

AUUGN

AUUG Inc. Newsletter

Volume 15, Number 1

February 1994



The AUUG Incorporated Newsletter

Volume 15 Number 1

February 1994

CONTENTS

AUUG General Information	3	
Editorial	5	
AUUG Institutional Members	6	
AUUG President's Page	9	
Liz is leaving	<i>Janet Jackson</i>	9
AUUG - Call for Participation	<i>Peter Wishart</i>	10
AUUG Inc 1994 Annual Elections - Call for Nominations	11	
AUUG Inc 1994 Annual Elections - Nomination Form	12	
The Answer is	<i>Liz Framann</i>	13
AUUG Updates Agreements	15	
AUUG & Strategic Publishing Group Reach Agreement	16	
Conference Announcements		
AUUG 94 Call for Papers	17	
Berkeley 4.3 - 4.4BSD Workshops	23	
AUUG Summer Conference Series	27	
UPDATE: UniForum NZ'94	29	
KIWI Invitation to AUUG Members	31	
Open System Publications	32	
AUUG Local Chapters		
AUUG Inc. - Victorian Chapter	33	
Update on AUUG Inc. - Victorian Chapter Activities	<i>Stephen Prince</i>	34
From the Western Front	<i>Janet Jackson</i>	35
WAUG - Meeting Reviews	<i>Janet Jackson</i>	36
Report on '94 Winter Usenix	<i>Greg Rose</i>	39
AARNet Mail Affiliation	42	
Book Reviews	46	
Prentice Hall Book Order Form	52	
WoodsLane Information	53	
!AUUGN - from AUUGN Volume 4, Number 3		
Perkin-Elmer Supports Unix	54	
Technical Reports	56	

Sam's Serial Listening Device (Mark 3)	<i>Sam Pascoe</i>	56
A Workbench for a Tools Based Mail Handler	<i>Neil Brown</i>	58
Stashbase	<i>Scott Merrilees</i>	65
From login: - Volume 18, Number 5		
Security Tool Review: TCP Wrappers		68
SAGE Views		70
Perl 5.0 Overview		73
An Update on UNIX-Related Standards Activities		76
Unix Tricks & Traps	<i>Janet Jackson</i>	78
Summary of Management Committee Minutes - 10th December 1993		80
AUUG Membership Categories		82
AUUG Forms		83
AUUGN Volume 14 Index	<i>Stephen Prince</i>	87
Calendar of Events		94

ADVERTISEMENTS

X/Open		22
ZIRCON Systems		32
Advanced User Systems Pty Ltd		35
COGNOS Pty Ltd		37
Prentice Hall Book Order Form		52
WoodsLane Information		53
UniForum NZ'94		64
Softway		68

Copyright © 1994 AUUG Incorporated. All rights reserved.

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

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 permission of AUUG Incorporated.

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

AUUG General Information

Memberships and Subscriptions

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

Membership and General Correspondence

All correspondence for the AUUG should be addressed to:-

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

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

AUUG Business Manager

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

Phone: TBA
Fax: TBA
Email: TBA

AUUG Executive

President	Phil McCrea <i>pmc@atom.ansto.gov.au</i> ANSAMS Private Mail Bag 1 Menai NSW 2234	Vice-President	Glenn Huxtable <i>glenn@cs.uwa.edu.au</i> University of Western Australia Computer Science Department Nedlands WA 6009
Secretary	Peter Wishart <i>peter.wishart@csis.dit.csiro.au</i> CSIRO Div. of Information Technology GPO Box 664 Canberra ACT 2601	Treasurer	Frank Crawford <i>frank@atom.ansto.gov.au</i> ANSAMS Private Mail Bag 1 Menai NSW 2234
Committee Members	Greg Birnie <i>greg@lna.oz.au</i> Leeds & Northrup Australia P/L 42 McKechnie Dr. Brisbane Tech. Park Eight Mile Plains QLD 4113		Stephen Boucher <i>stephen@mtiame.mtia.oz.au</i> MTIA 509 St. Kilda Rd. Melbourne VIC 3004
	Chris Maltby <i>chris@softway.sw.oz.au</i> Softway Pty. Ltd. P.O. Box 305 Strawberry Hills NSW 2021		Michael Paddon <i>mwp@iconix.oz.au</i> Iconix Pty Ltd 851 Dandenong Rd East Malvern VIC 3145
	Rick Stevenson <i>rick@stallion.oz.au</i> Stallion Technologies Pty. Ltd. 56 Sylvan Rd. Toowong, QLD 4066		

AUUG General Information

Next AUUG Meeting

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

Advertising

Advertisements to be included in AUUGN are welcome. They should conform to the standards of other contributions (see page 5). Advertising rates are \$120 for a quarter page, \$180 for half a page, \$300 for the first A4 page, \$250 for a second page, \$500 for the inside cover and \$750 for the back cover. There is a 20% discount for bulk ordering (ie, when you pay for three issues or more in advance). Contact the business manager for details.

Mailing Lists

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

Back Issues

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

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

Conference Proceedings

A limited number of the Conference Proceedings for AUUG'92 and AUUG'93 are still available, at \$50 for members and \$60 for non-members. Contact the AUUG secretariat.

Acknowledgement

This newsletter was produced with the kind assistance of and on equipment provided by the Australian Nuclear Science and Technology Organisation. A copy of FrameMaker for use in the production of the newsletter has been provided by Platform Technologies.

Disclaimer

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

AUUG Newsletter

Editorial

Welcome to AUUGN Volume 15 Number 1, and welcome to a new year. This looks to be a busy year, at least it has started that way for me.

By the time this edition reaches you, we will be in the middle of the Summer Conference Series. Hopefully you either attended or are planning to attend one of the conferences, and it proves worthwhile. A lot of people have put effort into the organisation this year. The highlight of this years conference series is the *travelling roadshow* featuring Kirk McKusick, and while this may not occur every year, the AUUG management committee intend to try it again in the future.

As you will read elsewhere in this issue, we also have some sad news, Liz Fraumann is leaving us, to return to the USA. We have employed Catrina Dwyer as our new Business Manager. There are a number of tributes to Liz included in this issue. Farewell and good luck Liz, welcome Catrina.

Also in this issue are a number of other informative items, such as a review of a major new Unix book being released by Prentice-Hall, important information about the upcoming AUUG94 and Uniforum NZ conferences, and a number of interesting papers. There are also a number of AUUG administrative matters, in particular nominations for the management committee elections. Remember, this is your organisation, so get behind it.

Last, but definitely not least, we have an index of all the articles published in AUUGN Vol 14. This is thanks to Stephen Prince, who is indexing all the previous volumes of AUUGN. We hope to publish these some time this year, and we will continue to publish indexes in future volumes. Thanks Stephen.

Jagoda Crawford

AUUGN Correspondence

All correspondence regarding the AUUGN should be addressed to:-

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

AUUGN Book Reviews

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

Contributions

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

Volume 15 No 2	Friday 25th March
Volume 15 No 3	Friday 27th May
Volume 15 No 4	Friday 29th July
Volume 15 No 5	Friday 23rd September
Volume 15 No 6	Friday 25th November

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

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

Hardcopy submissions should be on A4 with 30 mm margins, and 30 mm left at the bottom so that the AUUGN footers can be pasted on to the page. Small page numbers printed in the footer area would help.

AUUG Institutional Members as at 01/02/1994

A. Goninan & Co. Limited	Cognos Pty. Ltd.
A.N.U.	Colonial Mutual
AAII	Com Net Solutions
Aberfoyle Resource Limited	Com Tech Communications
ACAY Network Computing Pty.Ltd.	Commercial Dynamics
ACAY Network Computing Pty.Ltd.	Communica Software Consultants
Actrol Parts	Composite Buyers Ltd.
Adept Software	Computechnics Pty. Ltd.
Advanced Software Engineering	Computer De Tokyo Corporation
Alcatel Australia	Computer Law Corporation
Amalgamated Television Services	Computer Sciences of Australia Pty. Ltd.
Amdahl Pacific Services	Computer Software Packages
Andersen Consulting	Computer Systems (Australia) Pty. Ltd.
ANI Manufacturing Group	Comsys International Pty. Ltd.
Animal Logic Research Pty. Ltd.	Copper Refineries Pty. Ltd.
ANSTO	Corinthian Engineering Pty. Ltd.
Anti-Cancer Council of Victoria	Corporate Systems Planning
Atlas Computer Systems	Corporate Workgroup Resources
Attorney Generals' Dept.	CSIRO Division of Information Technology
Attorney-General's Dept.	CSIRO Division of Manufacturing Technology
AUSOM Inc.	CSIRO Division of Wool Technology
Australian Archives	Curtin University of Technology
Australian Bureau of Statistics	Customised Software Solutions Centre
Australian Computing & Communications Institute	Cyberdyne Systems Corporation Pty. Ltd.
Australian Defence Industries Ltd.	Cyberscience Corporation Pty. Ltd.
Australian Electoral Commission	Cybersource Pty. Ltd.
Australian Film Television and Radio School	Data General Australia
Australian Information Processing Centre Pty. Ltd.	Datacraft Technologies
Australian Museum	Dawn Technologies
Australian National Audit Office	Deakin University
Australian Software Innovations	Defence Housing Authority
Australian Submarine Corporation	Defence Service Homes
Australian Taxation Office	Dep. of Environment & Natural Resources
Australian Technology Resources (ACT) Pty. Ltd.	Dept. of Agricultural & Rural Affairs
Australian Technology Resources (WA) Pty. Ltd.	Dept. of Business & Employment
Australian Tourist Commission	Dept. of Defence
Australian Wool Research & Promotion Organisation	Dept. of Education, QLD
AWA Defence Industries	Dept. of Family Services & Aboriginal & Islander Affairs
B & D Australia	Dept. of Industrial Relations, Employment, Training & Further Education
Bain & Company	Dept. of Justice
BHA Computer Pty. Limited	Dept. of Planning & Development
BHP Information Technology	Dept. of State Services
BHP Minerals Exploration Department	Dept. of the Premier & Cabinet
BHP Petroleum	Dept. of the Treasury
BHP Research - Melbourne Laboratories	Dept. of Transport
BHP Research - Newcastle Laboratories	DEVETIR
Bond University	Digital Equipment Corp. (Australia) Pty. Ltd.
Burdett, Buckridge & Young Ltd.	DSTO, Lab 73
Bureau of Meteorology	EASAMS (Australia) Limited
Bytecraft Pty. Ltd.	Electricity Trust of South Australia
C.I.S.R.A.	Electronic Financial Services Limited
Cadcom Solutions Pty. Ltd.	Engineering Computer Services Pty. Ltd.
Cape Grim B.A.P.S	Equinet Pty. Ltd.
Capricorn Coal Management Pty. Ltd.	Equity Systems Pty. Limited
CelsiusTech Australia	Ericsson Australia Pty. Ltd.
Chief Secretary's Dept.	ERIN Unit, Australian National Parks & Wildlife Service
CITEC	ESRI Australia Pty. Ltd.
Classified Computers Pty. Ltd.	FGH Decision Support Systems Pty. Ltd.
Clegg Driscoll Consultants Pty. Ltd.	Financial Network Services
Co-Cam Computer Group	
Coal & Allied Operations	

