that source control is a vendor-specific application. Unfortunately, the UPE working group has had problems resolving the controversy and Hal Jespersen has proposed that POSIX.2c (yes, you heard it right, .2c) be assigned as a PAR for working on this topic. (What happened to .2b? POSIX.2b is the working group that will prepare revisions and clarifications of Dot 2 Classic – which isn't even finished balloting.)

Report on IEEE 1003.3: Test Methods

Doris Lebovits <lebovits@attunix.att.com> reports on the January 8-12, 1990 meeting in New Orleans, LA:

Dot three's job is to do test methods for all of the other 1003 standards. This was the working group's fifteenth meeting. We reviewed the ballot status of P1003.1 test methods, worked on P1003.2 test methods, and created a steering committee.

Review of ballot status and Dot two verification

The P1003.3 standard will consist of several parts: Part I is generic test methods, and part II is test methods for measuring P1003.1 conformance, including test assertions. Part III of P1003.3 will contain test methods and assertions for measuring P1003.2 conformance. As other P1003 standards evolve, they will be covered as separate parts in the P1003.3 standard.

Each day was divided into two sessions: mornings, we did technical review of parts I

and II, afternoons were spent writing assertions for part III. AT&T, NIST, OSF, Mindcraft, IBM, DEC, HP, Data General, Cray Research, Unisys, Perennial, and Unisoft Ltd. were represented. [Editor's complaint: I see no user representation at all.]

It took twelve meetings of the previous P1003.3 working group to prepare the draft that is now balloting. The technical review for the Draft 10 ballot was completed. Draft 11 was re-circulated late February 1990 and closed March 23, 1990. The balloting group is approximately ninety members. X/Open submitted a list of assertions for P1003.1a. This list was included as an appendix to Draft 11. Balloters were expected to review this appendix as part of their ballot. We anticipate an approved P1003.3 standard in the third quarter of 1990.

This is the third meeting for developing a verification standard against the P1003.2 standard. The P1003.2 assertion writing and review were done in small groups. Some of the assertions were based upon P1003.2 Draft 9.

A steering committee and some new officers.

The chair, Roger Martin, instigated the creation of a test-methods steering committee to help alleviate the increasing dot-three work load all the other proliferating groups are creating. The committee will coordinate the activities of all test-methods groups, monitor the groups' conformance to test methods, and write and approve Project Authorization Requests (PARs). Membership will be dynamic, limited to four to six, and new members will be chosen based on long term commitment, new ideas, and technical/managerial skills. Roger suggested an initial makeup - Roger Martin (NIST, Steering Committee Chair), Anita Mundkur (HP), Andrew Twigger (Unisoft), Bruce Weiner (Mindcraft), and Lowell Johnson (Unisys) - and the working group approved. It's a non-controversial mix of established P1003.3 members.

The Standards Executive Committee (SEC) has approved both the committee and its

membership. Their first assignment is to document procedures.

In addition, new officers were chosen for the P1003.2 Test Methods activities. Ray Wilkes, of Unisys, is Chair, Jim Moe of Cray Research, is Co-chair, Lowell Johnson of Unisys is Secretary, and Andrew Twigger of Unisoft Ltd is Technical Editor.

Report on IEEE 1003.4: Real-time Extensions

Rick Greer <rick@ism.isc.com> reports on the January 8-12, 1990 meeting in New Orleans, LA:

1003.4 goes to ballot

The big news in 1003.4 is that some of it is ready for balloting. The current draft is a 330-page, eclectic collection of real-time features. Some (e.g., asynchronous event notification) address significant deficiencies in the dot-1 base, but others (e.g., IPC message passing) seem to be of limited value. It remains to be seen whether the limited applicability of some of the proposed features is enough to shoot down the entire ballot.

One area that may cause trouble is the shared-memory model in the *Language-Specific Requirements* section. While this language-independent model addresses a real need - serialization of reads and writes in the presence of simultaneous updates to a common store - it does so rather formally; people uncomfortable with formal mathematical models may be put off by it. The fact remains, however, that both dot 1 and the ANSI C standard failed to address this problem, which is critically important in shared-memory multiprocessor architectures.

Threads

The threads proposal is only an appendix in the current draft, and won't be subject to formal ballot. Though there were too many loose ends in the threads proposal to send it to

:login: 15:3

ballot in this round, most of them were tied up in New Orleans. We should have a ballotable draft ready after the April meeting.

Meanwhile, the active membership in the threads "small group" is changing. Representation from the Ada community has grown from two to six; almost a quarter of the active membership is now familiar with Ada and its multitasking model. Most threads people, including me, are also becoming active in the new multiprocessor study group.

Discussion within the multiprocessor group promises to be quite lively, since the threads group's more contentious issues (e.g., signals) were skirted by defining high-level interfaces, leaving details of low-level behavior unspecified. The multiprocessor group, on the other hand, must deal with the low-level behavior of multiprocessor configurations, and many of the old arguments have already resurfaced (e.g., should signal state be maintained per-process or per-thread?). Using high-level interface specifications to dodge low-level implementation issues does have its problems, though. People unaware of more subtle implementation issues tend to view new high-level interfaces as unnecessary complications. It's difficult to convince them that, even if consensus could be reached regarding the behavior of primitive functions, we would still need high-level interfaces (or rigid coding disciplines) to guarantee that independently developed routines use primitives consistently when addressing common problems. The real sticker here has been how to asynchronously terminate a thread and cause it to execute cleanup code. Everyone agrees that this is necessary. Some members, particularly those from AT&T/USO, feel that the best way to provide this facility is by minor enhancements to traditional UNIX signals, but most of the group feels that the best way to deal with notorious signal races in a uniform, language-independent manner, is to adopt a high-level interface, modeled after one used by DEC/SRC.

1003.4 turns into .4, .4A, .4B, .4C, and .14

There are three other major, ongoing efforts in dot 4: language-independent specification of the real-time extensions, identification and specification of other important non-threads, real-time extensions that didn't make it into the current ballot, and specification of a real-time application profile. The first is farthest along, but none is anywhere near completion. Recognizing that these efforts were separate from the current proposal, and from one another, the working group submitted four new Program Action Requests (PARs). The Sponsor Executive Committee (SEC) approved all four, and decided that the application-profile effort was so distinct that it needed a new number. The working group's five PARs are now:

the current ballot	1003.4
threads	1003.4A
language independence	1003.4B
further real-time extensions	1003.4C
real-time application profile	1003.14

**Report on IEEE 1003.7:
System Administration**

Martin Kirk <mkirk@axion.bt.co.uk> reports on the January 8-12, 1990 meeting in New Orleans, LA:

The System Administration working group is developing portable interfaces for administering computer systems, which will provide traditional systems-administration functions such as managing users, file systems, and devices.

The working group began with a base document similar to the draft System Administration FIPS produced by NIST in September 1988, containing a set of commands based on existing functionality. It addressed only the single machine case, and the group quickly saw that it formed an inadequate basis for extension to networked systems.

Three competing models were advanced to cope with heterogeneous networks. All three assumed that there would be a standard interface, but differed in the scope of the underlying administrative database and the degree of interoperability. To update a network of 100 systems, supplied by five different vendors, the three models had:

1. one database per system, requiring any operation to be performed 100 times
2. one database per vendor, requiring each operation to be performed five times
3. one database for the entire network requiring each operation to be performed only once.

The working group chose Model 3, which offered the greatest interoperability, the most benefit, and the biggest technical challenge. The working group also chose an object-oriented approach. [Editor: USENIX can take some credit for this, having prepared a whitepaper that recommended precisely this approach.]

Because system administration is closely related to network administration, in that both are concerned with managing objects distributed across a heterogeneous network, the group adopted an object template based on the work of the OSI Network Management Forum. The template uses Abstract Syntax Notation One (ASN-1), to specify the attributes and characteristics of objects.

Currently, the group's major task is to develop object class definitions. Some of the object classes, such as the user object class, seem relatively straightforward, with attributes such as login-name, numeric-uid, group-id, home directory, and login shell. Others, such as the device object class, introduce major questions: How far is it appropriate to go in defining sub-classes such as disk-devices and tape-devices?

The standard will not specify implementations. Information about a user can be stored in whatever fashion seems appropriate: in a traditional place, such as */etc/passwd*, or in a database.

When the object-class definitions are complete, the next task will be to specify both a command-line interface and a programmatic interface to manipulate the objects. The latter will have both a language-independent specification and a C-language binding. All objects will support a core set of four operations – *create*, *delete*, *set-attribute*, and *get-attribute* – and probably a fifth to check consistency. In addition, there will be operations specific to particular object classes, such as a *mount* operation for file systems.

I am happy with the general approach, but there may be trouble ahead on the command interface front. At present, this is the canonical form:

```
<object> -o <operation> <attributes>
```

such as

```
user -o add name=jsh,uid=423,group=editors
```

or something of that general style. I expect that there will be complaints once it sinks home that this removes old favorites such as “mount” from the system administration canon.

Though the standard is designed for heterogeneous network administration, the working group has not really tackled interoperability. Someone must address this critical area, but it may ultimately be the IEEE TCOS networking groups.

Dot seven is currently aiming at a mock ballot in 1991, and a full ballot in either 1992 or 1993.

Disclaimer: The views contained herein are my personal opinions and do not necessarily have any relation to those of my employer.

Report on IEEE 1003.8: Transparent File Access

Jason Zions <jason@cnd.hp.COM> reports on the January 8-12, 1990 meeting in New Orleans, LA:

;login: 15:3

1003.8 breaks up

The networking work has been reorganized; what was one committee is now five. At this meeting, the Sponsors' Executive Committee (SEC) approved all the networking Project Authorization Requests (PARs) and forwarded them to the IEEE Standards Board for final approval. In the past, 1003.8 was responsible for half-a-dozen types of networking issues. From now on, 1003.8 will restrict itself to transparent file access (TFA); the other work will be distributed to four new groups. The new structure is:

1003.8 TFA
Transparent File Access

1003.12 PII or P2P
Protocol Independent Interfaces, or Process to Process

1003.13 RPC
Remote Procedure Call

12xx PDI
Protocol Dependent Interfaces, a.k.a. s-IOSI
FTAM and ACSE

12yy NS/DS
Name Spaces and Directory Services, maybe
X.500

The SEC tentatively assigned 1200-series numbers to NS/DS and PDI, because they intend these standards to apply to any operating system, not just one that's UNIX-like. (There's one exception: NS/DS must identify the name spaces required by the 1003 standards and determine some means of managing them.)

TFA decides what to do about NFS

The meeting was a landmark for TFA. Until now, no consensus on overall direction had been achieved. We spent a great deal of time discussing the philosophy and goals for a Full TFA and Subset TFA, but no common understanding had been reached in the minds of all members; we wandered between extremes of, "Full means 1003.1!" and, "But NFS sure seems to be good enough for users; after all, they're still buying it."

It became clear that some agreement had to be reached for progress to be made. Many TFA attendees had never worked on a POSIX committee before and didn't quite understand the POSIX consensus process, but after a joint meeting of 1003.1 and TFA, the exact scope and structure of work were finally hashed out. The group's work items are described below.

1. Full TFA

This piece will contain minor additions and changes to 1003.1-1988 to specify its behavior when operating on remote files. Work will include extending already-defined interfaces (e.g., new *stat()* information), defining new errors, defining failure and recovery semantics, and so on.

Semantically, a remote file accessed under Full TFA will be indistinguishable from a local file. A strictly conforming POSIX application will run completely unaltered in a Full TFA environment.

2. Subset TFA

This piece will define both a core subset of 1003.1-1988 that can work correctly over a variety of remote-file-access protocols ("the Core") and a number of additional optional feature sets. The specification will form additional text for IS 9945-1 (ISO's version of 1003.1).

The intent is to have Subset TFA work on the widest variety of protocols consistent with a useful Core; if a remote-file-access protocol is so constraining that any Core based on it would be too small to support useful applications, it will be excluded.

FTAM, the International Standard File Transfer and Access Method (IS 8571), will shape decisions about what will go into the Core, for a variety of reasons.

- It is the weakest common mechanism for remote file access.

- The standard has little chance of success at the ISO level unless it is clearly cognizant of FTAM.

- Nothing weaker than FTAM is likely to prove useful to application writers.

- People are clamoring for a simple interface to FTAM; the open/read/write/close style of Subset TFA meets that need.

The difference in functionality between the Core and Full interfaces will be divided into blocks of capabilities (the “feature sets” mentioned above), which might be provided by other commonly used file-sharing mechanisms. A Core-conforming application will be able to inquire (via *pathconf()*) what functionality, over and above the Core, is available on a per-file basis, and alter its behavior accordingly.

The Core will meet an expressed need to know “what doesn’t work right” over common file sharing protocols. For example, Sun might define NFS’s functionality in terms like, “NFS provides Core Subset functionality, plus the `_PC_LOCKING`, `_PC_DIRECTORIES`, and `_PC_TIMES` capability sets.” An application programmer could use such a specification to determine exactly what features of 1003.1-1988 were safe to use in an NFS environment.

This scheme also permits continued development of remote-file-access protocols. Any mechanism that supports at least the Core will conform to the standard. This encourages vendors and researchers to develop mechanisms that combine the Core and its options with other advantages (very high performance, very high robustness, good behavior over WANs, etc.), while giving users a well-defined interface for applications that will work in all such environments.

3. A Data-Stream Encoding (DSE) supporting the Full TFA Interface

This will provide the mechanism necessary for interoperation of client and server systems. 1003.8 will only develop the encoding; no binding to any particular protocol stack or suite is planned. (Such bindings will be done by working groups chartered to develop profiles to satisfy particular needs.)

Work on the DSE will probably not begin for at least another six months. There are now

two existing proprietary mechanisms that provide the appropriate functionality: SVR4 RFS and the Andrew File System (AFS v.3 from Transarc). The committee hopes at least one (if not both) of these products’ DSEs will be released to POSIX for standardization. If both are, there will probably be a gun-battle over which to base the standard on.

There was good progress on the first two work items. The group hopes to have a meaningful draft available by April, and would like to go to ballot by the end of the year. This quick ballot will help compensate for the small working group by bringing major ballot objections to the surface early. (Much coordination with other 1003 working groups, especially 1003.1, will also help.) The balloting process will probably be quite lengthy: on the order of 12-15 months.

Report on IEEE 1003.11: Transaction Processing

Bob Snead <bobs@ico.isc.com> reports on the January 8-12, 1990 meeting in New Orleans, LA:

Context

Our charter is to develop an application profile for POSIX Transaction Processing (TP). We’re wrestling with both the content of our profile and the idea of a profile, since profiles are new to POSIX. [Editor: Jim Isaak reviewed application profiles in the February issue of IEEE Computer.]

The content is influenced by two other TP efforts: OSI’s DTP and X/Open’s XTP. We must handle OSI DTP, just to gain ISO acceptance – a goal of all the 1003 efforts. In theory, XTP is just another proprietary concern. In fact, XTP’s ongoing deliberations are currently confidential. Moreover, X/Open isn’t an official standards body so we can’t officially reference XTP in our profile. Nevertheless, XTP will carry considerable weight, since it will be a multi-vendor consensus on how to do UNIX TP.

;login: 15:3

Models

As at previous meetings, we spent much time discussing TP models. For the most part these discussions were based on a snapshot of XTP's model released to non-X/Open members some time ago. Each model we discussed consisted of three or four of the following elements: Application Programs (APs), Resource Managers (RMs, like database managers), Communications Managers (CMs, like TCP/IP), and Transaction Managers (TMs, which enforce the transaction protocol among APs, CMs and RMs). Here, in chronological order, were the major topics of discussion.

We discussed whether a CM might just be an instance of an RM (viewing an instance of a communications protocol or link as a resource), but concluded that attributes of CMs make them fundamentally different beasts (though, to be honest, it's still not clear to me why).

We considered several models based on XTP, but differing from one another in the roles of the CM and the interfaces between the AP and CM. We concluded that each communications protocol would have to have its own CM, and that our model must support multiple concurrently active CMs. A CM, though, is more than just its protocol support. It has to include support for additional functionality required for DTP. We never concluded whether or not an AP should talk directly to a CM, or to a CM via the TM.

Requirements

In the course of the model discussions, it became clear that many of us had different requirements in mind, so we shifted our focus to requirements to try to reach some consensus. Ultimately, we decided that POSIX TP must:

1. be mappable onto OSI DTP,
2. support global (distributed) transactions,
3. support chained and unchained transactions,

4. support a conversational mode,
5. provide data conversion (e.g., ASN.1),
6. ensure that POSIX RPC supports DTP semantics,
7. ensure that DTP can be accomplished through RPC,
8. provide for location independence via directory services, and
9. provide for security of data.

Exercises

We decided to break the modeling deadlock by focusing on the AP/TM interface and ignoring communication. We worked several examples, following ISO DTP services but using an RPC paradigm, and concluded that an API based in RPCs would need at least four services:

- one for a caller to start a transaction,
- one for a callee to find out if it is participating in a transaction,
- one for a callee to abort a transaction,
- one for a caller to commit or abort a transaction.

We also identified the following assumptions for TP via RPC:

- A thread of control (TOC) can be in at most one transaction at any given time.
- If one TOC communicates with another, the latter joins the former's transaction by default.
- No nested transactions are permitted.
- A GTRID (Global TRansaction ID) can be associated with multiple TOCs and multiple RMs.
- A transaction has only one initiator and only the initiator can issue *commit*. Any TOC may abort.

Report on IEEE 1003.12: Inter-Process Communication

Steve Head <smh@hpda.HP.COM> reports on the January 8-12, 1990 meeting in New Orleans, LA:

Overview

P1003.12 is the IEEE POSIX Network Inter-Process Communication (IPC) committee (formerly P1003.8/2). The committee is currently working on two potential interfaces: a detailed interface (DNI) and a simple interface (SNI).

At this meeting, the group arrived at a high-level description of a name-to-address translation facility, and decided the question of XTI versus sockets versus "something else" in favor of "something else." The group began discussing connection setup, and continued discussing SNI. Finally, the POSIX steering committee (SEC) changed the group's name to P1003.12.

There were about twelve attendees.

Detail

1. SNI reviewed

A UC Berkeley SNI proposal is gradually taking shape. The proposal describes both objects and functions that act on them. Some of these objects and functions have analogues in the socket world, while most of the others are composites.

The most recent additions are *sni_save()* and *sni_restore()*. *sni_save()* takes a snapshot of an endpoint and saves it in a string, suitable for passing to a child process through an argument or the environment. *sni_restore()* restores the library state of an endpoint from that string.

The committee has had two goals for SNI. For naive users, it should simplify the networking interface. For vendors, it should allow implementation of interfaces over complex protocol stacks (such as ACSE – or something above ACSE – over OSI-7).

One issue that came up was what the application programmer would target for. If DNI and SNI retain distinct differences, SNI-based applications risk outgrowing SNI's capabilities. One alternative would be to combine DNI and (the current) SNI to allow seamless expansion into protocol-specific hooks, without recoding of applications.

Next meeting, UNISYS is expected to present an alternative SNI proposal.

2. Naming

The group discussed name-to-address translation for DNI in detail, specified an interface at a high level, and intends to pass it to the naming group. The specification is:

given:

```
hostname/"entity"
service/"facility"
type/"context"
protocol or protocol family
```

return:

```
set of {
  address
  any input parameters that were
  completely or partially wild-carded
}
```

SNI might need something similar, but without the protocol / protocol-family / address-family parameter. (SNI is protocol-independent.)

The interface lets applications defer deciding which protocol- or address-family to use until after the query. It will also permit load-balancing, a technique to optimize data-transfer performance over slower interfaces (such as multiple serial point-to-point links).

The group deferred discussing both performance (time and memory) and which input parameters could be wild-carded.

3. XTI versus sockets

The XTI versus sockets issue came up briefly while discussing passive-endpoint functions. The group resolved to incorporate the best of XTI, sockets, and possibly other extensions, into DNI.

;login: 15:3

The group decided not to require full XTI-type functionality, and accepts the risk that porting XTI-based applications to DNI may require source code changes. A potential advantage of this decision is that the standard can leave out the mistakes of XTI and sockets. Also, vendors remain free to supply the older interfaces on the side.

A UCB representative will prepare a new DNI proposal between now and the next meeting.

4. P1003.8/2 → P1003.12

The SEC gave network IPC its own separate number: P1003.12. This change will be formally approved at the IEEE standards board meeting, a couple of months from now.

5. Potential overlaps with P1003.4

For several meetings, both P1003.12 and P1003.4 have been aware of their potentially overlapping coverage of process-to-process communication on a single, local system. Since there should be only one interface for common functions, and any characteristics peculiar to local IPC can be supported by protocol-specific options under DNI, P1003.12's position is that it should handle all IPC. The group has asked the networking steering committee chair, Tim Baker, to relay this position to the SEC.

Future Meetings and Significant Dates:

The Spring 1990 meeting will address SNI/DNI connection setup/tear-down and SNI/DNI data transfer.

Report on IEEE 1201: User Interface

Peter H. Salus <peter@uunet.uu.net> reports on the January 8-12, 1990 meeting in New Orleans, LA:

What's happening?

P1201 purports to concern itself with the user interface. As of the New Orleans meeting, P1201 comprised .1 (Applications Programming Interface), .2 (Graphical User Interface),

.3 (Human-Computer Interaction), and .4 (XLib) subgroups.

Working backwards through these, 1201 has recommended that XLib go to ballot directly, a proposal which seems to have so shocked the SEC that they put off deciding on balloting till April. Steve Jobs told the USENIX audience in Phoenix, in June 1987, that X was "brain-damaged." Whether that's true or not, X has won, and just putting XLib to a vote makes good sense.

1201.3, under the chairmanship of Richard Seacord, has had a number of interesting discussions and presentations (of which I attended several, though not all). The major problem here is that we are nowhere near knowing what the "standard" for an interface might really require. However, the explorations are valuable, and a forum like this can be informative.

This leaves me with the GUI and the API. Both in Brussels and in New Orleans were skirmishes in the GUI wars: battalions of employees of OSF and its member companies arrayed in opposition to those of UI or USO and theirs, with a pair of observers from NeXT and Apple taking and placing bets on the sidelines.

I assure readers that have never attended these meetings that acrimonious backbiting and vituperation are the order of the day in both camps. Though a former employee of OSF, I wouldn't hesitate to condemn the behavior of both sides, but the blame rests elsewhere. Where? In the tourists. See below, but for my money, too many folks like to travel and too many people have caught the "open systems/open standards" bug.

So long as the market remains unsettled about Motif, NeXTStep, OPEN LOOK, and Presentation Manager (to say nothing of Apple's Macintosh interface and IBM's CUA) [Editor: That's "Common User Application", a part of SAA], the meetings of 1201.1 and 1201.2 will serve as tilting grounds, not occasions for useful discussion.

From my point of view, until the market (which means the big boys and the users) has a shake-out, .1 and .2 can only serve as debate platforms or end up recommending standards that are either the intersection of OPEN LOOK and Motif or their union. It might be that .2 can come to some sort of conclusion on the various style guides without .1, but I see the products being waved, not the function banners.

Why is it turning out this way?

All of this is prologue ("The past is prologue," writes Shakespeare in *The Tempest*) to a commentary on the TCOS-standards industry. [Editor: TCOS, the Technical Committee on Operating Systems, is the IEEE organization under which both 1201 and 1003 fall.]

Over the past 40 years, ISO has approved or accepted over 20,000 standards, which concern almost everything imaginable from hockey masks to medical prostheses to the hinging of radar masts on inland-waterway vessels. The standards have arisen in a variety of ways, most emanating from one of the regional or 70-odd national standards bodies. Typically, it has taken from four to ten years to progress from raising a committee to approving a standard. The result of this has been general agreement within the concerned industry prior to the issuance of an international standard. Wall plugs are an excellent example of what happens when the engineers and bureaucrats issue a standard without industry consensus.

I am far from convinced that the ever-increasing number of 1003 and 1201 (sub)committees is productive or useful, and embarrassed and appalled at their continuing proliferation. There are currently at least six or seven standards for diskettes. Do we really need that many for graphical user interfaces? I think not. Might we get what happened in the record industry (i.e., 45s for short cuts; 33s for long works and anthologies) if we wait? I think so.

Moreover, does the standards process really require more than two or three folks per

company? There were 38 in attendance at the ISO/IEC Joint Technical Committee on Application Portability meeting in September (including the secretariat); there were nearly 300 in New Orleans. My perception is that going to a POSIX meeting is a perk. Holding the meetings in Hawaii, New Orleans, and Snowbird does little to dissuade me. The New Orleans host was OSF; the Snowbird host is Unisys. Though the new Unisys is a big entity, I didn't realize they had a site in Snowbird; nor OSF one in New Orleans.

C'mon, lets get back to work, not meetings for the holiday or for the sake of meetings. 1003.1 did good, solid work. Some of the other groups are doing work, too. Partying ain't part of it. Bah!

Report on ANSI X3J11: C Programming Language

Doug Gwyn <gwyn@brl.MIL> reports on the state of ANSI C:

There is now a C standard

After the one appeal of CBEMA X3's approval of the proposed ANSI C standard was eventually voluntarily withdrawn by the appellant, the ANSI Board of Standards Review approved the proposed standard on December 14, 1989. (CBEMA is the Computer and Business Equipment Manufacturers' Association, the organization that sponsors X3.)

No appeals were received by ANSI within the time allotted, so there is now an official American National Standard for Programming Language C: ANS X3.159-1989. The technical content of the ANS is identical to that of the December 1988 X3J11 draft.

The X3J11 technical committee will enter an "interpretations" mode at the March 1990 meeting in New York City. During this phase, the committee will be considering requests for clarification and interpretation of the standard. It is anticipated that Technical Information Bulletins will be issued from time to time when it is felt that clarification of the intent of

;login: 15:3

the standard needs to be published. Such bulletins would not technically be considered part of the official standard; however, they should provide valuable guidance to both C implementors and C programmers.