AUUG Institutional Members as at 01/02/1994

Fire Fighting Enterprises
First State Computing
Flinders University
Fremantle Port Authority
Fujitsu Australia Ltd.
G.James Australia Pty. Ltd.
GEC Alsthom Information Technology
GEC Marconi Systems Ltd.
Geelong & District Water Board
Genasys II Pty. Ltd.
General Automation Pty. Ltd.
GIO Australia
Golden Circle Australia
Great Barrier Reef Marine Park Authority
Gribbles Pathology
Gunnedah Abattoir
Haltek Pty. Ltd.
Hammersley Iron
Heath Insurance
Hermes Precisa Australia Pty. Ltd.
Honeywell Ltd.
Honeywell Ltd.
Hong Kong Jockey Club Systems
(Australia) Pty. Ltd.
I.P.S Radio & Space Services
IBA Healthcare Pty. Ltd.
IBM Australia Ltd.
Iconix Pty. Ltd.
Ideas International Pty. Ltd.
Information Technology Consultants
Informed Technology
Insession Pty. Ltd.
Insurance & Superannuation Commission
Integration Design Pty. Ltd.
International Imaging Systems
Internode Systems Pty. Ltd.
ISR Group Ltd.
ISR Group Ltd.
ISR Group Ltd.
James Cook University of North Queensland
JTEC Pty. Ltd.
Knowledge Engineering Pty. Ltd.
Laboratory Systems Pty. Ltd.
Labtam Australia Pty. Ltd.
Land Information Centre
Land Titles Office
Leeds & Northrup Australia Pty. Limited
Legent Australia Pty. Ltd.
Logica Pty. Ltd.
Lotus Development
Lyons Computer Pty. Ltd.
Macquarie University
Matcom Technologies
Mayne Nickless Courier Systems
Mayne Nickless Information Technology Services
Medical Benefits Funds of Australia Ltd.
Memtec Limited
Mentor Technologies Pty. Ltd.
Mercedes-Benz (Australia)
Metal Trades Industry Association
Mincom Pty. Ltd.
Minenco Pty. Ltd.

Mitsubishi Motors Australia Ltd.
Mitsui Computer Limited
Moldflow Pty. Ltd.
Motorola Computer Systems
MPA International Pty. Ltd.
Multibase Pty. Ltd.
Multiline BBS
Multiuser Solutions Pty. Ltd.
National Library of Australia
NCR Australia
NEC Australia Pty. Ltd.
Northern Territory Library Service
Northern Territory University
Novell
Novell Pty. Ltd.
NSW Agriculture
Object Design Pty. Ltd.
Object Oriented Pty. Ltd.
Object Technology International Pty. Ltd.
Objectif Pty. Ltd.
Ochre Development
Office of Fair Trading
Office of National Assessments
Office of the Director of Public Prosecutions
Olivetti Australia Pty. Ltd.
Open Software Associates Ltd.
OPSM
OSIX Pty. Ltd.
OzWare Developments Pty. Ltd.
Pacific Semiconductor Pty. Ltd.
Pacific Star Communications
Paxus
Petrosys Pty. Ltd.
Philips PTS
Platform Technologies Pty. Ltd.
Port of Melbourne Authority
Powerhouse Museum
PPIT Pty. Ltd.
PPIT Pty. Ltd.
Process Software Solutions Pty. Ltd.
Prospect Electricity
pTizan Computer Services Pty. Ltd.
Public Works Department, Information Services
Pulse Club Computers Pty. Ltd.
Pyramid Technology Corporation Pty. Ltd.
Qantek
Qantek
QLD Electricity Commission
Quality Bakers Pty. Ltd.
Quality By Design Pty. Ltd.
Redland Shire Council
Rehabilitation Tasmania
Release4
Renison Golfields Consolidated Ltd.
Repatriation General Hospital, Hollywood
RGC Minerals Sands, Divisional Office
Rinbina Pty. Ltd.
Royal Melbourne Institute of Technology
Scitec Communication Systems
Sculptor 4GL+SQL
SEQEB Business Systems
Shire of Eltham

AUUG Institutional Members as at 01/02/1994

Siemens Nixdorf Information Systems Pty. Ltd.	University of Technology, Sydney
Smargon ARC	Unipac Pty. Ltd.
Snowy Mountains Authority	Vanoco Pty. Ltd.
SoftGen Pacific Pty. Ltd.	Vicomp
Software Development International Pty. Ltd.	Victoria University of Technology
Softway Pty. Ltd.	VME Systems Pty. Ltd.
Sony Technology Centre of Australia	Walter & Eliza Hall Institute
South Australian Co-operative Bulk Handling	Wang Australia Pty. Ltd.
St. Gregory's Armenian School	Water Board
St. Vincent's Private Hospital	Western Mining Corporation
Stallion Technologies Pty. Ltd.	Woodside Offshore Petroleum
Standards Australia	Work Health Authority
State Bank of NSW	Workstations Plus
State Super (SSIMC)	XEDOC Software Development Pty. Ltd.
Steelmark Eagle & Globe	Zircon Systems Pty. Ltd.
Sterling Software	Zurich Australian Insurance
Storage Technology of Australia	
Strategic Information Technologies Pty. Ltd.	
Sunburst Regency Foods	
Swinburne Institute of Technology	
Sydney Electricity	
Sydney Ports Authority	
System Builder Development Pty. Ltd.	
Systems Development Telecom Australia	
TAB of Queensland	
TAFE NSW, Information Systems Division	
Tandem Computers	
Tattersall Sweep Consultation	
Technical Software Services	
TechNIX Consulting Group International	
Telecom Australia	
Telecom Australia Corporate Customer	
Telecom Network Engineering Computer Support Services	
Telecom Payphone Services	
The Far North QLD Electricity Board	
The Fulcrum Consulting Group	
The Preston Group	
The Roads & Traffic Authority	
The Southport School	
The University of Western Australia	
Thomas Cook Ltd.	
TNT Australia Information Technology	
Toshiba International Corporation Pty. Ltd.	
Tower Software Engineering Pty. Ltd.	
Tower Technology Pty. Ltd.	
Tradelink Plumbing Supplies Centres	
Turbosoft Pty. Ltd.	
TUSC Computer Systems Pty. Ltd.	
UCCQ	
Unidata Australia	
Uninet Consulting Pty. Ltd.	
Unisys Australia Ltd.	
University of Adelaide	
University of Melbourne	
University of New England, Dept. of Maths, Stats & Computer Science	
University of New South Wales	
University of Queensland	
University of South Australia	
University of Sydney	
University of Tasmania	

AUUG President's Page

Farewell to Liz

It's with some sadness that we farewell Liz Fraumann, who has been the AUUG Business manager for the last couple of years. Liz has decided to return to the US for personal reasons, and we will certainly miss her. Liz's experience at UNIX International in the US before she came to Australia provided her with a fine background for her work here at AUUG, and she brought with her a wealth of ideas, which will remain her legacy. Many of you will have seen Janet Jackson's 'ode to Liz', which is published elsewhere in this issue.

... but welcome Catrina

I do have some good news to report, however: the Management Committee has appointed Catrina Dwyer as Business Manager to replace Liz. Catrina has excellent credentials for the job, having been an account manager, until recently, for USL in Europe. Although based in London, Catrina's fluent French gave her the credentials to manage USL's French based accounts.

We will provide more of a profile on Catrina in the next issue.

Summer Conferences

As I write, the first of the summer conferences have begun, and I have heard good reports. The Kirk McKusick tour is working out well, and I must thank Glenn Huxtable for seeding the idea and working things out with Kirk. It's provided a focus for the summer conferences, which have for the first time been coordinated, by virtue of the need to schedule Kirk's visits. Again, we must thank Liz F for this coordination role. We also need to send a strong vote of thanks to the organisers from the State Chapters.

Phil McCrea

The following, written by Janet Jackson, has been reprinted from aus.org.auug.

Copyright (c) 1994 Janet Jackson

=====

Liz is leaving

=====

Liz is leaving.
I guess we'll never know
just how much growth we owe
to Liz's honest smile,
sincerity and style.
Conferences have loomed --
and local chapters bloomed --
all easier to do
with Liz to pull us through.
Now Liz is leaving.
I feel a bit of a mug
but Liz is a walking hug
and part of the soul of AUUG
and Liz is leaving.
Liz, whom I rarely met
but whose face I won't forget
and who'd better stay on the Net.

Janet Jackson
<jackson@cwr.uwa.edu.au>
Systems Manager
Centre for Water Research
The University of Western Australia

CALL FOR PARTICIPATION

Throughout its history AUUG Inc. has relied heavily on the generous support of members who voluntarily perform a range of services from President, to AUUGN Editor and more recently as chapter officers and committee. Without the enthusiasm, commitment and effort provided by these volunteers, AUUG would not be the successful organisation that it is today. However we always need more members to help in making AUUG an even better organisation.

Participation through chapter committee and events is a good way to get activities happening in your area. Your local chapter committee would be happy to take your suggestions and offers of assistance. We always need fresh new faces to complement the seasoned volunteers. If you have ideas and a little time to spare, perhaps you could consider participation in your local chapter or the national management committee.

Each year AUUG members elect a national management committee consisting of the officers, President, Vice-President, Secretary, Treasurer and five general committee members. This committee is charged with running AUUG Inc. The Returning Officer and Assistance Returning Officer (who run the elections) are also elected each year.

In last years election for the national management committee we had insufficient nominations to fill all positions on the management committee. This meant that the management committee was forced to co-opt members onto the committee to ensure a full complement. This year we hope to fill all positions from the election process.

The management committee holds office from 1st July through to 30th June of the next year. They meet formally for about one day once every two months, in a place convenient to most committee members (usually in Sydney). Reasonable travel costs for the meeting are reimbursed. There is frequent use of e-mail to communicate issues and discuss ideas. While e-mail access is not required it does help with discussions between meetings. If you do not have e-mail access, do not be deterred from nominating for the committee, I am sure we could work out some effective communication (perhaps by FAX or a guest e-mail account somewhere). As well as attending meetings, committee members should be prepared to take on occasional special project activities (e.g. technical interface between AUUG and AARNET, membership drives, interface with local chapters).

Some AUUG officers have obligations imposed by the constitution (e.g. Treasurer to handle finance and accounts). The President has a special role as principal ambassador for AUUG and nominees for this position must be prepared to devote a fair amount of time to being the public face of AUUG Inc. The management committee is supported in its activities by the AUUG Secretariat and a Business Manager.