USENIX Standards Watchdog Committee Update

Jeffrey S. Haemer <jsh@ico.isc.com> reports on winter-quarter activities of the watchdog committee:

1003.0: A Guide to POSIX-Based Open Systems

Dot zero, the POSIX guide group, continues to suffer from bureaucratic inertia. It complains that its forty or so attendees are insufficient to allow rapid progress, yet in a year-and-a-half they've just created a table of contents. Some people think this reflects badly on the group. I think this is completely wrong.

Admittedly, the economics of producing the POSIX guide itself are unfavorable. A large fraction of the attendees are highly-placed or key employees of large corporations and influential organizations. A back-of-the-envelope calculation puts salary expenditures alone, for each one week dot zero meeting, close to six figures. Had the committee delegated the entire task to one or two full-time people, it would be done. The fine overviews UniForum occasionally publishes are proofs by example.

How, then, does dot zero benefit the user community? The meetings give influential people from the most important corporations in the commercial UNIX arena a way to get together in the same room (or after hours in the same city) and discuss the direction of UNIX without risking an anti-trust suit.

USENIX meetings serve a similar purpose for more technical segments of the UNIX community. To some degree, UniForum meetings serve an analogous purpose for other segments of the industry. But where else is there such a

concentration of high-level, UNIX vendor management except, perhaps, at meetings of the Hamilton or Archer groups, or of the board of directors of X/Open? Attendees support POSIX, and influence their companies to become involved. Because POSIX is a good thing, so are dot zero meetings.

1003.1: System Services and C Language Binding

Dot one is well ahead of the rest of 1003; look here to see the future. The initial standard is done, published, and government-approved as FIPS 151-1. The group is now working on supplements, which come in two flavors: nit-picks and corrections (1003.1a) and real additions (1003.1b). But to speak of "the group" is misleading; these two working groups have a strikingly different makeup from the group that created dot one. Many who were passionately and intimately involved in the production of the Ugly Green Book have moved on, either to other committees or out of the standards game. The working groups are now small numbers of hard-core, dot-one devotees. For .1a, this isn't a problem – that's exactly the kind of person needed for nit-picking.

Watch .1b like a hawk, though. Any new functionality, slipped into supplements and appendices, carries the same risks as riders on congressional bills; if it can be slipped in unobtrusively enough, or with the right timing, it can be awful and still ride on the coattails of the main body. Bad deeds done here will both inflict irresistible harm, and diminish the credibility of dot 1.

I recommend resisting any effort to add functionality for which there aren't existing implementations in wide use, and about which there isn't already general consensus. Design-by-standards-committee efforts should be deferred to other more ignorable standards.

1003.2: Shell and Utilities

Dot 2 is still firmly in the dot one mold. Dot 2 Classic is balloting away, and should soon be both done, government approved and FIPS-ified, with a set of test assertions that companies like Mindcraft can sell test suites for. When this is done, a large number of systems will advertise compliance with 1003.1, 1003.2, and X3.159 and provide, for most users, a standard “UNIX.”

Even the controversial UPE is mostly codifying existing practice. Arguments are over places where more than one practice is widespread, for example, source-code control, where partisans of SCCS struggle with partisans of RCS. (Actually, that’s not true. What’s really happening is that the group’s shying away from this area because they’re worried about a struggle. “Tar wars” seems to have spoiled the industry’s appetite for making difficult decisions about contentious topics.)

Parenthetically, I’ll admit to being mystified by the dim view some folks take of the UPE. I actually put programmer portability above program portability, since, when I go looking for new jobs I can’t take my software with me, but do want to be sure that I can still use *vi*. (Of course, most members of working groups are sponsored by vendors.)

The equivalent of .1a already has a name: .2b. Even the bad of dot one is mirrored here. Truly controversial proposals are being pushed off to the as-yet unborn .2c, which should produce a *deja vu* feeling in those already watching .1b. (“But,” you remark, “you always say that.”) And, just as .1 sometimes shied away from real decisions, in order to avoid upsetting anyone, .2 occasionally reacts to vendor inconsistency by proposing solutions that avoid upsetting any vendor by penalizing all users. As an example, the committee proposes requiring a C compiler (good), and naming it *c89* (bad, but I complained about this loud and long last time). An important motivation for the new name is that *cc* already invokes the K&R C compiler on many vendors’ platforms, and specifying a flag to choose one behavior or the other would conflict with someone’s

existing implementation; any given letter is already preempted by some vendor.

I’m not convinced by this argument. I have consulted the Ouija® board in my office, normally used only for project scheduling, and will now predict the effects of this sidestep, if approved:

- In two years, everyone will have a *c89* compiler, to comply with a government FIPS. Shell scripts and makefiles will continue to invoke *cc*, but be less portable than they are now.

- On a few conformant machines, there will be no *cc* command. This will break an enormous number of programs, and solutions will vary from user to user, project to project, and installation to installation.

- On other machines, *cc* will produce one flavor or the other. Most, but not all, machines will link *cc* to *c89*. This will break a variety of things, but not consistently enough to allow a portable solution.

- On some of these machines, flags will make *c89* compile K&R C. The flag will vary from vendor to vendor.

In short, we who do ports will have to keep track of how to invoke the C compiler on each of our target machines; .2 will not have enhanced portability in this area of our work.

Finally, like .1, my unease over a small number of problems stands in stark relief to the generally high opinion I have of the work done by this group.

1003.3: Test Methods

Dot three, a tiny mirror of the overall POSIX effort, is proliferating because it has no choice. It will now have a subcommittee to develop test assertions for each of the other POSIX efforts, and has acquired a steering committee to oversee the subgroups. Whether this is a better choice than having each POSIX committee develop its own test assertions isn’t clear – I see plusses and minuses for each approach. Still, all in all, the group seems to know what it’s doing, and is willing to do it.

;login: 15:3

Dot three isn't always popular; one hears complaints that they come up with interpretations that seem contrary to the intention of the original standards committees. On the other hand, that seems as good a reason as any for their existence. They form a combination system-test and quality assurance group for the other committees, generating all the friction one expects from any such organization.

A dot three member did take the time to divulge an unexpected answer to a question I raised in my last report – what motivates someone to be in dot three? For a few folks, it's obvious: MindCraft employees attend because their company develops and sells test suites. Others are also there because they're really interested in testing. But think: if you want an overview of all of POSIX, what group should you attend? There are three candidates: dot zero, but then you'd have to buy an expensive wardrobe; the SEC, but that group is mostly institutional representatives, officers, and overworked committee chairs; or dot three, which examines each standard in detail as it nears completion. If you're thinking of joining a working group, and want this sort of vantage point, I'm certain the group has plenty of work to hand out.

1003.4: Real-Time Extensions

The real-time group now has five PARs: .4, .4a,, .4b, .4c, and .14. The first of these went to ballot after the New Orleans meeting. Threads, controversial enough to be omitted from .4, has been pushed into .4a. (Things too controversial to go into threads will be pushed into the multiprocessor group, which should be a lot of fun.)

(The remarks below in brackets and with -SP are taken from a response posted to *comp.std.unix* by Simon Patience of the Open Software Foundation; see also the article of comment by John S. Quarterman that follows. -Ed.)

[This is not actually true. Pthreads was never in the draft of 1003.4 proper but was an appendix. After New Orleans when .4 was ready to ballot, pthreads was not and so could

not become a real chapter of its own within .4 and so got its own PAR. It had nothing to do with being controversial. Your parenthetical comment is pure fantasy also. -SP].

The threads subgroup (1003.4A) has attempted to kill the .4 ballot by a block vote for rejection. One correspondent says they are doing this because .4 is no good without threads. (I'm told that two "large, non-vendor organizations" are part of the coalition against the 1003.4 ballot. There is rumored to be a special, invitation-only, threads-strategy meeting by these two groups immediately preceding the Utah meeting. Can anyone confirm this and supply more details?)

[More misinformation here. The Common Reference Ballot was written by a number of people from different organisations some of whom attended the threads group and some didn't. The endorsements for it came from a significantly wider audience than the threads group, some of whom I believe have not been to a .4 meeting either, or at least regularly. The objections were not related to threads except where an interface was impossible to be used in a multi-threaded environment.

The rumor of a pre-Utah meeting is completely overblown. OSF and UI regularly meet, with representatives of our respective member organizations, to discuss technical matters to try and maximize commonality between our two systems, especially at the interface level. The subjects include threads as this is an emerging technology area, but it is certainly not restricted to threads. As the people involved in this also attend POSIX meetings, it is natural to take advantage of the fact that we are all going to be in the same place. The meetings take place regularly and more frequently than POSIX meetings. We think this level of cooperation is the sort of thing the industry would expect us to do, especially the end user community, rather than indulge in the UNIX wars that are restricted to the Trade Press. -SP]

University of California's Computer Science Research Group (the folks who bring us Berkley UNIX) is also voting against the .4

ballot as a block. This stand has nothing to do with the lack of a threads proposal; the vote objects to the working group's addition of completely new and (their words) "lame" features to UNIX. An amusing twist, this. To a traditional standards activity, one vendor block voting against another; POSIX adds one research group (CSRG) voting against another (.4).

[I believe that this was just an endorsement of the Common Reference Ballot mentioned above, which was submitted by someone at Berkeley. -SP]

The threads group itself is divided over whether they are doing an interface to OS-kernel services or an applications library. They are also divided about whether they are doing an interface to language-independent, concurrent programming services, or just a C-language extension.

In general, .4A seems to be a small core of activists pushing ahead with a clear agenda, with an opposition that complains but appears incapable of putting together a detailed unified counter-proposal. Both the rush to go to ballot, and the move to tie success of the rest of 1003.4 to threads, should be causes for scrutiny.

[I can't think where you get this idea from. There is no desire that I know of to tie threads to the rest of .4. The people involved are highly motivated and think that the time is right to standardize on a thread interface before the industry become too divergent. It is felt by many people that there is enough experience in the industry and academia to write a good usable standard and are trying to do so. -SP]

Interestingly, if threads are forced back into the base .4 standard, it may end up causing another problem. The ACM's ARTEWG (the special interest group on Ada's runtime environment working group) is likely to vote in a block against 1003.4 if it contains a threads proposal that does not support Ada in a natural way.

[This is not likely to happen as I said above. The threads group are talking to the Ada people (constantly it feels like :-)) and it is hoped that when the draft is ready for balloting most of the Ada folks will be happy. There is a problem with scope which has never really been properly defined with respect to Ada, especially Ada runtime.

Your overall tone was one of suspicion that there is a subversive plot going on and that half of POSIX is being taken over by a small number of people in the threads group. This is clearly ridiculous as it could never happen; the consensus process prohibits it. -SP]

The Ada folks are concerned that there be an underlying, OS-level model of concurrency consistent with both the C-threads and Ada tasking models. This seems especially important to them if Ada applications want to use standard services written using C libraries which are implemented using C-threads (e.g., windowing and database access). Such a model would also be important for support of Ada compilation systems, which are typically produced by independent software houses to operate on a variety of operating systems and machine architectures.

Dot 4b is a language-independence effort. What's interesting here is that real-time was one of the groups that the SEC grandfathered out of the requirement that POSIX standards be language-independent. (Other exemptions included other standards well along, like .1, and standards that were intrinsically language-dependent, like .9, FORTRAN bindings). Despite that exemption, real-time may be the first group to write a language-independent binding.

Real-time also has PARs for .4C, a place to put stuff that didn't make it into .4 (i.e., .4 is to .4C as .1 is to .1B), and .14, the real-time profile.

Language-independence Study Group

I want to straighten out something I was confused about in the last summary report. (Thanks to Jeff Kimmel, of the language-independence study group, for taking the time

;login: 15:3

to explain this.) Language-independence is a sop to ISO. Two prices we pay to gain rapid international approval of the POSIX standards are an agreement to hand ISO standards formatted in their preferred style, to which end the IEEE is providing editorial assistance, and a commitment to a direction ISO intends to take for all its standards: language independence.

And, to clear up another misconception, Steve McDowell worried, in his last .7 snitch report, that ISO requires language-independent specification languages to themselves be standardized. This would force POSIX to use something frightening like VDL. Fortunately, that turns out only to be true for *formal* specification languages: languages from which one can derive correctness proofs. ISO isn't interested in proofs, only in divorcing specifications from specific programming languages. They don't want to give an unfair advantage to languages in which the things being standardized are likely to be initially implemented, like C or FORTRAN, over more international languages, like ALGOL-66. In other words, POSIX will probably produce specs in ASN.1 or even English. (That's "language independent." Get it?.)

1003.5: Ada Bindings

Dot five didn't officially meet in New Orleans, partly to give .5 members more time to attend other groups. Dot five members kept saying things to puzzled members of other committees like, "We're not really meeting," "I'm not really here," and "Well, I *am* here, but don't tell our chair, Steve Deller." One member graciously volunteers this short, but timely, update:

"The Ada binding group (P1003.5) just finished an intensive working meeting at Florida State, in Tallahassee. The meeting went very smoothly. We resolved all the issues brought up by the recent mock ballot, and expect to have a revised draft ready for the April POSIX meeting. That draft is supposed to be given some finishing touches at the meeting, and then sent out for formal ballot."