Nominations for the election this year close on the 14th of April 1994. A call for nominations and nomination form are included in this issue of AUUGN. Participation in the election process, either as a nominee, or as an elector, is your way of influencing the directions of AUUG. Please exercise your rights to ensure that the management committee remains representative of the interests of AUUG members. Please do not be deterred if you do not have ready access to the required AUUG members to sign your nomination form. Just contact your local chapter committee or contact me directly (contact details below) and we will organise the necessary signatures for your nomination form.

We look forward to 1994/95 being another year of success for AUUG and with your help we can make it even better than 1993/94.

If you would like any more information on the roles of AUUG officers and committee members or have any comments or feedback then please do not hesitate to contact anyone on the current management committee (contact details in the front of AUUGN).

Peter Wishart
Secretary - AUUG Inc.

Phone: (06) 275 0908 (W) FAX: (06) 257 6325
(06) 247 2996 (H) E-mail: peter.wishart@csis.dit.csiro.au.

AUUG Inc. 1994 Annual Elections Call for Nominations

Nominations are invited for the following positions in AUUG Inc.:

President
Vice-President
Secretary
Treasurer
Ordinary Committee Member (5 positions)
Returning Officer
Assistant Returning Officer

Nominations must be made in writing and must be signed by the nominee and three (3) financial members of AUUG Inc and indicate which position(s) are sought. A sample nomination form is attached. Nominees must be financial members of AUUG Inc and may nominate for any or all of the above positions. While any member may be nominated to more than one position, no person can be elected to more than one position. Election to positions is determined in the order shown above.

Nominees should include a policy statement of up to 200 words with the nomination. This word count shall not include sections of the statement stating in point form name, personal details and positions held on, or by appointment of, the AUUG Management Committee and chapters.

Policy statements exceeding the word limit shall be truncated at the word limit when included in the ballot information.

Nominations must be received by the Secretary by 14th April 1994. They may be lodged by one of the following methods:

(1) by post to:

The Secretary
AUUG Inc.
PO Box 366
Kensington NSW 2033

(must be received no later than 2 business days after April 14th and be postmarked no later than 12 midday on April 14th 1994)

(2) by hand to:

The Secretary (Peter Wishart)
OR the AUUG Inc Secretariat
no later than 5 pm on April 14th 1994.

(3) by FAX to:

The Secretary (FAX: (06) 257 6325)
OR the AUUG Inc. Secretariat (FAX: (02) 332 4066)
no later than 5 pm on April 14th 1994.

AUUG Inc.
1994 Annual Elections
Nomination Form

We,

(1) Name: _____ AUUG Member #: _____ and

(2) Name: _____ AUUG Member #: _____ and

(3) Name: _____ AUUG Member #: _____
being current financial members of AUUG Inc do hereby nominate
_____ for the following
position(s):¹

President

Vice-President

Secretary

Treasurer

Ordinary Committee Member (5 positions to be filled)

Returning Officer

Assistant Returning Officer

Signed (1) _____ Date _____

Signed (2) _____ Date _____

Signed (3) _____ Date _____

I, Name: _____ AUUG Member #: _____
do hereby consent to my nomination to the above position(s). and declare that I
am currently a financial member of AUUG Inc.

Signed _____ Date _____

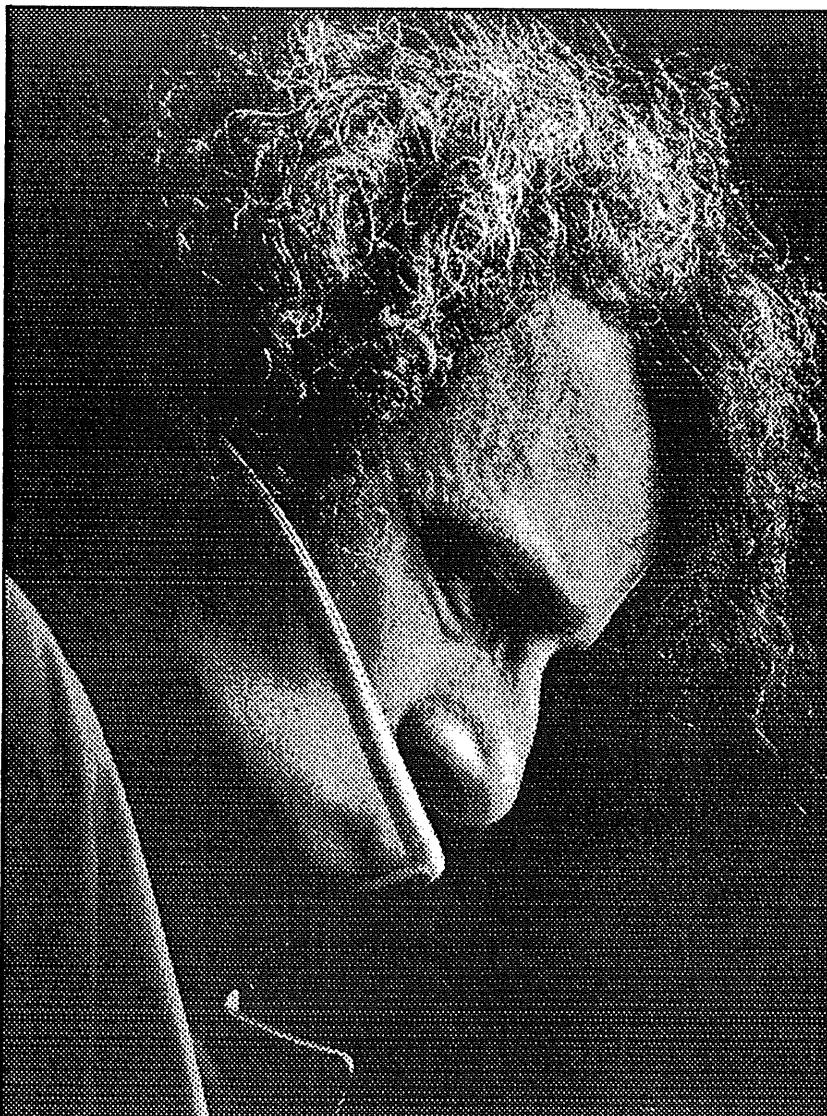
¹ Strike out positions for which nomination is not desired. Each person may be elected to at most one position, and the ballot for positions shall be determined in the order shown on this nomination form.

The Answer is...

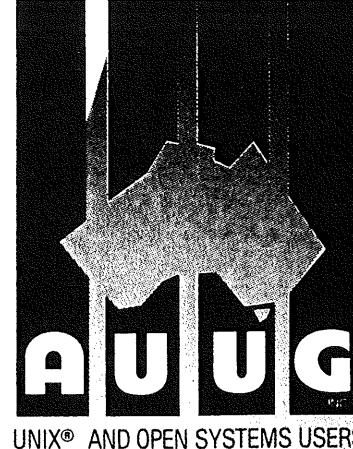
by Liz Fraumann

It was a blustery San Francisco afternoon when I first met Cliff Stoll, in March of '93. We were to meet in the lobby of the Marriott hotel. In great anticipation, I wondered how would I recognise this one man out of the hoards of people here for the UniForum Conference. His one clue, "I'll be wearing a red turtleneck."

At 12:30pm Roger and I strolled into the lobby and sure enough it was packed with "suits." We saw friends and acquaintances from years gone by but no red turtleneck. Each minute seemed to go slower than the previous. Soon, 30 had passed and in the front door rolls a bicycle under the direction of a man in a red turtleneck. He was right, I couldn't miss him. Enter Cliff Stoll. The usual elite character of the Bell Captain refused to secure Mr. Stoll's bicycle for our luncheon meeting, so Cliff in his carefree nature said, "I'll be back." (He did not have Arnold's accent though). He made a mad dash down the street, purchased a new lock, and secured his trusty bike. We finally were seated to a table where we could talk about his trip to Australia for AUUG '93 and get to know each other better. One question which both Roger and I were keen to solve was Morris' palindrome posed to Cliff during his visit to NSA. What was the answer?



Clifford Stoll, Author of *The Cuckoo's Egg*



UNIX® AND OPEN SYSTEMS USERS

The following page is a copy of "the answer." It also contains diagrams on the Hubble mishap and "secret" Chinese writing (yes, I had it translated). The palindrome is really quite logical when someone like Cliff explains it and if you say the numbers out loud (three ones, two twos, one, one,) the next in Morris' series is:

312211

You can take it from here...

Cheers and as Cliff would say,
"Warm Smiles!"

*"... we fooled around with puzzles and palindromes for a while, until he wrote out this series of numbers:
1, 11, 21, 1211, 111221.
Complete the series, Cliff."*

A U U G I N C . S E C R E T A R I A T
P O B O X 3 6 6 K E N S I N G T O N
N S W 2 0 3 3 7 0 G L E N M O R E R O A D
P A D D I N G T O N N S W 2 0 2 1 A U S T R A L I A
I N C . I N V I C T O R I A A C N A 0 0 1 6 6 3 6 N
P H O N E + 6 1 2 3 6 1 5 9 9 4 F A X + 6 1 2 3 3 2 4 0 6 6

Vol 15 No 1

1 ONE ONE
1 1

Two
2

ONES

1

ONE TWO

ONE ONE

E

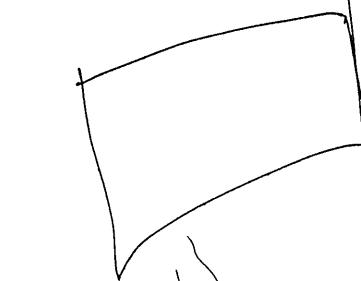
1 2 1)

1 1 2 2)

3 1 2 2 1

1 3 1 1 2 2 2 1

1 1 1 3 2 1 3 2 1



此

金

天

地

山

水

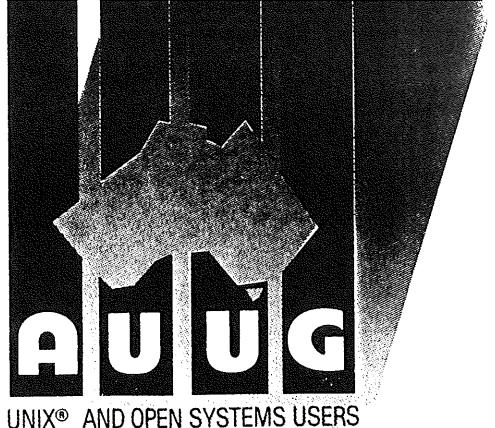
火

土

木

3 3
3 3
1 1
2 2
4 4
可
王
申
理
山
水
火
土
木
山
水

Announcement →



AUUG UPDATES AGREEMENTS

Sydney, NSW --25 January 1994

We are pleased to announce all vendors supplying discounts to AUUG members in 1993 have extended their agreements through 1994.