1003.8: Transparent File Access

As expected, what used to be dot 8 has split into several groups. There was a meeting on the last day, in which chairs of each of the newly-formed POSIX networking-related groups gave status reports. At that meeting, one attendee objected that the models and APIs that come out of these groups increase portability, but do little or nothing to ensure interoperability. Surely, networking standards should have interoperability as a primary goal, he complained. While the current groups don't have solving this problem as part of their charter, many attendees agreed that the complaint is valid, and something should be done on this front. Keep your eye on this problem.

While the other subgroups have new numbers, the group standardizing transparent file access (TFA) retains the dot 8 name.

Six months ago, TFA was torn between a faction wanting to canonize NFS, and another insisting on something that supports full dot 1 semantics. Now, the group has achieved consensus. They'll provide several standards: a core subset with which FTAM will comply, a set of extensions to the core with which various versions of NFS will comply to various degrees, and a full standard that will support full dot 1 semantics. This compromise recognizes the *de facto* international standard without sacrificing a commitment to dot 1.

1003.9: FORTRAN Bindings

Dot 9 is in the middle of editorial cleanup in preparation for balloting. Emphasis until now has been on content, so the draft developed with many styles and formats. Much of the last meeting was spent trying to even things up.

Since things are drawing to a close, you might expect meetings to be sedate. If you read the .9 postings in *comp.std.unix*, you'll know that's not true. When I walked in on the .9 meeting the group was in the middle of a heated discussion. Someone had proposed adding several functions to increase portability of FORTRAN programs. One specific example

was a function that would return the maximum REAL for the implementation. While there is little question of the utility of such a function, there were two sorts of illuminating objections:

1. Some members of the group objected that the standard was not intended to increase portability of FORTRAN programs, only to provide FORTRAN bindings to the .1 standard. (Indeed, unlike .5, .9 makes no attempt to be a stand-alone document. It freely uses pointers into .1.) Others countered that the section being discussed corresponds to section 8, *Language-Specific Service for the C Programming Language*, of the Ugly Green Book; that the group's goal *is* improving application portability; and that additions that further that goal are completely within the group's charter.

2. One member objected strenuously that many of these additions required REAL support. I was utterly mystified by this objection, until the group patiently explained that, though .9 is an F77 binding, it won't require F77 compliance, and won't use all the features of F77. For example, these new functions were .9's first use of REALs. What the member was objecting to was that without the added functions, a vendor could advertise .9 compliance with an integer-only FORTRAN compiler. Adding these new functions would require that the vendor's FORTRAN compiler actually handle REALs. Think about that.

The ultimate (and, in my opinion, correct) decision was to add the functions, but you can see that there are interesting philosophical divisions in this group. Similar divisions actually exist in all the groups, but the discussions in .9 seem to be more direct and get resolved more quickly. Chalk it up to more programmers, fewer politicians.

1003.10: Study Group on Supercomputing

Dot ten has two subgroups, Profile and Batch, each working on a document.

The Supercomputing Application Environment Profile specifies a set of standards, along with options and parameters needed for supercomputing application environments. The

current draft, 1.0, is still rough, but specifies most of the required standards. At the April meeting, the Profile subgroup will hold a joint session with dot 0 and the other profile working groups (.11, .14, and the multiprocessing study group) to discuss profiles.

Batch Extensions for Portable Operating Systems describes a standard batch management system based on NQS (the Network Queuing System, available from NASA Ames). The batch subgroup began its work within /usr/group's supercomputing working group, has been meeting eight times a year, and is now on draft 1.2. When complete, the document will specify required extensions to POSIX, including interfaces for checkpoint/restart and resource control, utilities for job submission/management and batch system administration, and a network application-level protocol. The subgroup has submitted a PAR for the batch work, which the SEC will consider at their April meeting.

1003.11: Transaction Processing Study Group

Good news in transaction processing. Dot 11 has been trying to work out what model of transaction processing to adopt. Because many committee members are also active in other committees specifying other TP models, the committee had a running start, but progress has been slowed somewhat because there are at least three camps: those who favor the ISO model, those who favor the X/Open model, and those who believe that discussion of concrete models is premature.

Part way through the New Orleans meeting the committee took a break from modeling to explore what an API to a transaction processing system might look like. This, finally, provided a fairly uncontentious topic on which all members could collaborate, and the committee seems to have been able to generate real agreement rather quickly. Success breeds success, and this may smooth the way to find other areas that the committee can make progress.

One warning: working out a sample API may serve only to clarify the committee's

;login: 15:3

thinking about the requirements of their application profile, but I wouldn't be shocked to see the committee eventually submit a PAR for the work. If that happens, ask yourself whether the committee should be designing APIs for an area where there isn't yet industry consensus.

1003.12: Protocol Independent Application Interfaces

Dot 12, process to process communication, is one of the groups derived from the division of the old dot 8 group. The big news from this group is that they've made a real decision in the struggle between XTI and sockets. The group has decided to invent a new interface, which they hope will combine the best of both and avoid the mistakes of each. This is important. It is the first time since the beginning of the committee (several years ago, counting its origins in /usr/group) that it has actually taken a stand on the question. The issue has come up often in past meetings, but until now been deferred by the group.

On other fronts, the group is still trying to produce two APIs: a detailed network interface and a simple network interface. I worry a bit about having two disjoint interface standards in the same area. Are two standards better than none? (On the other hand, having two raises the possibility of splitting the group into two separate, numbered groups at some later date, a popular POSIX pastime.) Recognizing the danger in this split approach, some members of the group are considering whether it might be possible to specify a single expandable interface.

12xx: Protocol Dependent Interfaces for OSI

This new dot 8 spin-off, chaired by Kester Fong, is looking at protocol-dependent networking interfaces. They'll begin by concentrating on FTAM. I predict this group will make rapid progress, because its composition is dominated by users.

To help prevent its work from being an Aristotelian exercise in abstract design, the group has begun to collect all the examples it can find of applications based on FTAM. If

you have, or know of, any such examples, please pass them on. Kester's e-mail address is FONG%AESv01.GM@HAC2ARPA.HAC.COM.

1201: User Interface

1201 is growing to four groups: .1 (Applications Programming Interface), .2 (Graphical User Interface), .3 (Human-Computer Interaction), and .4 (XLib). This serves as a focus for an interesting philosophical issue.

As many readers realize, there is widespread sentiment outside of these groups that 1201 should, instead, shrink to zero groups – that standards in this area are premature. Even more interesting is that the same sentiment is widespread inside the groups. The level of dissatisfaction does vary from group to group. Out of curiosity, I requested a vote for dissolution at the first New Orleans meeting of 1201.3. Fewer than one-third of the attendees voted to dissolve. This contrasts with a similar vote in Brussels in 1201.2, where nearly half of the attendees voted to dissolve. With this much anti-1201 sentiment, isn't there a way to get the IEEE to reconsider the activity? Apparently not.

At the last USENIX, in Washington D.C., Jim Isaak, the SEC chair, explained to the well-attended standards BOF that there is really no easy way to dissolve a committee. If volunteers show up to staff the working group, follow the IEEE rules, and eventually circulate a ballot that passes, they've created an IEEE standard. This means, if you don't like the idea, you currently have only three options.

1. Join the balloting group and vote any proposal down. Not easy; you have to have a good reason for voting no. Of course, "This standard is premature; the direction of industry is too unclear" may be good enough.

2. Join the working group and filibuster until the direction the standard should take does become clear. (Of course, that would be expensive, and lose you popularity points.)

3. Let the group declare a standard and hope everyone ignores it. This one's dangerous because NIST won't, which means the vendors

can't, which means users probably won't be permitted to, and will, at least, have to carry the code around as excess baggage.

So, I'm curious. If you don't like what's going on here, which do you intend to do? (Okay, I'm not that picky. If you like what 1201's doing but object to some other portion of what Doug Gwyn calls "the standards juggernaut," what are you doing about it?)

X3J11: C Language Standard

Closing on an upbeat note, we have a C standard. What more newsworthy item could you ask for?

From: John S. Quarterman, USENIX Standards Liaison, <jsq@usenix.org>.

The summary report from Jeff Haemer, the USENIX Standards Watchdog Committee Report Editor, is in general just the kind of thing we try to publish. However, there were a few problems with it. In particular, the comments about a supposed block vote against

1003.4 originated by a threads subgroup were inaccurate. There was in fact a common reference ballot that originated with UCB CSRG. It addressed many points throughout the 1003.4 draft document. It was referenced in numerous negative ballots, including several from Institutional Representatives. (USENIX did not reference it in a ballot, but only due to time pressure: USENIX supports it in principal.)

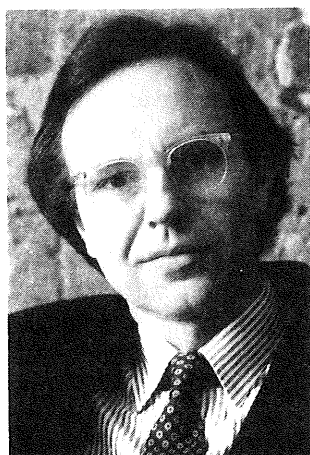
These errors in Jeff's report were due to inadequate review before publication, which occurred because I was out of the country as he finished the report. It was important to get the summary posted on the networks before the Utah standards committee meeting, and turnaround time to substitute reviewers turned out to be greater than anticipated. My apologies for this coordination problem. We will attempt to prevent this kind of situation in the future by more thorough review, including having each section about a specific committee reviewed by the corresponding Watchdog Committee volunteer in addition to being reviewed by me.

Call Doc Strange

Colston Sanger
doc.strange@gid.co.uk

GiD Ltd

with a little help from 'Cheery Bye' Neil Todd



Colston Sanger has left the Olivetti International Education Centre. He is now a sort of senior consultant/tea boy with GiD Ltd and a visiting lecturer in the Faculty of Engineering, Science and Mathematics at Middlesex Polytechnic.

Managing Users

This issue's topic is managing users — you know, *them*, the ones who cause all the trouble, but who also pay the bills. More precisely, the topic is how to assign login names and how to nudge them gently into groups, particularly across distributed filesystems.

Assigning *gids* and *uids*

While it is sometimes sorely tempting to regard all your users as simply *them*, on closer inspection they will often seem to fall fairly naturally into classes or groups. There's the "I'm a programmer, I've got magic fingers" group, the "I'm too important or too busy to read that" group, the "Oh, is that what it does" group...

Seriously, why not group them by function or by the need to share information? In a university, the obvious groups are likely to be staff, research or postgraduate students, and undergraduates. You might want to divide these further by department. You might have `psych_staff`, `soc_staff` and `comp_sci_staff`, for example, for psychology, sociology and computer science staff respectively. Similarly, you might want to divide

the postgraduate and undergraduate students by department, by course or by year. In a commercial organisation, the groups might be `wordpro`, `accounts`, `personnel` and `sales`, where `sales` might again be further divided by region.

When assigning numeric *gids*, it makes sense to increment by more than simply the next available number. For example, you might assign the numeric *gid* 100 to the `wordpro` group, 200 to `accounts`, 300 to `personnel` and so on. Then, when you come to assigning numeric *uids*, `sharon` and `tracy` in the word-processing department can be numeric *uids* 101 and 102, `kevin` in `accounts` can be 201 and `sean` in `personnel` can be 301. The advantage of this scheme is the extra level of redundancy it provides: if your `/etc/passwd` file should one day 'accidentally disappear' without trace or backup, then you have a better chance of sorting out what belongs to whom from the numeric *gids* and *uids* stored in the *i*-nodes (as shown by an `ls -l`) than if you had just assigned *gids* and *uids* randomly.

In System V, users are only ever a member of one group — the group to which you as system

manager assign them when you add them as a user. If they need to share information with the members of another group, probably the easiest and most secure way to do it is to use the `newgrp` shell built-in command to temporarily change group to the other group.

For this to work, however, you have to set things up properly. If you look up the manual page for `/etc/group` you will see that it says that the fourth field in each line is 'a comma-separated list of all users allowed in the group'. It's not actually

```
you:x:204:200:Member of staff:/u/staff/you:
```

(This machine has a shadow password file.)

```
staff::200:
student::300:
```

You try a `newgrp` command:

```
$ id
uid=204(you) gid=200(staff)
$
$ newgrp student
newgrp: Sorry
```

Doesn't work. Now add yourself as a member of the `student` group in `/etc/group`:

```
staff::200:
student::300:you

$ id
uid=204(you) gid=200(staff)
$
$ newgrp student
$
$ id
uid=204(you) gid=300(student)
$
$ newgrp
$
$ id
uid=204(you) gid=200(staff)
```

And back again.

I suppose I ought to mention that Berkeley UNIX has the concept of a group of groups. If you use PC-NFS, however, don't be caught by the 'gotcha' that you are only ever a member of your base group (the one defined in `/etc/passwd`) regardless of whether or not you are filesharing with a BSD machine.

— or else the wording has always been ambiguous and confusing (to me). What the fourth field really is, is a list of users who are authorised to use the `newgrp` command to change to that group.