We wish to thank the following vendors and suppliers for their support:

- ComTech: Training Courses, contact is: Ian Schoefield - (02) 317-3088
- Connect.com.au: Network Connection, contact is Hugh Irvine - (03) 528-2239
- DIAL.ix: Network Connection, contact is Jeff Johnson - (09) 244-3233 or (02) 948-6918
- MHS: Network Connection, contact is Elaine Pensabene - (02) 550-4448
- NetComm: World Blazer Modems*, contact is AUUG Secretariat - (02) 361-5994
* Price for 1994 has been dropped to \$1500.00!
- Prentice Hall: Books, contact is Sandra Bendall - (02) 939-1333
- Softway: Training, contact is Elizabeth Mahey - (02) 698-2322
- Newly announced in this issue is Strategic Publishing Group: Publications *MIS, & Informatics*, contact is Peter Helft - (02) 267-2084.

AUUG members are also invited to attend affiliate organisation events (such as UniForum and USENIX) at member prices!

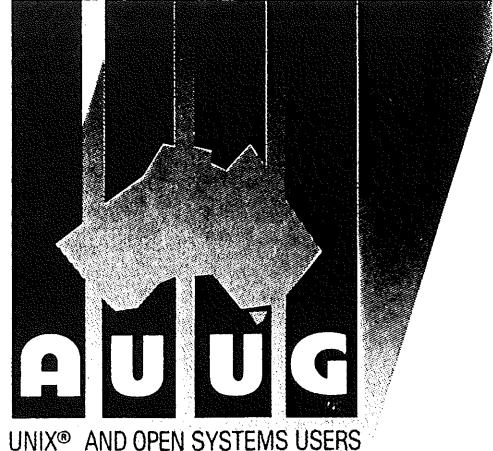
If you have suggestions for other services or equipment you would like access to at a discount or are currently with a corporation who could provide discounts to AUUG members please contact AUUG on the number or email listed below.

For further information:

AUUG Inc.
(02) 361-5994
e-mail:whfoda@acms.auug.oz.au

A U U G I N C . S E C R E T A R I A T
P O B O X 3 6 6 K E N S I N G T O N
N S W 2 0 3 3 7 0 G L E N M O R E R O A D
P A D D I N G T O N N S W 2 0 2 1 A U S T R A L I A
I N C . I N V I C T O R I A A C N A 0 0 1 6 6 3 6 1
P H O N E + 6 1 2 3 6 1 5 9 9 4 F A X + 6 1 2 3 3 2 4 0 6 1

Announcement →



AUUG & STRATEGIC PUBLISHING GROUP REACH AGREEMENT

Sydney, NSW -- 5 January 1994

In a continuing effort to improve benefits for members, we are pleased to announce Strategic Publishing Group and AUUG have reached an agreement to provide discounted subscriptions to members. "We are very pleased the relationship has evolved with the publishers of one of the leading IT publications, *MIS*, and the ACS's *Informatics*," said AUUG Business Manager, Liz Fraumann.

Managing Director of Strategic Publishing Group, Alistair Gordon states, "We propose to post a copy of *MIS* magazine to all AUUG members in March 1994, together with a special letter and subscription form."

AUUG members who have not yet subscribed to the magazine may subscribe during 1994 for \$49 per annum. This offer also extends to those interested in subscribing to the ACS publication *Informatics* as well.

If any portion of your contact details have changed since December 1993, please submit the Notification of Change Details found in this issue to ensure you will receive your copy of *MIS*.

For further information:

AUUG Inc.
(02) 361-5994
e-mail:whfoda@acms.auug.oz.au



**OPEN SYSTEMS.
LOOKING INTO THE FUTURE.**

**Australian Unix &
Open Systems
Conference and
Exhibition 1994**

Conference and Exhibition Secretariat

70 Glenmore Road

PO Box 468 Paddington

NSW 2021 Australia

Telephone (02) 332 4622

Faxsimile (02) 332 4066

World Congress Centre

Melbourne Australia

7th to 9th September 1994

CALL FOR PAPERS

**Australian UNIX and Open Systems Users Group
Annual Conference, September 6-9, 1994
Melbourne, Australia**

"Open Systems. Looking into the future"

Over the past several years we have heard about 'Open Systems' and the benefits it can bring for organisations facing the integration challenges of the 1990's. Looking to the future, we can see many emerging (and existing) technologies such as client/server computing, global networking, high performance computing and object oriented software that are of importance and concern to users and IT managers around the globe. This years theme, "Looking into the future" seeks to highlight these areas.

AUUG '94 solicits papers on all aspects of UNIX and Open Systems, and particularly on successful applications and implementations of Open Systems technology.

Events:

AUUG '94 will be a four day conference, commencing September 6, 1994. The first day will be devoted to tutorial presentations, followed by three days of papers, work-in-progress and product update sessions, and BOFs.

Tutorials:

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



**OPEN SYSTEMS.
LOOKING INTO THE FUTURE.**

**Australian Unix &
*Open Systems
Conference and
Exhibition 1994***

Conference and Exhibition Secretariat

70 Glenmore Road

PO Box 468 Paddington

NSW 2021 Australia

Telephone (02) 332 4622

Faxsimile (02) 332 4066

World Congress Centre

Melbourne Australia

7th to 9th September 1994

Papers:

AUUG '94 provides dual Technical and Management tracks for the presentations.

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

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

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

Prize for the Best Student Paper:

A cash prize of \$500 will be awarded for the best paper submitted by a full-time student at an accredited tertiary education institution.

Work-in-Progress (WIP) and Product Update (PU) Sessions:

A great success in '93, these brief 15 minute sessions are designed to report on current work with fundamental aspects highlighted, and to allow the chance for vendors to 'advertise' products. Product specification sheets should be submitted with your abstract.

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

Are you interested in discussing particular problem areas, sharing arcana on favourite programs, using the internet, or other controversial topics? During the lunch hour and at the end of each presentation day, one hour time slots for BOFs will be available. We distinguish two types of BOF; general interest and vendor sponsored. Please contact the Committee if you

A U U G
1 9 9 4

**OPEN SYSTEMS.
LOOKING INTO THE FUTURE.**

**Australian Unix &
Open Systems
Conference and
Exhibition 1994**

Conference and Exhibition Secretariat

70 Glenmore Road

PO Box 468 Paddington

NSW 2021 Australia

Telephone (02) 332 4622

Faxsimile (02) 332 4066

World Congress Centre

Melbourne Australia

7th to 9th September 1994

would like to organise a Birds-of-a-Feather Session. There may be some facilities charge to vendor sponsored events.

Speaker Incentives:

Presenters of papers are afforded free conference registration.

Product Update, Work-in-Progress and panelist presenters get single day registration and can 'top up' to get full attendance.

Tutorial presenters may choose 25% of the profit for their session OR full conference registration.

Form of Submissions:

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

1. Author name(s), postal addresses, telephone numbers, FAX and e-mail addresses.
2. A biographical sketch not to exceed 100 words.
3. Abstract: not to exceed 100 words
4. Outline: 1-4 pages giving details of the approach or algorithms pursued. Shorter outlines will not give the programme committee enough information to judge your work fairly, and, in most cases, this means your paper will be rejected. Longer outlines and full papers simply cannot be read by the committee in the time available. However, you may append a full paper to your outline; this is sometimes useful during evaluation.



**OPEN SYSTEMS.
LOOKING INTO THE FUTURE.**

**Australian Unix &
Open Systems
Conference and
Exhibition 1994**

Conference and Exhibition Secretariat

70 Glenmore Road

PO Box 468 Paddington

NSW 2021 Australia

Telephone (02) 332 4622

Faxsimile (02) 332 4066

World Congress Centre

Melbourne Australia

7th to 9th September 1994

5. References to any relevant literature
6. Audio-visual requirements: 35 mm slides are preferred, however, overheads will be accepted. Hand written or typewriter generated overheads will not be accepted.

Acceptance:

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

All participants in WIP, PU and BOF sessions will receive written confirmation once accepted.

Conference Committee:

Ian Hoyle - BHP Research (chair)
Hugh Irvine - connect.com.au
Craig Bishop - Geelong District Water Board
Colin Kempter - Wellfleet
Adrian Booth - Adrian Booth Consultants
Michael Paddon - Iconix P/L
Phil McCrea - ANSAMS

Relevant Dates:

Abstract and outlines due: March 18, 1994

Notifications to authors: April 11, 1994

Final Papers due: July 29, 1994



**OPEN SYSTEMS.
LOOKING INTO THE FUTURE.**

**Australian Unix &
Open Systems
Conference and
Exhibition 1994**

Conference and Exhibition Secretariat
70 Glenmore Road
PO Box 468 Paddington
NSW 2021 Australia

Telephone (02) 332 4622

Faxsimile (02) 332 4066

World Congress Centre

Melbourne Australia

7th to 9th September 1994

Addresses:

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

AUUG '94 Programme
P.O. Box 366
Kensington NSW 2033
AUSTRALIA

e-mail: auug94@bhp.com.au

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

Tutorial abstracts to: auug94-tutorials@bhp.com.au

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

x/Open®

Committed to Open Systems

New Publications
New Publications
New Publications

C206 Management Protocol Profiles (XMPP)

34p. \$72.00

This Management Protocol Profiles CAE Specification addresses communications using the CMIP and SNMP management protocols. RPC-based mechanisms will be addressed in future documents. This document supersedes the XMPP Preliminary Specification, which was published in January 1992.

C303 ACSE/Presentation Services API (XAP)

254p. \$126.00

XAP is an Application Programming Interface to the connection-oriented services of the Presentation Layer of the OSI protocol stack, including access to the ACSE application service element from the Application Layer. X/Open has defined this API as an interface to support portable implementations of application-specific OSI services and non-OSI applications. This specification describes the XAP API and defines the functions and data structures which it provides for use by applications.

C307 Data Management: SQL Remote Database Access

76p. \$108.00

This specification relies on the ISO/IEC RDA SQL standard, which specifies a message format for remote communication of SQL database language statements (query and update) to a remote database. This specification defines uses of the message fields and other implementation information such as sequencing and optional features. It shows how SQL statements map to the Remote Database Access (RDA) protocol. This document is the result of a joint collaborative effort between the X/Open data Management Working Group and the SQL Access Group

C318 X/Open Transport Interface (XTI), Version 2

264p. \$126.00

The X/Open Transport Interface (XTI) specifies a transport service interface that is independent of any specific transport provider. It is concerned primarily with the ISO Transport Service Definition (connection-oriented or connectionless). However, it may be adapted for use over other types of provider. In particular, this specification includes Appendices showing how XTI maps to TCP and UDP, NETBIOS, minimum OSI functionality (mOSI), and SNA. This Version 2 of the XTI Specification replaces the previous version published in January 1992. The main body of the specification remains unchanged from that previous version, the main differences being the addition of Appendices covering mOSI and SNA.