Let me demonstrate. Assume you are a member of the `staff` group, but that sometimes (for whatever reason) you want temporarily to become a member of the `student` group. Here are the relevant entries in `/etc/passwd` and `/etc/group`:

Naming schemes

There is, of course, a step previous to all this: what if you have more than one clever trevor in your organisation? In any organisation of more than half a dozen people it's a virtual certainty that there will be some who share the same first name. In recognition of this potential problem, some system managers use the initial letters of users' first, middle and last names for login names, perhaps with a common or group prefix. Others don't use personal login names at all: instead, they use logins by job function — `wordpro1`, `wordpro2`, for example. To my mind, though, that's a bit impersonal. If you're trying to encourage people to give up their fear and loathing of computers, it's not exactly helpful. It could also make it more difficult to spot unauthorised use of a login. Certainly in an office environment, there's a lot to be said for using login names that correspond to the names or initials that are used on existing distribution or circulation lists. Whatever you decide, the point is it's worth thinking about a login naming scheme from the outset.

Where to put them

Once you have sorted out the groups, the next step is to decide where to put them. Again, it makes sense to create separate disk partitions for each group, mounting one as `/u/wordpro`, for example, another as `/u/accounts`.

What do you gain by this? Well, you now have the potential to spread your users' files across a

number of disk partitions. It means you can ensure that greedy users in one group cannot hog large amounts of disk space to the detriment of other groups (the downside is that you have to make a reasonable guess as to how large that group's partition needs to be, and you lose the automatic and dynamic allocation of space between groups as users create and delete files). You also have the possibility of load balancing across disks and disk controllers, and can dump or backup the different groups at different frequencies from each other.

Moreover, if you use a distributed filesystem such as NFS, you can restrict the files you wish to export much more easily (remember that export restrictions in NFS are on a per partition basis), thus your fileserver can, for example, export undergraduates' login directories to those machines on which undergraduate logins are permitted, while at the same time denying undergraduate access to staff or postgraduate login directories (which are presumably exported to other machines).

Sharing files between machines

Distributed filesystems are all the rage these days. There are two that are generally available: NFS and RFS. NFS works in a UNIX or heterogeneous environment, whereas RFS is for UNIX only. Since they are both available in System V Release 4.0 (implemented under the Virtual File System — VFS), it's maybe worth discussing how they deal with user and group ids mapped across the entire distributed filesystem.

NFS assumes that you have a flat namespace (strictly, *uid* space) across all machines. That's to say, the assumption is that files with a *uid* of *n* on my machine are owned by the same person as files with a *uid* of *n* on your machine — or, to put it another way, if I can become the user with *uid* *n* on my machine and I can access your machine *via* NFS, then I own all files with *uid* *n* on your machine. The only mapping that NFS will do for you is to map requests that come with a *uid* of *root* (0) into requests coming from the user *nobody* (typically -2 or 32767), thus ensuring that *root* on one machine cannot operate with superuser privileges on another machine. That's it.

RFS, on the other hand, provides you with the ability to map users and groups globally or on a per machine basis. Under RFS, if you never set up

mapping, all remote users will be mapped to a special guest id, represented by an id number that is one higher than the maximum allowed on your system. By default, the maximum number of users and groups on a system is 60000, so the special guest id number is 60001. When a remote user does an `ls -l` of your files, they will appear to be owned by *uid* 60001 or 60002. The 60001 means the file was created by a remote user, whereas the 60002 means the file was created by one of your local users and, therefore, remote users can only access the file if they have other permissions.

User mapping increases the power and flexibility of RFS. For example, you may want to map some or all remote users into particular local users' permissions. If you are the administrator of several machines, you may want to map all *root* logins together across the machines so that you will be able to modify any remote resources mounted on any machine you are working from.

Alternatively, you may want to set up a group of machines to have the same `/etc/passwd` and `/etc/group` files so that when a user creates a file he or she maintains sole ownership of it, regardless of where the file actually resides. With this transparent mapping, you could share resources that require a consistent view of user ownership. For example, you could share your `/usr/mail` directory, mount it on `/usr/mail` on other machines and have one mail directory for the entire set of machines.

Or you may want to map users from one machine in a different way than users from another machine. For example, you may want to map all users from one machine into *uid* 600, from another machine into 700 and from another into 800 so that you can monitor which remote machine's users are creating files within your resources.

How to set up user mapping in RFS

You will need to create a set of 'mapping translation tables'. These tables will be used by your machine to process requests from remote users for access to resources belonging to you that are mounted on their machines.

The command to create translation tables is `idload`. When you run `idload` without any options it does the following:

- reads the rules files to determine how you want to set up mapping
- reads the `/etc/passwd` and `/etc/group` files on your machine and copies those from other machines as required
- creates translation tables.

`idload` has two options: `-n` lets you do a trial run without actually changing the mapping; and `-k` lets you see the mapping that is currently in effect.

You will need three sets of files, as follows. First, you need rules files, *i.e.*, the files `uid.rules` and `gid.rules` located in the `/usr/nserve/auth.info` directory. The information you add to these files tells the `idload` command how to create the translation tables. Second, you need the `/etc/passwd` and `/etc/group` files on your machine. Although you don't modify these files, you will need the information in them. For example, if you map by local name, these files are read to translate the names to numeric *ids*. Third, you need remote `passwd` and `group` files. Because mapping translation tables are sets of numbers, if you want to map a remote user by name, you must have a copy of the `passwd` and `group` files from the remote user's machine. These files should be placed in the `/usr/nserve/auth.info/domain/nodename` directories, where *domain* and *nodename* are replaced by the remote machine's RFS domain and nodenames respectively.

To make the discussion more concrete, here is an example `uid.rules` file:

```
global
default transparent

host rfsdemo.sixnine
exclude 0
map all
```

Essentially, `uid.rules` contains two 'blocks' of rules: the `global` block, which defines the permissions that will apply to the users on all machines for which there is no specific mapping; and one or more `host` blocks, one for each remote machine you want to map specifically. Both blocks are optional.

Within a `global` block, the `default` line can be either omitted, in which case the default 60001 is assumed, or `transparent`, which means

that each user will have the permissions of the user with the same *uid* on your machine (most useful when the `/etc/passwd` files are identical on all machines) or you can use default *local*, where *local* is any local *uid* or name, meaning that any users who are not specifically mapped will have the permissions of the particular *local* user on your machine.

If you want to exclude certain users from having the permissions defined in the `default` line, you can add `exclude` lines. For example, if you use `default transparent`, you may want to `exclude 0` to make sure that remote root users don't have permission to modify files owned by the local root on your resources. You can either `exclude remote-id` or a range of `remote-ids`, as in `exclude 0` or `exclude 0-99`. In either case, the excluded remote user would then only have the permissions of the guest `id` (60001).

You can also add `map` lines to map specific remote *uids* to local *uids* or names. You can either `map remote-id:local-id_or_name` or simply `map remote-id`. For example, if you use the second form, `map 0` would give a remote root the same permissions as the local root.

The format of `host` blocks is `host RFS_domain.nodename` as in:

```
host rfsdemo.sixnine
```

`host` blocks can also include `default`, `exclude` and `map` lines. In a `host` block, `map` can be followed by the additional keyword `all`, meaning that all remote user names should be mapped to the permissions of those users with the same names on your machine.

The `gid.rules` file has the same general format as the `uid.rules` file, except that now you are mapping groups rather than users.

If, when you created the `uid.rules` and `gid.rules` files, you referenced any remote users or groups by name, you will need copies of the remote `/etc/passwd` and `/etc/group` files to put in `/usr/nserve/auth.info/domain/nodename` directories on your machine (note that `map all` maps by name). You'll need to obtain copies of these files by any suitable file transfer method (such as `uucp`), create directories as required and install them on your machine.

You are now ready to run `idload` with the `-n` option. This will print a listing of the mapping rules without creating the translation tables:

```
#
# idload -n
TYPE MACHINE REM_ID REM_NAME LOC_ID LOC_NAME
USR GLOBAL DEFAULT n/a transparent n/a
USR rfsdemo.sixnine DEFAULT n/a 60001 guest_id
USR rfsdemo.sixnine 0 n/a 60001 guest_id
USR rfsdemo.sixnine 1 daemon 1 daemon
USR rfsdemo.sixnine 2 bin 2 bin
USR rfsdemo.sixnine 3 sys 3 sys
USR rfsdemo.sixnine 4 adm 4 adm
USR rfsdemo.sixnine 5 uucp 5 uucp
USR rfsdemo.sixnine 10 nuucp 10 nuucp
USR rfsdemo.sixnine 37 listen 37 listen
USR rfsdemo.sixnine 70 trouble 70 trouble
USR rfsdemo.sixnine 71 lp 71 lp
USR rfsdemo.sixnine 124 colston 294 colston
USR rfsdemo.sixnine 525 demo 693 demo

GRP GLOBAL DEFAULT n/a transparent n/a
GRP rfsdemo.sixnine DEFAULT n/a 60001 guest_id
GRP rfsdemo.sixnine 0 n/a 60001 guest_id
GRP rfsdemo.sixnine 1 other 1 other
GRP rfsdemo.sixnine 2 bin 2 bin
GRP rfsdemo.sixnine 3 sys 3 sys
GRP rfsdemo.sixnine 4 adm 4 adm
GRP rfsdemo.sixnine 6 mail 6 mail
GRP rfsdemo.sixnine 12 daemon 12 daemon
GRP rfsdemo.sixnine 100 staff 200 staff
GRP rfsdemo.sixnine 300 visitor 400 visitor
GRP rfsdemo.sixnine 500 demo 600 demo
```

If these mapping rules are acceptable, you can run `idload` without any options. This will create the translation tables. Finally, run `idload -k` to print the mapping rules that are now in effect.

In the next issue

In the next issue, maybe an article on YP.¹

1. Yellow Pages is a registered trademark of British Telecom in the UK.



AT&T Column

Gill Mogg
gill@uel.uucp

Unix Europe Limited (UEL)
International House
Ealing Broadway
London W5 5DB

Gill Mogg is in Market Communications at Unix Europe Limited.



The guest writer this issue is:

Vijayakumar Vijayaratnam Senior Consultant – AT&T UNIX Software Operation, Europe

Mr Vijayaratnam has been involved with the Information Technology industry for the past 10 years. He has developed and provided consultancy in the area of application/system software, on hardware ranging from PCs to mainframes, running a variety of operating systems. He is currently working with Databases and Transaction Processing systems in a UNIX environment.

Transaction Processing – into the Open Systems Environment by Vijayakumar Vijayaratnam, AT&T UNIX Software Operation, Europe

Introduction

Mainframe environments have always been considered the appropriate medium for the processing of high volume transactions. Open Systems, specifically the UNIX operating system, have achieved limited penetration in this particular field.

Changes in the status quo are, however, under way. Increasingly users have recognised that the characteristics of the UNIX System – freedom of choice in hardware, freedom of choice of database software, compliance with industry networking standards – are equally essential and beneficial in building a transaction processing system.

The wider use of the UNIX operating system, coupled with the technological advancements of UNIX-based data management applications, means that it is becoming common for institutions to look for UNIX based TP systems to address the needs of their data management environments, with

special emphasis on systems that can provide such functionalities as concurrency controls, resource management, scheduling and the prioritisation of tasks, normally found on large proprietary mainframe systems.

What is Transaction Processing?

Transaction Processing (TP) involves computer applications which affect us directly in our daily lives. The classic TP example is a hotel or airline reservation system, in which a person at a terminal talks directly to a computer database about the actual, up-to-the-minute room or seat availability position while you, the customer, wait to see if the reservation (ie transaction) is confirmed. Other applications are in the financial and manufacturing areas.

In computer terms, TP systems are designed to provide the highest throughput in the shortest possible time to a large user base. The basic characteristic of the TP environment is that users

are often performing similar or identical tasks. For example, in a transaction system, many users may be completing the same order entry screen prior to a customer order being entered in the system. In this instance, the different characteristics of the traditional UNIX environment are immediately obvious. Under UNIX, users are performing very different kinds of functions – one may be performing a compile while another is editing a document etc.

A second major characteristic of the TP environment is the predictable nature of the input. A TP system is built on a limited number of input types – add, update, query, delete etc. All such interaction with the system is performed using electronic forms or screens. In a typical environment there may be between one and a hundred such forms. The duration of interaction between the form and the transaction system is very short, often involving a short program in which the input is verified and a response relayed to the user.

TP systems rely to a large degree on a mechanism which prioritises the running of the tasks. One example of this is a customer service application, in which the transaction dealing with retrieving the customer details must have precedence over, say, a task which is performing a management report.

The components of TP system include a transaction manager, a database management system and the business application itself. The transaction manager provides communications and co-ordination in a TP system, the application provides the forms processing and business logic, and the database management system manages the storage and retrieval of data.

Today, there is a tendency towards distributed TP environments, in which users access a number of databases scattered over a wide geographical area. In an environment of this type, there is a need for a system capable of managing tasks in a timely and efficient manner, as well as providing the basic functionalities such as robustness and concurrency controls. The system which handles this function, the TP MONITOR, is thus an integral part of the daily activities of all types of organisations concerned with providing access to a large user base reliant on instantaneous access to stored information.

What is a Transaction?

A transaction is a set of operations (a unit of work) that results in the transformation of the database from one consistent state to another. This, however, is not how the end user sees it. In his terms, the transaction begins with the entry of data on a screen or a form, continues with the scheduling of the transaction type through a TP MONITOR which provides the services to the request, and concludes with a response to the user, often in the form of a report. In a distributed environment, the term transaction describes a unit of work that may be composed of information gathered from a number of physical locations, but represented to the user as a logical unit.

Taking TP into the Open Systems Arena

Scepticism about the capabilities of Open Systems for TP remains strong. Some organisations are vehement in their claims that such processing can be expedited successfully only on proprietary systems with proven track records. On the other hand, moves towards a genuine open systems based TP platform are already well established. Meeting such a challenge depends on two operations which are key to open systems in general – the definition of standards, and the development of products which conform to them.

Standards

In the database realm there is an existing standard – SQL – which defines the interface to the database and allows multiple applications to work on a single database.

TP has been the subject of standards activity for some time. X/Open's first group on the matter was formed as long ago as 1987, and produced a White Paper that Summer. The XTP Group, its successor, has continued with the work since March 1988, with the intention of defining a standard for On-Line Transaction Processing (OLTP). A second X/Open working group, the Data Management Working Group (DMWG), concentrates on the specification of standards for database environments. The goal of the two groups is to define a TP model and a common set of interfaces that will enable providers of TP and DB systems to achieve the objectives of open systems, by providing applications that conform to these interface standards.

Three axioms represent the basis of the TP standards work undertaken by the X/Open committees: interoperability, portability and interchangeability. Interoperability is the capacity to write transaction programs that draw on the resources of several different Resource Manager (RMS), perhaps at different sites, and perhaps produced by different vendors. Customers will be able to perform multi-site update to heterogeneous RMS. The portability of applications is designed to ensure that customers can move their X/Open compliant code to different systems without changes. Interchangeability is the facility to exchange RMS without having to rewrite transaction programs, to support standards that X/Open has previously endorsed (such as SQL and Indexed Sequential Access Method, ISAM), and to permit a compliant system to operate in the framework of other standards, such as the ISO/TP protocols. As a result, the current work undertaken by the committees will preserve, as far as possible, existing RM interfaces.

The work of the XTP committee has so far yielded a model for distributed transaction processing (DTP), which will work well even in a non-distributed environment, and a set of routines, collectively known as the XA interfaces, that will enable RMS to communicate with TMS effectively in a heterogeneous environment.

The DTP model has three functional components:

The Application Program (AP), defines transactions and supervises the actions that constitute a transaction

The Transaction Manager (TM) assigns a global transaction identifier to transactions, monitors the progress of transactions, decides whether a transaction can be committed, and performs failure recovery.

The Resource Manager (RM), such as databases or file systems, uses shared resources as directed by the AP. This service interface may be SQL or ISAM. The TM also calls the RM to declare start, end and disposition of transactions.

For an Open System policy to operate successfully, all the above components must be able to communicate with one other. The application to RM interface is provided by standard SQL, while a vital element of the standard model is the XA Interface, responsible for the TM to RM interface. This incorporates the technically important *two phased commit* protocol

while retaining the SQL interface to the underlying databases. Two phased commit is the key protocol to allow distributed access to a database to make changes (for example a new flight reservation) while guaranteeing data integrity in case something goes wrong in the middle (for example, a tunnel project cuts the phone line). The XA Interfaces have been adopted as the standard for AP to RM interfaces, and are to be published in the X/Open Portability Guide.

AT&T has proposed an extension to the standard model called the Application Transaction Management Interface (ATMI), which gives the programmer transparent access to network communications primitives, automatic data conversion and transaction control operations. This covers the interface between the AP and TM.

All of this activity demonstrates that the move towards an Open Systems TP standard is by no means a recent undertaking. Instead, as a result of the work of X/Open, a superstructure is in place to which products under development at the present time can meaningfully conform.

Products

As we saw, the successful implementation of an open OLTP system depends on the availability of compatible products in three areas: the database, the application and the transaction manager. With the expansion of the UNIX system as a viable operating system for database and business technology applications, state of the art products are appearing, all of which conform to recognised standards. One example of this is the large number of database products from the major independent software vendors, such as ORACLE, INFORMIX and EMPRESS, which already incorporate the SQL standard.

Unlike SQL, an OLTP standard is a new phenomenon, and products are just starting to be announced in the marketplace. There is every reason for confidence that as these products make their impact on the market, the long standing myth of UNIX's weakness as a TP environment will vanish from memory!

TUXEDO – The Transaction Processing Manager for UNIX System V

The TUXEDO (TM) system recently announced by AT&T's UNIX Software Operation meets the X/Open standard and provides an open, distributed TP monitor available to computer

manufacturers and applications vendors. TUXEDO has been designed to exploit the strengths of UNIX System V in networking to allow the distribution of OLTP applications across networks and across multiple servers in multiprocessor systems. These multiprocessors, exploiting RISC chip technology, will provide the high capacity databases serving over a thousand simultaneous users which have up to now required the expensive mainframes for their solutions.

TUXEDO System V Release 4.0 incorporates two

components that can be licensed and deployed separately: the System/T Transaction Manager and the System/D DBMS. Both components incorporate the XTP transaction model, including the XA interface which allows TUXEDO System/T to control transactions for compliant vendor databases while retaining their native SQL interface. It also incorporates the ATMI interface, which AT&T has proposed as a standard application interface for transaction processing.

Article republished courtesy of Systems International magazine.

!%@: A Directory to Electronic Mail Addressing and Networks Second Edition, 1990

The new 1990 edition of !%A: A Directory of Electronic Mail Addressing and Networks, by Donnalyn Frey and Rick Adams will be available in June, 1990. This new edition provides readers with a directory and usage guide to over 130 of the world's research and educational networks, as well as commercial networks. The network information has been updated for 1990, with many new networks added. The directory makes it easy for readers to find networks they can use to reach other people around the world and guides readers in how to use them. It also assists readers in finding someone's email address and sending mail. The book is in an easy-to-use short reference format.

The directory is of use to system administrators who field electronic mail questions, network administrators who work with networks in other countries, researchers who want to get in touch with other researchers, conference attendees with many contacts, and others who routinely send email. Each network section contains general information about the network, as well as address structure and format, connections to other sites or networks, facilities available to users, contact name and address, cross references to other networks, network architecture, future plans, date of the last update, and a map showing the network location. Also included is a three-way index to network name, network type, and country, as well as a list of many of the world's second and third level domains.

This new edition contains:

- information on new networks such as AlterNet, CANET, CA*net, EASInet, InterEUnet, IXI, MFENET-II, TUVAKA, XLINK, and YNET
- updated information on networks that are reorganizing or have reorganized, such as BIONET, ESNET, MFENET, NYSERnet, and OnTyme
- information on networks in the Soviet Union, Eastern European countries, and the People's Republic of China
- networks not in the first edition, such as ATT Mail, KREOnet, and SCIENCEnet
- updates of most of the existing networks described in the first edition which was published in 1989.

This new edition is the most up-to-date guide for directing your electronic mail; it is a real time saver. The book will continue to be updated every ten to twelve months. Readers who fill out the response card in the book have the option of either receiving notification of updates or receiving the updated edition automatically at a 25% discount.

Puzzle Corner

Mick Farmer
mick@cs.bbk.ac.uk



Birkbeck College
Malet St
London WC1
England

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

Hello peeps,

Solution to Puzzle Number 10

If two thirds (40/60) failed on Compilers and three quarters (45/60) failed on Graphics then the *minimum* number failing both is 25/60 of the class (the maximum number may be much higher). In addition, if four fifths (48/60) failed on Networks then the smallest number now failing is 13/60 of the class. This fraction equals 26 students, which means 120 students failed.

Solution to Puzzle Number 11

This is another simple problem solvable on paper, or using something like Prolog. There are three possibilities given the basic facts. The unique solution gives Ms Portuguese offering French and Hungarian.

A complete solution is available to anyone interested.

Puzzle Number 12

A flat roof tile 10" \times 4" \times 3/16" weighing two lbs. rests, with its long dimension along along the slope, on a smooth wooden roof having an angle

of 20 degrees to the horizontal as illustrated in Figure 1.

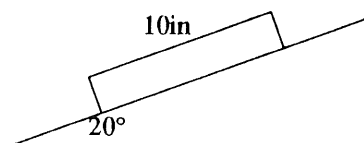


Figure 1 The Creeping Roof Tile

The tile is not attached to the roof, but is kept from sliding down by its friction, the coefficient being 0.5. In the morning of a winter day the tile is at a temperature of 0°F. During the day it is warmed by the sun to 50°F, and at nightfall it is again chilled to 0°F. After such a cycle of heating and cooling, will the tile have moved from its morning position by the end of the day, and if so, how much and in what direction? Assume the tile has a thermal coefficient of expansion of 6×10^{-6} inches per inch per degree and no expansion of the roof.

Puzzle Number 13

I recently came across an old-fashioned toaster that had two heating elements, one on each side, with a door on each side to hold the toast. Thus only one side of the bread could be toasted at a time, but two pieces could be toasted simultaneously. It takes two hands to insert or remove each slice. To turn the slice over it is merely necessary to push the toaster door all the way down, and allow the spring to bring it back. Thus two slices can be turned at the same time, but only one can be inserted or removed. The

time to toast a side is exactly 0.50 minutes. Time to turn over is 0.02 minutes. Time to remove toasted slice and place on plate is 0.05 minutes, and the time to take a slice of bread from the plate and put it in the toaster is 0.05 minutes. The problem is to find the shortest possible time required to toast three slices of bread on both sides, starting with bread on plate, and returning toast to plate. Assume the toaster is warmed and ready to go.

Loads-a-puzzles,

Mick

A New Data Encoding Scheme

*Andrew B Cheese
abc@cs.nott.ac.uk
Julian P Onions
jpo@cs.nott.ac.uk*

*Computer Science Department
Nottingham University
Nottingham.*

Introduction

The computer industry from its very inception has used the binary system to represent all types of information. This system has much to recommend it's use. It is simple to construct hardware based upon this system and all the normal arithmetic and logical operations can be performed upon it.

However, it is the authors opinions that this system is not as optimal as it could be and that some rather simple changes to the method of encoding data can bring about large increases in storage, communications and reliability.

The Idea

The basic ideas stem from the seemingly simple idea of replacing the binary method of encoding by a different scheme. In essence, all that is required is to take the value normally represented by a 1 or true state, and replace this with two 0's or false states. e.g.

decimal	binary	new scheme
9	1 0 0 1	00 0 0 00
13	1 1 0 1	00 00 0 00

At first sight there doesn't seem to be much of an advantage in this scheme of encoding but as the rest of this paper will attempt to prove, the benefits are enormous when applied in the proper way.

This method, whilst not binary is also not strictly unary. It has therefore been christened sesquinary (from sesqui – one and a half).

Applications

File Storage

An intelligent operating system can make great use of the encoding. As an example, a file need not be stored as the complete set of bits. All that is required is for the operating system to keep count of the "number" of zero's in the file. In the case of the UNIX System this would mean that the entire disc would consist of inodes. Each inode, instead of referencing blocks would keep a count of the number of zeros. For large files, double and triple indirection could be applied – see the section below on compression. Obviously, for small files, the single indirection is more cost-effective but with larger files it would pay to move towards more indirection as a saving of space. A flag in the inode could keep count of the number of indirections currently performed.

This scheme does have some overhead in the updating of random access files, in that the operating system must first "unpack" the file, perform the update, and then repack the file. This could probably be done in virtual memory for most operations though.

Networking

In networking, this method really comes into it's own. To begin with, there are practically no bandwidth limitations. The problems inherent in normal communication over serial and phone lines stems from the ability to detect the transitions between two states. Once this transition is removed, and the data is in effect transitionless, the bandwidth of the circuit is only reliant on the speed with which zeros can be pumped down the

line by the hardware (and the rate at which they can be received of course).

Another advantage comes in the standard ethernet environment. Normally an ethernet transceiver must wait for a clear slot to arrive, transmit the packet and detect if a collision occurred, if so it must retry. With the all zero encoding method transmission can take place at any point, there is in effect nothing on the ethernet that can scramble the signal as all hosts are transmitting zeros.

Compression

As hinted at above the possibilities for compression are fantastic. You can forget Huffman encoding and Lempel-Ziv can take a walk! The compression techniques can reduce any amount of data to 1 number, although that number may be larger than the convenient word size of a given architecture. The basic algorithm is outlined below.

```
while (length(data) > 1) {
    data = count_zeros(data);
    iteration ++;
}
return iteration;
```

This can be also be changed to do essentially the above but in N steps for large files.

Hardware

It is expected that there may be some implementation problems associated with the hardware of this device. However, the benefits appear to outweigh the drawbacks in many ways. To begin with, the memory using this technique should be simple. There is no need to invert bits or to even sense the bits – they should all be zero anyway. Memory failure can be detected very easily, no need for complex CRC checks – any 1 bits are obviously due to failing memory.

Another advantage is that all memory is effectively permanent, as there is no state to be saved. This means computers built using this model should be unaffected by power-outs and be impervious to crashes.

Encryption

This scheme also seems to lend itself to data encryption. The details have not been fully worked out and may appear in a second paper once the decryption algorithms have been straightened out.