G207 Systems Management: Reference Model

68p. \$108.00

This guide is one of several documents within X/Open's Systems Management programme (XSM), the primary aim of which is to promote development of management software that allows an administrator of a distributed system to manage a network of heterogeneous systems as a single logical system. This Guide provides an architectural overview of the Systems Management model, and identifies the various components of the model. It employs object-oriented techniques. This is a high-level model which encompasses both OMG-compliant object-oriented technology and the OSI interaction, and indicates an approach for coexistence of both models.

G302 Systems Management: Managed Object Guide (XMOG)

82p. \$108.00

This guide is one of several documents within X/Open's Systems Management programme (XSM), the primary aim of which is to promote development of management software that allows an administrator of a distributed system to manage a network of heterogeneous systems as a single logical system. This Guide introduces the essential nature of managed objects, and the necessary framework for defining them. It discusses the managed object definition and development process, and explores the issues involved in registering them. It also discusses conformance testing.

G307 Distributed TP: Reference Model, Version 2

44p. \$120.00

This document provides a functional description of the X/Open Distributed Transaction Processing (DTP) model, a software architecture that allows multiple application programs to share resources provided by multiple resource managers, and allows their work to be coordinated into global transactions. It describes the use of the DTP model within the X/Open Common Applications Environment (CAS) and is a prerequisite to other present and emerging X/Open documents that address DTP. This document supersedes the previous version of the Reference Model published in 1991 (Doc. No. G120). It has been updated to take account of DTP interfaces developed by X/Open, particularly with regard to how Communication Resource Managers (CRMS) fit within this X/Open DTP model.

P303 Data Management: SQL Call Level Interface (CLI)

210p. \$126.00

This document defines the SQL Call Level interface (CLI). It provides application programmers with an application programming interface (API) for database access. Both C and COBOL bindings are given. Readers should be familiar with the X/Open CAE Specification, Structured Query Language (SQL), Doc. No. C201. This document is the result of a joint collaborative effort between the X/Open Data Management Working Group and the SQL Access Group.

P312 X/Open DCE: Remote Procedure Call

666p. \$162.00

This document specifies Remote Procedure Call (RPC) services, interface, protocols, encoding rules and the Interface Definition Language (IDL). The purpose of this document is to provide a portability guide for RPC application programs and a conformance specification for RPC Implementations. This document includes text excerpted and/or derived from the Open Software Foundation Application Environment Specification for Distributed Computing (AES/DC) with the permission of OSF.

Q931 X/Open Technical Programme

\$270.00

The X/Open Technical Programme is a binder with separately bound parts that describes the Technical Programmes to be undertaken by X/Open over the next three years. Part 1 provides a Technical Roadmap. Parts 2 through 12 each describe a major technical programme area. It is primarily intended for users interested in implementing an open systems strategy and in planning the development or procurement of open systems solutions; for vendors (both software and systems vendors) interested in pursuing an open systems product strategy; and for systems integrators interested in providing open systems integration services and facilities.

For the full publications list, or to place an order, contact Tony Blackmore:

Tel: 03 879 7412, Fax: 03 879 7570 or Email 100036.1124@compuserve.com

~~Announcement~~ →



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

WHAT: Berkeley 4.3 - 4.4BSD Workshops

WHEN: February - March 1994

WHERE: All states

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

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

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

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

One Day Workshop:

An Introduction to UNIX Kernel Internals: Data Structures and Algorithms

Who Should Take this Course

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

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

more-more-more

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

Description

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

Session 1 - Kernel Overview

- Kernel terminology
- Basic kernel services
- Process structure

Session 2 - Kernel Resource Management

- Virtual memory management
- Scheduling
- Signals

Session 3 - Kernel I/O structure

- Special files
- Terminal handling
- File system implementation
- Network File System (NFS)

Session 4 - Networking and Interprocess Communication

- Concepts and terminology
- Basic IPC services

Course Text

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

more-more-more

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

Three Day Workshop (presented only in Melbourne):

UNIX Kernel Internals:

Data Structures, Algorithms, and Performance Tuning

Dr. Marshall Kirk McKusick
University of California at Berkeley

Who Should Take this Course

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

Description

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

Day 1 morning - Kernel Overview

- Kernel terminology
- Basic kernel services
- Process structure

Day 1 afternoon - Kernel Resource Management

- Virtual memory management
- Paging and swapping
- Scheduling
- Signals

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

more-more-more

Day 2 morning - Kernel I/O structure

- Special files
- Terminal handling
- Multiplexing I/O
- Autoconfiguration strategy
- Structure of a disk device driver

Day 2 afternoon - File System Overview

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

Day 3 morning - System Tuning

- Performance measurement
- System tuning
- Crash dump analysis
- Security issues

Day 3 afternoon - Interprocess Communication

- Concepts and terminology
- Basic IPC services
- Example use of IPC and network facilities

Course Text

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

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

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

Announcement →



For further information contact:

Organisers listed or,

Glenn Huxtable - VP - (09) 380-2878 tel

Peter Wishart - Sec. - (06) 275-0908 tel

Liz Fraumann - Business Manager (02) 953-3542 tel/fax

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

WHAT: AUUG Summer Conference Series

WHEN: February - March 1994

WHERE: All states

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

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

Melbourne: *Venue:* Clunies Ross House

Feb. 3-5 McKusick Workshop

Organiser: Stephen Prince

Chancery Lane Computer Services
Level 25
385 Bourke St.
Melbourne, 3000
(03) 608-0911 ph 608 0505 fax
email : sp@clcs.com.au

Darwin: *Venue* - NT University

Feb. 9 McKusick Workshop

Feb. 10-11 Conference & Workshops

Organiser: Phil Marker

Dept of Computer Science
Northern Territory University
P.O. Box 40146
(089) 466-382 tel • (089) 270-612 fax
e-mail: pjm@pandan.us.ntu.edu.au

Hobart: *Venue* - University of Tasmania, Hobart, Centenary Lecture Theatre

Feb. 15 Conference

Organiser: Steven Bittinger

ITS
University of Tasmania
GPO Box 252C
Hobart, TAS 7001
(002) 207-406 tel • (002) 207-488 fax
e-mail: Steven.Bittinger@its.utas.edu.au

Canberra: *Venue* -Work shops - Copland Bldg., ANU

Venue -Conference - National Convention Centre

Feb. 14 McKusick Workshop

Feb. 15-16 Workshops & Conference

Organiser: John Barlow

CISR, IBLOCK
ANU 0200 ACT
(06) 249-2930 tel • (06) 249-0747 fax
email : john.barlow@anu.edu.au

Perth: *Venue*: - Orchard Hotel

Feb. 22 McKusick Workshop

Feb. 23 Conference

Organiser: Adrian Booth

Adrian Booth Computing Consultants
7 Glenrowan Place
Willetton, WA 6155
(09) 354-4936 tel • (09) 354-4936 fax
e-mail: abcc@dialix.oz.au

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

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

more-more-more

Adelaide: *Venue - TBD*
Feb. 28 McKusick Workshop
Mar. 1 Conference

Organiser: Michael Wagner
Systems Services Pty. Ltd.
32 Grenfell St.
Adelaide, SA 5000
(08) 212-2800 tel • (08) 231-0321 fax
e-mail: mhw@syserv.com.au

Sydney: *Venue - TBD*
Mar. 3 McKusick Workshop
Mar. 4 Conference

Organiser: Brenda Parsons
P.O. Box 43
Darlinghurst 2010
(02) 131-003 Ext. 2403 tel
(02) 808 2797 fax
e-mail: parson@coulomb.pcc.oz.au

Hobart: *Venue - University of Tasmania Staff Club (Hobart)*
Mar. 9-10 McKusick Workshop

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

Melbourne: *Venue- Clunies Ross House*
Mar. 15-16 Tutorials & Conference

Organiser: Arnold Pears
Computer Science and Computer Engineering
La Trobe University
Bundoora, VIC 3083
(03) 479-1144 tel • (03) 470-4915 fax
e-mail: pears@latcs1.lat.oz.au

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

Brisbane: *Venue - TBD*
Mar. 24 McKusick Workshop
Mar. 25 Conference

Organiser: Michael Henning
CiTR
University of Queensland 4072
(07) 365-4321 tel • (07) 365-4399 fax
e-mail: michi@ctr.uq.oz.au

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

UPDATE: UNIFORUM NZ

UniForum NZ '94 Programme Looking Good!

All I can say about the response to the call for papers this year is .. AMAZING! Consequently we have been able to put together a really great programme. We have six tutorials - one full day and 5 half day, several keynote and plenary sessions, our normal two streams plus a third stream consisting of double-length workshops. We only stopped there because we ran out of space at the hotel to hold everything!

We approached some of the main IT education companies this year and suggested they may want to present a tutorial based on one of their normal courses. To our delight they are all coming to the party. We have a full day "Introduction to Unix" tutorial presented by Alan Robson of Auldhouse Computer Services, a half day tutorial on "Object Oriented Systems and Relational Databases" by John Watson of Software Education Associates, and another half day on "Working with UnixWare - A Practical Workshop" by Michael Scott of Datamatic Networks. A unique opportunity to obtain top, professional education at our normal modest tutorial rates.

As well as those mentioned above we have three more half day tutorials scheduled. Greg Rose from RoSecure Software in Australia is presenting an "Introduction to OSF DCE", an overview of the uses for DCE and the major building blocks which constitute it, and a popular tutorial at AUUG last year. John Paynter, Auckland University, covers "Setting up EDI in Organisations with Particular Reference to Message Design", and Robert Biddle of Victoria University is again presenting "Moving from C to C++: What Do You Gain?" as this topic has been extremely popular at the university over the past 12 months.

A keynote focus will be on the European scene. Peter Idoine of IBM NZ arrived back from a seven-year stint overseas just in time to put together a paper for UniForum NZ '94! Using a variety of case studies, Peter will describe "How Europe is Getting the Open Advantage" and provide a few lessons for New Zealand based on these experiences. Unfortunately at the time this was written we were still waiting for final confirmation from the other two keynote speakers so these are still under wraps - sadly not everything fits neatly into editorial deadlines - but we'll keep you posted!

Our plenary sessions also promise some interesting material. On Thursday afternoon Dan Young of GCS Ltd will clue us up on "New Communications Trends for Global Business Challenges". Friday afternoon could see some interesting interplay from our two back-to-back plenary speakers - Peter Brown from Novell, Australia, will be presenting "Novell's UNIX Strategy" and Dennis Freeman of Sunsoft will cover "Solaris for Intel". By the way, both speakers have been warned about the consequences of including "Sales-Pitch" content in their presentations but pack your water-pistols just in case!!