Parallelism and Data-flow

Again, this method has more advantages for parallel hardware. Shared memory is particularly easy to implement for the same reasons that the ethernet is easy - effectively there are no changes in memory state so collisions can't happen (unless defective memory is present).

Implementation

There have been some doubts raised about the hardware realisation of this technique, but in general this can probably be attributed either to the resistance to change generally found, or by manufacturers protecting their own interests. The vast benefits that this method seems to have though should mean that once it is taken up it will clean up in the computer industry.

Rules of AUUG Incorporated

NAME

1. The incorporated association shall be known as the **AUUG Incorporated**, abbreviated hereinafter to **AUUG**.

DEFINITIONS

2. (1) In these rules, unless otherwise stated:
“he”, “him” and “his” shall also be construed to mean “she”, “her” and “her” respectively;
“The Act” means the *Associations Incorporation Act 1981 (Vic)*;
“Financial year” means the period from 1 June to 31 May;
“General Committee Member” shall mean a general member of the Management Committee;
“mail” shall imply the transmission of information in written or printed form, first-class pre-paid, via the general post or public or private courier service;
“unfinancial member” shall mean any member whose most recent term of membership has expired and who has not yet paid the subscription for the next twelve month period;
“voting member” shall mean any member entitled to cast a vote.
- (2) In these Rules, a reference to the secretary of the AUUG is a reference:
(d) where a person holds office under these Rules as Secretary of the AUUG, to that person; and
(e) in any other case to the Public Officer of the AUUG.
- (3) Words or expressions in these rules shall be interpreted in accordance with, and subject to, the Act as in force from time to time.
- (4) If any doubt arises as to the proper construction or meaning of any clauses in these Rules, the decision of the Management Committee thereon shall be final and conclusive provided such decision be reduced to writing and recorded in the minutes of a meeting of the Management Committee.

AIMS

3. The aims for which the AUUG is established are to promote knowledge and understanding of Open Systems including but not restricted to the UNIX system, networking, graphics, user interfaces and programming and development environments, and related standards

For the furtherance of these aims and to achieve its purposes, the AUUG may carry out any or all of the following activities: conduct technical meetings, conferences, discussion groups, panels, lectures and other types of meeting; prepare and distribute a newsletter and other publications; collect software and distribute said software to its members for their use; verify licences of members for the purposes of administering the services of the AUUG; subscribe to or cooperate with or affiliate with or amalgamate with other associations formed elsewhere with similar aims; accumulate assets; and establish and promote other activities not included in the above list consistent with its aims for the benefit of its members.

ELIGIBILITY FOR MEMBERSHIP

4. Any individual or organisation who subscribes to the aims of the association, and who agrees to be bound by its rules and regulations and who has not been previously expelled from the association shall be eligible to join the AUUG.
5. An application for membership shall be in writing on the form approved by the Management Committee and shall provide such information as shall from time to time be prescribed by the Management Committee.
6. (1) Membership shall become current on the first day of the month following the date on which a valid membership application accompanied by payment of the appropriate entrance fee plus annual membership subscription is received by the Secretary, and shall continue for twelve months from that date.
(2) Upon completion of the initial membership period and any subsequent periods, membership may be renewed for a further period of twelve months by payment of an additional annual membership subscription.
7. (1) There shall be four classes of members: Ordinary members, Institutional members, Student members and Honorary Life Members.
(2) Any natural person who is eligible to be a member may become an Ordinary Member.
(3) Any person or organisation who is eligible to be a member may become an Institutional Member.
(4) Any full-time student who is eligible to be a member may become a Student Member.
(5) Any person who is an Ordinary Member of at least five years standing and who has rendered special services to the AUUG may be elected via a ballot of the members as an Honorary Life member.
(6) If before the first day of May the Secretary receives a petition from at least twenty voting members requesting the election of a member of the AUUG to the position of Honorary Life Member, then he shall arrange a ballot of the membership on this question to be conducted in conjunction with the annual election of Officers and General Committee Members.
8. All Ordinary, Institutional and Honorary Life Members whose membership is current shall be entitled to cast a vote.

MEMBERSHIP SUBSCRIPTIONS AND FEES

9. The Management Committee shall determine before the commencement of each financial year a scale of fees for entrance to the AUUG, and for annual subscriptions for each class of members to be applied during that financial year.

REGISTER OF MEMBERS

10. (1) The Secretary shall keep and maintain a register of members in which shall be entered the full name and address of each member and the register shall be available for inspection by members at the address of the Public Officer.
(2) Nothing in the previous subsection shall entitle any member to make a copy of the register of members, except with the permission of the Management Committee, and on such terms and conditions as the Management Committee shall from time to time determine.

TERMINATION OF MEMBERSHIP

11. (1) A member may resign his membership at any time by giving notice in writing to the Secretary. No member who resigns shall have any claim for a refund of subscriptions paid.
- (2) A member who has been unfinancial for more than two calendar months shall be deemed to have resigned his membership, and shall no longer be entitled to any privileges enjoyed by members.
- (3) Former members who have resigned will be entitled to rejoin the AUUG on the same basis as new members joining the AUUG.

EXPULSION OF MEMBERS

12. Upon receipt of a petition so requesting from twenty or more members, or half the membership, whichever is less, the Management Committee shall call upon any member to explain any alleged misconduct, and the Management Committee shall have power to suspend or expel any member who in its opinion has either been guilty of misconduct or has acted prejudicially to the interests of the AUUG or who has wilfully infringed any of the Rules of the AUUG.

GENERAL MEETINGS

13. The Annual General Meeting shall be held within the second half of each calendar year. The date and general location of each Annual General Meeting shall be determined at the preceding Annual General Meeting but either the date or location or both may be changed by the Management Committee if it proves impossible or highly inconvenient to meet at the location previously selected or on the date previously selected.
14. An ordinary general meeting of the AUUG shall be called by the Management Committee in conjunction with any technical meeting or conference or other function where attendance by a quarter or more of the voting members is expected by the Management Committee.
15. (1) Written notice of the time and place for each meeting and its agenda shall be mailed to each voting member of the AUUG at least four weeks before the date of the meeting.
- (2) Business conducted at such meetings shall be confined to matters included in the written agenda, reports from Officers, and resolutions instructing the Management Committee to conduct a formal ballot of the membership on matters of substance. Such resolutions shall not be binding on the Management Committee unless the meeting was attended by at least twenty voting members, or half the membership, whichever is less, and the resolution was supported by at least three-quarters of the members voting.
- (3) All voting members shall be entitled to cast one vote.
- (4) Any voting member may award his proxy to another voting member for the period of a single General meeting providing he so notifies the Secretary in writing at least 24 hours before the appointed time of commencement of the meeting.
16. (1) Upon receipt of a petition so requesting from twenty or more members, or half the membership, whichever is less, the Secretary shall call an Extraordinary

General meeting of the AUUG for a date no later than two calendar months after receipt of the petition.

- (2) The business of the meeting shall be confined to matters described in the petition and to other matters specifically provided for in these rules and recorded in the written agenda sent to all members by mail at least four weeks before the date set for the meeting.
 - (3) If the Management Committee does not cause a special general meeting to be held within two months after the date on which the petition is sent to the address of the Secretary, the members presenting the petition or any of them, may convene a special general meeting to be held not later than four months after the date of that petition.
 - (4) A special general meeting convened by members in pursuance of these rules shall be convened in the same manner as nearly as possible as that in which those meetings are convened by the Management Committee and all reasonable expenses incurred in convening the meeting shall be refunded by the AUUG to the persons incurring the expenses.
17. (1) For each general meeting, the quorum shall be fifty members personally present and entitled to vote.
- (2) If within an hour after the appointed time for the commencement of a general meeting, a quorum is not present, the meeting if convened upon the requisition of members shall be dissolved.
 - (3) In other cases the meeting shall be deferred to a place and time determined by the Management Committee. If that meeting is to be at the same location on the following day then notice of the meeting may be given by posting a notice at the location specifying the time of the meeting and the business to be conducted no less than four hours before the time of the meeting. In any other case notice shall be given as for any other General Meeting.
18. At all general meetings of the AUUG the Chair shall be taken by the President, or in his absence, the Vice-President, or in his absence by a member elected by the meeting.

OFFICERS

19. The Officers of the AUUG shall be: the **President**; the **Vice-President**; the **Secretary**; the **Treasurer**; the **Returning Officer**; and the **Assistant Returning Officer**.

MANAGEMENT COMMITTEE

20. The management and control of the business and general affairs of the AUUG shall be vested in a Management Committee of nine members, namely: the President; the Vice-President; the Secretary; the Treasurer; and five General Committee Members.

ELECTIONS

21. (1) The election of Officers and General Committee Members shall be by a postal ballot held annually.
- (2) Nominations for each position shall be received by the Secretary up until the fourteenth day of April each year. Each nomination must be in writing, must name the position or positions sought, must be signed by at least three voting

members, and must be countersigned by the nominated member who must be a financial voting member of the AUUG.

- (3) Where only one valid nomination is received for a particular position by the close of nominations, the nominee shall be declared elected forthwith, and no ballot for that position shall be held.
 - (4) Any position for which no nomination is received, or which remains unfilled after the election has been conducted, shall be considered as a vacancy on the Management Committee, and handled as specified in these rules.
 - (5) On or before the first day of May, the Secretary shall advise the Returning Officer of all valid nominations received, and if a ballot is required shall advise him of a date no later than the fifteenth day of May for the ballot for all contested positions, and shall provide him with a list of voting members.
 - (6) While any Ordinary Member may be nominated to more than one office or position, no person shall be elected to more than one position. Ballots shall be determined in the following order: for President, for Vice-President, for Secretary, for Treasurer, for General Committee Members, for Returning Officer, and lastly for Assistant Returning Officer.
 - (7) All voting members shall be entitled to cast one vote.
22. The term of office for all Officers and General Committee Members shall be for one year, from July 1 to June 30.

VACANCIES ON THE MANAGEMENT COMMITTEE

23. (1) The position of any General Committee Member shall be vacated if the member fails to attend any Management Committee meeting without furnishing a satisfactory explanation as to the cause of his absence, and if the Management Committee resolves that his office be vacated, or if the member ceases to be a member of the AUUG
- (2) Should the office of President be vacant, the Vice-President shall become President, and the office of Vice-President shall become vacant instead. If for this reason, or for any other, at any time any of the other principal Officers (Vice-President, Secretary or Treasurer) be unable to continue in office for any reason, then the Management Committee shall appoint one of their number to the vacant office.
 - (3) Should a vacancy occur among the other Officers, or among the General members of the Management Committee, then the Management Committee shall appoint an Ordinary Member of the AUUG to fill the vacancy.
 - (4) Should a vacancy occur as the result of the creation of a new position, the vacancy shall be filled as specified in these rules.
 - (5) The Management Committee shall make the approval of such appointments an order of business for the next General Meeting of the AUUG if any such meeting will be held before the next election of Officers and General Committee Members.

MANAGEMENT COMMITTEE MEETINGS

24. (1) The Management Committee shall meet formally at least twice per year.
- (2) Notification of time, place and agenda for each meeting shall be made in writing to each member of the Committee by the Secretary at least four weeks in advance.

- (3) All members of the AUUG are entitled to be present at such meetings, and may speak when invited by the Chairman, but only members of the Management Committee may vote.
25. At meetings of the Management Committee the President shall take the chair, or in his absence, the Vice-President, or in his absence a member of the Management Committee elected by the meeting.
26. The quorum for such meeting shall be five. If a quorum is not present at the nominated time for the start of the meeting, the commencement of the meeting may be delayed for up to one hour, and if at that time a quorum is still not present the meeting shall be dissolved.
27. Resolutions of the committee shall require a simple majority of the members present and voting. The chairman shall have a casting vote in the event of a tie.

DISTRIBUTION OF INCOME

28. The income and property of the AUUG however derived shall be applied solely towards the aims and purposes of the AUUG as set out in these Rules, and no portion thereof shall be paid or transferred directly or indirectly by way of dividend to any member of the AUUG at any time.
29. The AUUG shall not appoint a person who is a member of the Management Committee to any office in the gift of the association to the holder of which there is payable any remuneration by way of salary, fees or allowances.
30. Notwithstanding the previous section the AUUG may compensate the reasonable expenses actually incurred by any member in the conduct of the business of the AUUG under the direction of the Management Committee.

CHAPTERS

31. (1) Ten or more members of the AUUG may petition the Management Committee to form a **chapter** of the AUUG.
- (2) General rules for the organisation, operation, obligations and privileges of chapters shall be as resolved by the Management Committee or the membership as a whole from time to time.
- (3) Each chapter shall appoint a chapter committee consisting of at least a Chapter Chairman and a Secretary/Treasurer.
- (4) The chapter committee may convene meetings consistent with the aims of the AUUG, but may not enter into any financial commitments on behalf of or in the name of the AUUG except with the written approval of the Management Committee.

AFFILIATION OR AMALGAMATION WITH OTHER ORGANISATIONS

32. The Management Committee may at any time seek or discuss the possibility of affiliation or amalgamation with any other organisation whose aims are similar to or compatible with those of the AUUG. No agreement for affiliation or amalgamation may be finalised until the matter has received the assent of three-quarters of the members voting in a postal ballot.