In response to the demand for more time for technical workshops we have introduced a new stream this year devoted entirely to workshop sessions. These sessions will run for one and a half hours - the equivalent of two stream sessions back-to-back. We have an excellent lineup of workshops: Australian Greg Bond covers "Xterminals from Workstations - Why and How"; Ray Brownrigg is back with his popular and extremely useful workshop on "Unix Matchmakers - Regular Expressions"; John O'Gorman exposes the mysteries and wonders of "The Korn Shell"; Brent Summers discusses "Perl - What, Why and How?" and Anand Raman enlightens us on the "Key to Unix Expertise".

The stream sessions cover a wide range of topics. We have nine management-oriented papers, six technical papers and five In the combined category (management overview with light technical content). Particularly pleasing is the number of user papers that have been offered (and I didn't even have to nag this year!). Christine Major of Wattles Frozen Food will be discussing their implementation strategy. Chris Goodyer of the Fortex Group will relate his experiences "Towards the Promised Land" and Bob Walker of MAF Fisheries , after 8 years of open systems, decides whether it was a cost or a benefit. Bruce Miller of Methanex NZ is back with "The Aftermath of Downsizing", an update on his presentation last year. Alby Cartner of NZ Police, one of our most popular technical presenters last year, is also back again with a lighter, combined category paper "Icarus - Managing a Critical Project without Crashing and Burning".

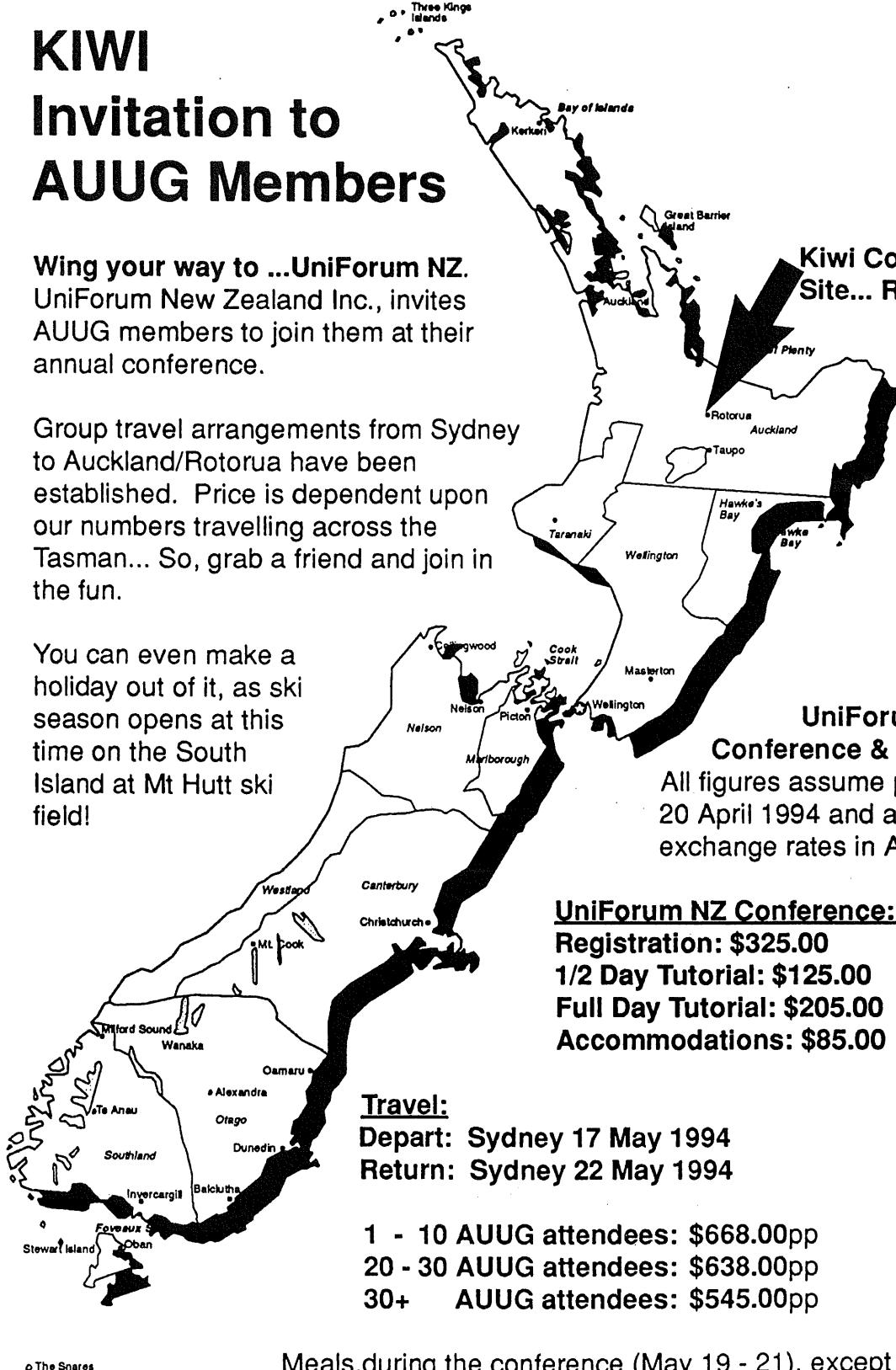
Other topics covered in the stream sessions include re-engineering, cluster management, connectivity, client/server, quality management, security and more. There is something there for everyone this year. The full provisional programme will be in the Registration Brochure which will be available mid-March from the AUUG Secretariat.

KIWI Invitation to AUUG Members

Wing your way to ...UniForum NZ. UniForum New Zealand Inc., invites AUUG members to join them at their annual conference.

Group travel arrangements from Sydney to Auckland/Rotorua have been established. Price is dependent upon our numbers travelling across the Tasman... So, grab a friend and join in the fun.

You can even make a holiday out of it, as ski season opens at this time on the South Island at Mt Hutt ski field!



Kiwi Conference
Site... Rotorua!

**UniForum NZ
Conference & Travel Package**
All figures assume payment prior to
20 April 1994 and are at current
exchange rates in Australian dollars.

UniForum NZ Conference:

Registration: \$325.00

1/2 Day Tutorial: \$125.00

Full Day Tutorial: \$205.00

Accommodations: \$85.00

Travel:

Depart: Sydney 17 May 1994

Return: Sydney 22 May 1994

1 - 10 AUUG attendees: \$668.00pp

20 - 30 AUUG attendees: \$638.00pp

30+ AUUG attendees: \$545.00pp

Meals during the conference (May 19 - 21), except breakfast are included in the conference registration. In fact, previous attendees have been noted to claim a stone or more accompanying them on their return travels!

If you are interested in the NZ experience, please contact the AUUG Secretariat (02) 361-5994 for registration information and Mike Wilson at Gentry Traveland on (02) 906-7000 for travel.

**INTRODUCTORY
OFFERS FOR
AUUG MEMBERS ONLY**

\$99*
**SINGLE LICENCE ONLY
(Save \$400)**

*Limit 1 copy per AUUG Member

"Works Over Any TCP/IP Transport!"

AIR SERIES 2.0

**Microsoft
and Novell
TCP/IP included
FREE!**

Internetworking Applications for Windows

SPRY Inc. is proud to announce the AIR Series, the first suite of Windows connectivity applications to offer Native integration into both your Novell or Microsoft Networking environments.

Why choose Air?

- 100% Windows DLL
- Installation is a snap
- Modular Product Structure

Transport Features

- Winsock Support • SLIP/PPP
- Native ODI and NDIS • DNS
- Netbios Support • SNMP

Internet Navigation Applications Available

- AIR MAIL™
- AIR News™
- AIR Gopher™

***Call today for more information
on the AIR Series 2.0!***



Host System Support

- DEC: VMS and ULTRIX
- HP UX
- IBM: AIX, VM, MVS
- UNISYS
- Sun Solaris
- HP LM/X
- BSD & System V Release 4

Applications & Features Available:

- Telnet • Network File Manager™(ftp) • NFS
- X-Windows Server • AIR tn3270™ • Line Printer Redirector
- FTP Server • RCP Server • NetWare Virtual Terminal (NVT)

SPRY™
Authorised Partner

ZIRCON
Systems

925 BOTANY ROAD
MASCOT NSW 2020
Phone: (02) 317 4055
Fax: (02) 669 3241

INTERNET
info@zircon.oz.au

Open System Publications

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

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

For example:

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

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

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

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

AUUG Inc. - Victorian Chapter

(formally SESSPOOLE)

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

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

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

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

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

Hope we'll see you there!

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

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

Update on AUUG Inc. - Victorian Chapter Activities

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

Well, my first column for 1994, happy new year to everyone.

What an eventful couple of months it's been. Late last year our secretary *Neil Murray* was gathering information on venues. Whilst in Clunies Ross House, he discovered that an accredited body will receive special benefits. So he set the wheels in motion to have us accredited, and on December 2 we received confirmation that we are an accredited body with the *Clunies Ross Memorial Foundation*. This basically gives AUUG-Vic a "home" for meetings, discount on venue facilities, plus numerous other benefits. The one side effect of it all, is that they decided to accredit AUUG as a national body and all it's chapters. Well done Neil.

Regular Meetings

AUUG-Vic has enjoyed the festive season with a social meeting each side of christmas, and is now ready to commence another year of alternate meetings. The next meeting, a technical one, is scheduled for the 2nd March. There are no definite topics or speakers at this stage.

Kirk McKusick Workshop

At the time of this writing, the workshop is less than one week away. As with any conference organisation, the work load seems to intensify as the day draws closer. I'll definitely be looking forward to that quiet drink when it's all over. :-)

As most of the Victorian members have realized, the "dated" registration form arrived very late. This was the result of some very unforeseen problems. Condolences from the AUUG-Vic committee go out to *Stephen Boucher* and his family.

Even with some "fire fighting", the registrations have been lower than anticipated, but are still flowing in at a steady rate from all around the country. All in all, it's shaping up to be an interesting three days, with lots of goodies for the participants.

Summer94 (Vic) Conference

This is certainly a hive of activity at present. The programme committee is currently putting together the final programme. IMHO it's looking great, with talks on real life experiences on Motif, a CASE tool, C++, network backups and a Virtual Reality application. The tutorial programme still hasn't been finalized. We hope to have registration

forms available, if not during the Kirk workshop, then shortly afterwards.

If you would like to know more, volunteer to speak or just volunteer your services, the respective contacts for the two events:

Tutorials: Robert Sturrock
15th March rns@deakin.edu.au
+61-052-27-2108
Computing and Communications Services
Deakin University
Geelong VIC 3217

Conference: Arnold Pears
16th March pears@latcs1.lat.oz.au
+61-3-479-1144
Computer Science Department
La Trobe University
Bundoora VIC 3083

Elections

Yes, it's getting very close. The hand over from the current to the new committee will happen after the summer conference. We hope some people are seriously considering running for the positions of office bearers, otherwise all the hard work by the current committee will just go down the drain after this term. For those thinking about nominating, I've placed a copy of the Victorian chapter rules on yarrina.connect.com.au for anonymous FTP. This PostScript document details the jobs each committee member is responsible for, and gives some general guidelines specific to the Victorian Chapter.

Other News

As most of you have no doubt heard, the AUUG Business Manager, *Liz Fraumann* is returning to the U.S. at the end of February. A lot of the work in running a chapter would have been much harder if it wasn't for the excellent job which Liz has done. The Victorian chapter would like to wish her all the best. We will miss you.

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

From the Western Front

Anyone want to buy an old XT clone? Last weekend, having got my troff macros -mpoem working, I moved the last of my home wordprocessing to my Unix system (BSDI's BSD/386 to be precise, and yes, that's a plug). No more DOS! No more switching printer cables about!

My Unix system is connected to Glenn's via the home Ethernet, which is teaching us about the difficulties of domestic systems administration. Now as well as who does the ironing we have to work out who does the backups. It turns out that in both cases, I won't do his, and he won't do mine. :-)

WAUG's contribution to AUUGN is rather light this time, especially if you don't count the parts written by me. This doesn't mean nothing's happening, though. The Perth summer conference is almost upon us. Adrian Booth, the organiser, has been trying to get people to talk and sponsor, and judging by the conference programme mailout has managed it pretty well. For several months Glenn Huxtable, who is the national summer conference coordinator, has done a lot of work to get Kirk McKusick's workshop itinerary right — making sure all the states get to have a workshop in conjunction with their conference, without running poor Kirk into the ground. (Yes, this is the same Glenn mentioned above.)

Mark Baker, our meeting organiser, continues to do an excellent job. We had a most enjoyable meeting in January: a presentation on CA-Unicenter that generated a lot of audience response. (See my review elsewhere in this issue.)

I could really use some WA people to contribute to AUUGN — especially to review and/or summarise our meetings. If you would be prepared to commit in advance to reviewing one or more meetings, please contact me.

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

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

Follow Sun's upgrade path to Motif

When Sun made the decision to switch to Motif - whose Motif did they choose?

It shouldn't be too much of a surprise.

They went to the same company that has already supplied IBM, DEC, Data General, NCR, ICL, NEC and Bull with Motif products for use on their workstations. And that has long championed Motif as the best way to get all UNIX workstations to look and feel the same.

Namely IXI.

What do you get by following Sun?

The latest, most advanced version of Motif available. It uses less memory, so runs your programs faster. And to make sure it stays the latest version, we update it free every quarter.

We provide the Motif Window Manager and shared library "toolkits" optimised for Sun workstations, including compatibility with Sun's Open Windows software. IXI's Motif is available for SunOS 4.1.x and Solaris 2.x on SPARC and Intel.

For further information, call us today on (02) 878-4777.



Advanced User Systems Pty Ltd

WAUG Meeting Review

We have no review of the December meeting: did we have one:-? (I was at Rottnest at the time.)

January

CA-Unicenter: Management Tools for Unix?

Peter Waterhouse, Computer Associates, & Paul Tayler, AMS

This meeting raised the heckling of graphical systems administration products, always a fun activity, to the level of a competition sport.

Unlike most such products, CA-Unicenter appeared comprehensive enough to generate an unprecedented number of audience questions. Peter did an admirable job of keeping the slide presentation on track while answering most of the questions satisfactorily (with occasional help from Paul). However when he got to the hands-on demonstration, and we could actually see the product in action, all hell broke loose.

We should have had prizes for most persistent line of questioning, for most convoluted scenario invented to test the product's mettle, and for most "hits" (where the speaker responded "In the next release").

Unicenter runs on HP Unix systems and is being ported to several others, but still not all of the ones I have to manage. Since it requires kernel modifications (to trap user actions so they can be checked against Unicenter's security criteria), I guess they've got to get each vendor to co-operate.

Unicenter is meant to help manage the usual things — security, disk space management, backups, user administration, accounting,...

The security features seemed quite comprehensive — possibly a bit over-the-top for some sites (for example, restricting what root can do is a mixed blessing). However, according to the speaker, the product is rated at only C1.

Unicenter comes from a mainframe background, which may be why it seemed to me to be rather heavy-duty. It really takes over the system. It didn't appear to be possible to plug in a replacement for one of Unicenter's components. The close integration of components was claimed as a strength by the speaker, but it could be a problem if you wanted to do something a bit differently.

Unicenter's graphical interface forms a "console" for the systems administrator or operator (you can set up several different customised consoles for different staff). Even on a big workstation screen this console looked extremely busy — I wouldn't want to sit looking at it all day.

There's another reason I wouldn't want to do that — I prefer my workspace to be as blank as possible, with alerts appearing (or being mailed) only when something is wrong, and tools running only when I need them. The rest of the time I just don't want to know. From the presentation, it wasn't clear to me whether Unicenter can be configured that way.

Unicenter has command-line versions of the GUI operations, which may be used to write scripts. The commands appeared to be pretty ugly — counterintuitive names and lots of parameters — but automatically generating them didn't seem to be an option. Automation (add 500 users while I'm at lunch, and mail me if anything goes wrong) looked difficult.

To summarise my opinions on Unicenter:

- Unicenter will do most of the things you want, if you don't mind doing them Unicenter's way.
- It suffers from the same problem as most commercial systems administration packages: the customer can't add any significant new functionality to it. The scripting facilities are cumbersome and the GUI is fixed.
- Since the source is proprietary, it doesn't look like it will ever run on every kind of system I'd need it on.

To summarise my opinions on the meeting:

- It was nice to see another vendor-sponsored meeting.
- It was an excellent talk, both as a presentation and as an interactive question-and-answer session. I think we got a good idea of what the product can and can't do, without any hard-sell. The WAUG audience haven't been so fired-up in years!

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



Cognos Pty Ltd
(Inc. in NSW)
A.C.N. 002 909 248
Level 3
110 Pacific Highway
St Leonards NSW 2065
Telephone: (02) 437 6655
Fax: (02) 438 1641

January 21, 1994

Ms. LIZ FRAUMANN
BUSINESS MANAGER
AUUG, INC.
22/74-76 MURDOCH STREET
CREMORNE NSW 2090 NSW 2090

Dear Ms. FRAUMANN,

If you're struggling to meet user demands with ever-diminishing resources, then *OpenTalk* is for you.

OpenTalk is a free, half-day workshop that shows you how to deliver quality, open applications faster than you ever thought possible.

Designed for MIS professionals, *OpenTalk* takes you through the process of building applications that run on both UNIX and proprietary platforms using the world's leading 4GL, PowerHouse. You will see how easy it is to build industrial-strength, client/server applications that support both Windows GUI and character-based terminals with the same applications.

OpenTalk also focuses on providing end-users with access to corporate data. Demonstrating the latest EIS and desktop query tools, it will explore the benefits of GUI-based applications.

The session is interactive and will take into account your specific requirements. For this reason, space is strictly limited. By the end of the morning, you'll have the necessary knowledge to plan and build business-critical applications.

Dates and locations: **February** **March**
 1 & 15 8 & 29

Location: Cognos Pty Ltd
 Level 3
 110 Pacific Highway
 St Leonards

Time: 9.00a.m. - 12.00Noon (lunch to be served)

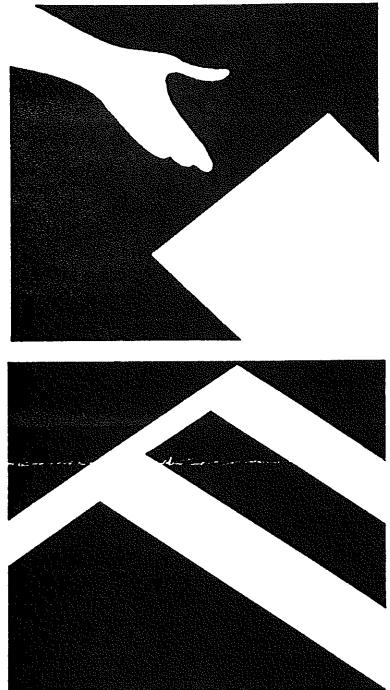
To register, simply complete and return the enclosed registration form by fax on (02) 438 1641, or call Meagan Lavender toll free on 008 811 910.

Yours sincerely
Cognos Pty Ltd

Meagan Lavender

Meagan Lavender
NSW, Sales Co-ordinator

REGISTRATION



COGNOS OPENTALK WORKSHOP REGISTRATION FORM

Please register the following people for the Cognos OpenTalk Workshop on:

- | | |
|--|---|
| <input type="checkbox"/> Tuesday, 1 February 1994 | <input type="checkbox"/> Tuesday, 8 March 1994 |
| <input type="checkbox"/> Tuesday, 15 February 1994 | <input type="checkbox"/> Tuesday, 29 March 1994 |

Name: _____

Title: _____

Name: _____

Title: _____

Company: _____

Address: _____

State: _____ Postcode: _____

Telephone: _____ Fax: _____

Please complete and fax back to Meagan Lavender (02) 438 1641

REPORT ON '94 WINTER USENIX

by

Greg Rose

The first USENIX event of 1994 was held in San Francisco on the 18th to 22nd of January. The format was reversed from that of previous years, with the three days of technical sessions on Monday through to Wednesday, followed by two days of tutorials. The USENIX Board of Directors met on the Sunday preceding. The SAGE Board met on Thursday. For ease of composition, I will write this report in chronological order.

I arrived on Friday the 15th, because the aircraft was full on the Saturday. This meant that I had all day Saturday to kill, so I spent it at the Zoo, which was not one of the world beaters. The wildlife on the public transport system was fairly interesting though. On Saturday evening I found the lobby of the Hilton was full of USENIX people checking in, to take advantage of the much cheaper airfares available with a Saturday night stay. General bar activities then proceeded.

There is a new card game called simply "Magic", which seems to be a licence to print money for the originators. From Saturday night on, there seemed to be no time at which there wasn't a game being played somewhere in the Hilton lobby.

On Sunday the Board of Directors of Usenix had their normal meeting, which I attended as an interested party, as a candidate for the upcoming election (that is a hint for any readers who are USENIX members!) and lastly as an informal representative of AUUG and SAGE/AU.

There was some amount of contention at the board meeting, with the Board finally acknowledging that the association is facing a mid-life crisis. Although the extent of the immediate trouble was not known, it was very clear that the attendance at this meeting was going to be poor, and this has severe financial implications for Usenix -- not only was their direct income down, but they get the actual meeting rooms free based on filling accommodation at the hotel, and face a penalty if the allocation of sleeping rooms is not filled. While I still don't know for sure whether this happened or not, I expect that they fell short.