DISSOLUTION OF THE AUUG

33. (1) Upon receipt of a petition requesting the dissolution of the AUUG from twenty or more members, or half the membership, whichever is less, the Secretary shall arrange for the question to be put to the membership by ballot no later than one month after the date that he receives the petition.
- (2) If three-quarters of the members voting agree, the AUUG shall be dissolved.
- (3) If upon the dissolution of the AUUG there remains after satisfaction of all its debts and liabilities any property whatsoever, the same shall not be paid to or distributed among the members or Chapters if any, but shall be given or transferred to some public educational institution, or other institution to be determined at or before the time of dissolution by resolution of the membership.

CHANGES TO THE RULES

34. Changes to these Rules may be initiated at the request of a General meeting, or by the Management Committee. All proposed changes must be approved by a three-quarters majority of the votes received in a postal ballot of the members before having effect.

RIGHTS OF MEMBERS

35. (1) Each member shall be entitled to attend all meetings of the AUUG, including meetings of the Management Committee, provided any prescribed attendance fee is paid.
- (2) Each member shall be sent a copy of the association's newsletter.
- (3) Each member entitled to vote in a ballot shall be sent notice in writing of all ballots and copies in writing of the annual reports of the Secretary and Treasurer.

THE SECRETARY

36. (1) The Secretary shall furnish to the Returning Officer a complete list of all voting members whenever this is required for the conduct of a ballot.
- (2) The Secretary shall keep or cause to be kept full and correct minutes of all resolutions and proceedings at General meetings and Management Committee meetings of the AUUG.
- (3) The Secretary shall conduct correspondence on behalf of the AUUG.
- (4) The Secretary shall, during his last month of office, prepare a written report on the state of the affairs of the AUUG for distribution to the membership.

THE TREASURER

37. (1) The Treasurer shall keep or cause to be kept correct accounts and books and records showing the financial affairs of the AUUG.
- (2) The Treasurer shall notify the President and Secretary in writing of the usual location of said accounts, books and records whenever this location is changed.
- (3) The Treasurer shall receive all fees and subscriptions and all other monies on account of the AUUG and provide receipts for the same. The Treasurer shall deposit all monies received into a bank account maintained by the AUUG.
- (4) The Treasurer shall receive accounts for payment for services rendered to the AUUG, and as directed by the Management Committee arrange for payment from the AUUG's account.

- (5) The Treasurer shall, during his last month of office, prepare or cause to be prepared a written report on the financial affairs of the AUUG for distribution to the membership.
- (6) The accounts and books referred to in sub-clause (1) shall be available for inspection by members.

FUNDS

38. The funds of the AUUG shall be derived from entrance fees, annual subscriptions, donations and such other sources as the Management Committee determines.
39. (1) Signing Officers for the AUUG's accounts shall be the President, the Vice-President, the Secretary, the Treasurer and one other General Committee Member chosen by the Management Committee.
 - (2) All cheques, drafts, and other orders for payment of money out of the funds of the AUUG, if for less than a limit established by the Management Committee, may be signed by only one Signing Officer.
 - (3) For other amounts, each such instrument must be signed by at least two Signing Officers.

SEAL

40. (1) The Common Seal of the AUUG shall be kept in the custody of the Secretary.
 - (2) The Common Seal shall not be affixed to any instrument except by authority of the Management Committee and the affixing of the Common Seal shall be attested by the signatures either of two members of the Management Committee or of one member of the Management Committee and the Public Officer of the AUUG.

EXECUTION OF CONTRACTS

41. The Management Committee, except as otherwise provided in these Rules, may prospectively or retroactively authorise any Officer or member of the AUUG to enter into any contract or execute and satisfy any instrument, and any such authority may be general or confined to specific instances, except that any contract whose dollar value exceeds an amount predetermined by the Management Committee must be specifically authorised in advance by the Management Committee.

VOTING

42. (1) All voting by the members with respect to the election of Officers and General Committee Members, with respect to the election of Honorary Life Members, with respect to changes to these Rules, and all other substantive matters shall be conducted by postal ballot.
 - (2) Every voting member of record as of the date of entry of a ballot into the mails shall be entitled to vote in the ballot.
 - (3) On all questions to be put to a ballot, the Secretary shall designate a date for the ballot to be placed in the mails, and the due date shall be four weeks after that date.
 - (4) The Returning Officer shall nominate the address to which voters shall return completed ballot papers by mail.

- (5) A ballot will not be counted if it is received after the due date or if the ballot paper does not comply with the instructions printed on it.
- (6) The ballots will be received by the Returning Officer, and counted by him and the Assistant Returning Officer.
- (7) The Returning Officer shall report the result of the ballot in writing to the Secretary no later than two weeks after the due date.
- (8) The formal procedures of voting shall be determined from time to time by the Management Committee.

AUUGN Back Issues

Here are the details of back issues of which we still hold copies. All prices are in Australian dollars and include surface mail within Australia. For overseas surface mail add \$2 per copy and for overseas airmail add \$10 per copy.

pre 1984	Vol 1-4	various	\$10 per copy
1984	Vol 5	Nos. 2, 3, 5, 6 Nos. 1,4	\$10 per copy unavailable
1985	Vol 6	Nos. 2, 3, 4, 5, 6 No. 1	\$10 per copy unavailable
1986	Vol 7	Nos. 1, 4-5, 6 Nos. 2-3 (Note 2-3 and 4-5 are combined issues)	\$10 per copy unavailable
1987	Vol 8	Nos. 1-4 Nos. 5, 6	unavailable \$10 per copy
1988	Vol 9	Nos. 1, 2, 3 Nos. 4, 5, 6	\$10 per copy \$15 per copy
1989	Vol 10	Nos. 1-6	\$15 per copy
1990	Vol 11	Nos. 1-4	\$15 per copy

Please note that we do not accept purchase orders for back issues except from Institutional members. Orders enclosing payment in Australian dollars should be sent to:

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

SESSPOOLE

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

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

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

To promote knowledge and understanding of Open Systems; and to promote knowledge and understanding of Open Bottles.

(Note that these aims have been updated in line with recent changes to the aims of AUUG Inc.)

SESSPOOLE was the first Chapter of AUUG Inc to be formed, and members of SESSPOOLE were involved in the staging of the AUUG Summer'90 and Summer'91 meetings.

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

Wednesday, 27th February, 1991

Thursday, 18th April, 1991

Tuesday, 28th May, 1991

Wednesday, 17th July, 1991

Thursday, 29th August, 1991

Hope we'll see you there!

For more information on SESSPOOLE and SESSPOOLE activities (including a description of how much fun it is to book a table in a restaurant under the name "SESSPOOLE"), contact either David Purdue (ph. (03) 353 3913, e-mail: auugn@munnari.oz.au) or John Carey (ph. (03) 587 1444, e-mail: john@labtam.labtam.oz.au), or keep a lookout for announcements in **aus.auug**.

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 if you are currently really an AUUG member, examine the mailing label of this AUUGN. In the lower right corner you will find information about your current membership status. The first letter is your membership type code, N for regular members, S for students, and I for institutions. Then follows your membership expiration date, in the format exp=MM/YY. The remaining information is for internal use.

Check that your membership isn't about to expire (or worse, hasn't expired already). Ask your colleagues if they received this issue of AUUGN, tell them that if not, it probably means that their membership has lapsed, or perhaps, they were never a member at all! Feel free to copy the membership forms, give one to everyone that you know.

If you want to join AUUG, or renew your membership, you will find forms in this issue of AUUGN. Send the appropriate form (with remittance) to the address indicated on it, and your membership will (re-)commence.

As a service to members, AUUG has arranged to accept payments via credit card. You can use your Bankcard (within Australia only), or your Visa or Mastercard by simply completing the authorisation on the application form.

AUUG Incorporated

Application for Institutional Membership

Australian UNIX* systems Users' Group.

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

To apply for institutional membership of the AUUG, complete this form, and return it with payment in Australian Dollars, or credit card authorisation, to:

AUUG Membership Secretary
 PO Box 366
 Kensington NSW 2033
 Australia

• Foreign applicants please send a bank draft drawn on an Australian bank, or credit card authorisation, and remember to select either surface or air mail.

This form is valid only until 31st May, 1991

..... does hereby apply for

- New/Renewal* Institutional Membership of AUUG \$325.00
- International Surface Mail \$ 40.00
- International Air Mail \$120.00

Total remitted AUD\$ _____
 (cheque, money order, credit card)

* Delete one.

I/We agree that this membership will be subject to the rules and by-laws of the AUUG as in force from time to time, and that this membership will run for 12 consecutive months commencing on the first day of the month following that during which this application is processed.

I/We understand that I/we will receive two copies of the AUUG newsletter, and may send two representatives to AUUG sponsored events at member rates, though I/we will have only one vote in AUUG elections, and other ballots as required.

Date: ___ / ___ / ___ Signed: _____

Title: _____

Tick this box if you wish your name & address withheld from mailing lists made available to vendors.

For our mailing database - please type or print clearly:

Administrative contact, and formal representative:

Name: Phone: (bh)

Address: (ah)

..... Net Address:

..... Write "Unchanged" if details have not altered and this is a renewal.

Please charge \$_____ to my/our Bankcard Visa Mastercard.

Account number: _____ . Expiry date: ___ / ___ .

Name on card: _____ Signed: _____

Office use only: Please complete the other side.

Chq: bank _____ bsb _____ - a/c _____ # _____

Date: ___ / ___ / ___ \$ _____ CC type ___ V# _____

Who: _____ Member# _____

Please send newsletters to the following addresses:

Name: Phone: (bh)
Address: (ah)
.....
..... Net Address:
.....
.....

Name: Phone: (bh)
Address: (ah)
.....
..... Net Address:
.....
.....

Write "unchanged" if this is a renewal, and details are not to be altered.

Please indicate which Unix licences you hold, and include copies of the title and signature pages of each, if these have not been sent previously.

Note: Recent licences usually revoke earlier ones, please indicate only licences which are current, and indicate any which have been revoked since your last membership form was submitted.

Note: Most binary licensees will have a System III or System V (of one variant or another) binary licence, even if the system supplied by your vendor is based upon V7 or 4BSD. There is no such thing as a BSD binary licence, and V7 binary licences were very rare, and expensive.

- System V.3 source
- System V.2 source
- System V source
- System III source
- 4.2 or 4.3 BSD source
- 4.1 BSD source
- V7 source
- Other (*Indicate which*)
- System V.3 binary
- System V.2 binary
- System V binary
- System III binary

AUUG Incorporated

Application for Ordinary, or Student, Membership Australian UNIX* systems Users' Group.

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

To apply for membership of the AUUG, complete this form, and return it with payment in Australian Dollars, or credit card authorisation, to:

AUUG Membership Secretary
P O Box 366
Kensington NSW 2033
Australia

- Please don't send purchase orders — perhaps your purchasing department will consider this form to be an invoice.
- Foreign applicants please send a bank draft drawn on an Australian bank, or credit card authorisation, and remember to select either surface or air mail.

This form is valid only until 31st May, 1991

I, do hereby apply for

- Renewal/New* Membership of the AUUG \$78.00
- Renewal/New* Student Membership \$45.00 (note certification on other side)
- International Surface Mail \$20.00
- International Air Mail \$60.00 (note local zone rate available)

Total remitted AUD\$ _____
(cheque, money order, credit card)

* Delete one.

I agree that this membership will be subject to the rules and by-laws of the AUUG as in force from time to time, and that this membership will run for 12 consecutive months commencing on the first day of the month following that during which this application is processed.

Date: ___ / ___ / ___ Signed: _____

Tick this box if you wish your name & address withheld from mailing lists made available to vendors.

For our mailing database - please type or print clearly:

Name: Phone: (bh)
 Address: (ah)

 Net Address:

 Write "Unchanged" if details have not
 altered and this is a renewal.

Please charge \$_____ to my Bankcard Visa Mastercard.
 Account number: _____ . Expiry date: ___ / ___ .
 Name on card: _____ Signed: _____

Office use only:
 Chq: bank _____ bsb _____ - a/c _____ # _____
 Date: ___ / ___ / ___ \$ CC type ___ V# _____
 Who: _____ Member# _____

Student Member Certification *(to be completed by a member of the academic staff)*

I, certify that
..... *(name)*
is a full time student at *(institution)*
and is expected to graduate approximately / / .

Title: _____

Signature: _____

AUUG Incorporated

Application for Newsletter Subscription

Australian UNIX* systems Users' Group.

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

Non members who wish to apply for a subscription to the Australian UNIX systems User Group Newsletter, or members who desire additional subscriptions, should complete this form and return it to:

AUUG Membership Secretary
 P O 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, 1991

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 Australian UNIX* systems Users' Group.

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

If you have changed your mailing address, please complete this form, and return it to:

AUUG Membership Secretary
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: Phone: (bh)

Address: (ah)

..... Net Address:

.....

.....

.....

New address (leave unaltered details blank)

Name: Phone: (bh)

Address: (ah)

..... Net Address:

.....

.....

.....

Office use only:

Date: ___/___/___

Who: _____

Memb# _____