The reason for the bad attendance at the conference was mostly put down to the lack of a very strong program, which was in turn put down to the combination of (a) too many general conferences, (b) too many small workshops and symposia, and (c) the close proximity of the preceding LISA (Large Installation System Administration, run by SAGE) conference. The general shape of the proposal to fix the problem was to reduce to one general conference per year, run some of the smaller events at the same time and place, and to make sure that LISA, which was for the first time bigger than the general conference, was geographically and/or spatially separated from the main conference.

Unfortunately, the contractual commitments to hotels and so on make it very difficult for USENIX to react to the problem with any rapidity at all.

There was a lot of discussion about provision of "member services", which evoked a sense of *deja vu* in me. While much of the discussion focussed on traditional things like software collections, more content in the publications, discounts on things, etc., I can't help but feel that there was less call for such services when the organisation was active in standardisation (and it wasn't fashionable), and when generic BSD manuals were printed when manufacturers were reorganising their own versions. In other words, I think the search for member benefits needs to be based on things which the organisation is particularly suited for, rather than things which could be provided by anyone. After all, CD-ROMs full of software could be (and are) produced by lots of people for profit, so there is no good reason for them to be produced by user groups. Organising and passing on discounts is a good, but not overwhelming, thing.

As an aside, there were still a few "Mentally Contaminated" badges around. This reminded me to give an update of the state of the USL vs. BSDI lawsuit. A decision was expected quite some time ago, but has not yet been handed down. Nobody appears to know what the holdup is, but most people are impressed with the judge's understanding of the issues, and feel that he is just making an effort to be very correct.

As a further aside, I also heard that someone is using a pre-existing software exchange agreement to block the transfer of the UNIX trademark from USL/Novell to X/Open. Guess who? Microsoft! Part of System V.4 is

based on Xenix. Grunt.

Monday saw the keynote speech from John Perry Barlow, one of the founders of the Electronic Frontier Foundation. He is an excellent and amusing speaker, but also very thought provoking. One of the hot topics at the moment is the Vice President's proposed Data Superhighway (this Usenix's badge said "Don't Become Road Kill on the Data SuperHighway"). Barlow spoke at length about the lack of understanding at high levels of the implications of current policy towards privacy and cryptography. The talk was also broadcast on the Mbone, but at that time of day I imagine few Australians were watching (kre?).

The first technical session had two papers by Udi Manber of the University of Arizona. These were "Finding Similar Files in a Large File System", and "GLIMPSE: A Tool to Search through Entire File Systems" (with Sun Wu co-author). Udi was also the author of "agrep", a sort of grep with approximate matches. These papers are best summed up by quoting a question from the audience: "Where do you come up with these great ideas for tools?". Read these papers and watch for Manber. The latter won the "best paper" award.

While these papers were in progress I was flitting to the "invited talks" room, where Marc Donner of Morgan Stanley talked about "UNIX on Wall Street". Marc is an old Go-playing friend from our days at IBM Research, and he talked lucidly about issues of system management in a world-spanning "it had better be back up in ten minutes" network. He did become less than lucid, although no less convincing, when talking about vendor's hardware support contracts.

I skipped the next couple of sessions when some panic messages came in from Australia, but from my point of view they were not very important in content anyway. Unfortunately a lot of people seemed to concur.

On Tuesday the content picked up a little, although as usual I was concerned that the invited talks acted to draw the audience away from the refereed sessions, which is self defeating in the long term.

Bill Waite, one of the best known names in programming languages and compilers, gave a great talk entitled "Beyond Lex and YACC: How to Generate the Whole Compiler". Only afterward did I notice that his tie had a picture detailing a final approach to an instrument landing, so we talked about flying more than compilers.

I also heard the invited talk "The Facts About Fax", by Ed McCreight of Adobe. This cleared up an awful lot of things for me.

The afternoon session was probably the only one in which the papers track was better attended than the invited talks track. The invited talks track was a summary of the USENIX Symposium on Mobile and Location Independent Computing, held recently in Boston. The clear message was that Wireless is easy, but Mobile (with or without wires) was much harder. I'm writing this review sitting in a friend's kitchen in New Jersey, but getting it to Jagoda is an inconvenient problem to solve.

The technical track had both of the Bell Labs papers. Phil Winterbottom talked about "ACID: a Debugger Built From a Language", which was sort of a reverse direction look at a debugging tool. I still think that Winterbottom's papers are something to look for. Rob Pike then talked about "Acme: A user interface for Programmers", and this was a thoughtful and contentious paper as could be expected from Pike. He got the "best presentation" award for it, although I don't think it was deserved -- he just had far more attendees at his session than anybody else. Reputation does that I guess. Emacs evangelists were seen honing carving knives afterwards.

The late session was highlights from the LISA conference held in Monterey California, and work in progress reports. I missed both, although both had important content.

The Usenix reception was at the Exploratorium again, as it almost always is when in San Francisco. This is sometimes billed as the "best children's science museum in the world" and I don't think I could dispute that. I've been there about five times, and still very much enjoy it.

Wednesday Morning's Invited talk was "Video Compression - What Do You Do When Everything is Changing?" Although I still don't know how a discrete cosine transform works, I at least now understand how it lets you get ridiculous compression ratios. The talk was far broader than its title, and gave lots of useful overview of JPEG, MPEG II and HDTV, along with the standards nightmare surrounding them.

This was followed by Stephen Johnson, President of both Usenix itself and Melismatic Software, "Objecting Vol 15 No 1

to Objects". This was a very well reasoned discussion of the promises of Object Oriented Programming, and the areas in which it has failed to deliver. Time well spent.

Aside: in subsequent conversations, a number of people have mentioned to me that they are beginning to see an industry backlash against OOP, and to a lesser extent against C++ -- the same sort of reaction that killed ALGOL. Johnson's talk solidified this feeling for me. See more below about POSIX.

The last invited talks session was a review of the Tcl/Tk workshop. There is no question that these represent very important developments, but I am increasingly getting the feeling that people are missing the point. 130,000 line Tk programs are not what the language was intended to do, and it really is a pretty bad language for large applications like that. However, it is definitely addressing an extremely large problem area that has no other decent solutions at the moment.

Reading back on my notes and what I've just written, I have realised just how weak the technical program was. That is not to say that I didn't get a lot of value from the conference -- I did. But most of it was represented by the personal contacts (schmoozing) or invited talks. I hope it is not too late for Usenix to correct the trend.

The closing session had a number of prizes and some funny videos. The Conference Competition was to "define your own Industry Alliance, and their product". The winner was "AT&T Bell Labs and Lorena Bobbitt: Plan 4-and-a-half" with one of the runners-up being "Steven Spielberg and SUN: Jurassic Sparc". I can't remember the rest, but most were funny.

On Thursday there were tutorials, none of which I attended, as well as the SAGE board meeting. The old SAGE board was re-elected unchanged except that Carol Kubicki didn't run again, and was replaced by Paul Evans. This makes the board Pat, Pat, Paul, Paul, Peg, Steve and Elizabeth. Mind if we call you Psteve and Pliz? Seriously, though: Elizabeth Zwicky is now President. The others are Pat Parseghian, Pat Wilson, Paul Evans, Paul Moriarty, Peg Schafer, and Steve Simmons.

For the first time the LISA conference was also perceived to have had a weak program, although most of this was attributed to accepting too many papers in an attempt to fill a very ambitious multi-track program. I don't think they'll make that mistake again. Otherwise, SAGE and LISA are growing rapidly and meeting most objectives. This is an area where Usenix needs to (and will) concentrate more resources.

One of the things that SAGE has been asked to do is to take a more active role in the POSIX 1337 System Administration Standards activities. (This is the new number of 1003.7, in case you hadn't heard.) This committee became bogged down in specifying a distributed object mode for administration, when the existing practice had not even been agreed. I think this would be a worthwhile thing for SAGE to be involved in, but it is not quite clear whether they will, or how to do it either.

That was it for me at this conference. Subsequent feedback indicates that the tutorials were very well received, as usual.

The next Usenix conference is in Boston in June, and it is the 25th anniversary of the creation of Unix, so there are some real events planned. It is also the first on the east coast for some time; hopefully attendance will be up again, even if it is just a temporary thing. If you come to one, this is the one you want to come to...



Dear Site Administrator,

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

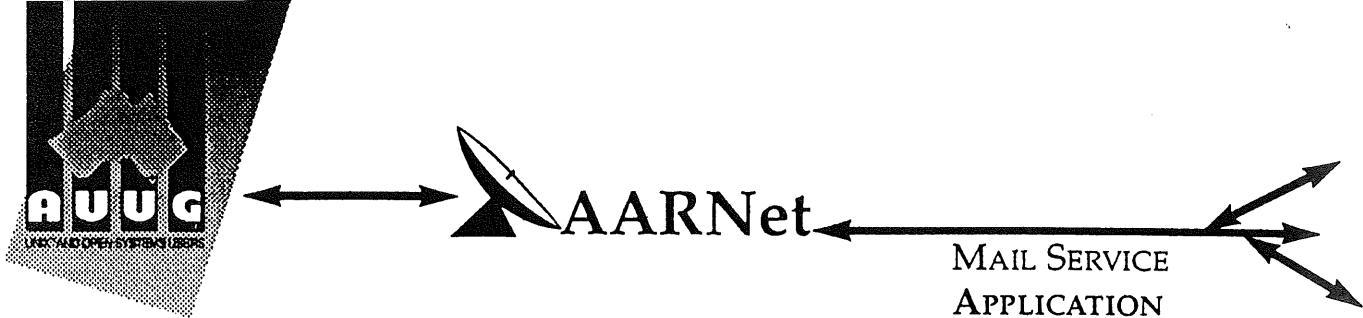
The fee is intended to cover use of AARNet bandwidth for your network traffic. Registration with AARNet, however, provides ONLY the registration of your address in worldwide address tables - your site will be unreachable without this registration. The fee does NOT cover the costs involved in obtaining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

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

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

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

Regards,
 Chris Maltby
 AUUG-AARNET Administrator
 AUUG Inc.



On behalf of the organisation listed below I wish to apply to be a Mail Service Affiliate Member of AARNet, and accordingly request that AUUG Incorporated arrange for the Australian Vice-Chancellors' Committee (AVCC) to maintain on my behalf an electronic mail delivery record in the Australian Academic and Research Network (AARNet) to allow my organisation to send and receive electronic mail carried across AARNet.

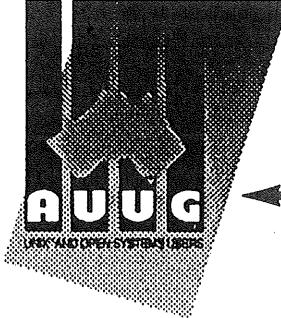
I understand that the AVCC may consult the recorded logs of my organisation's usage of AARNet facilities for 1990, and determine that I am ineligible for registration under the terms of the agreement between AVCC and AUUG Inc. I understand that AUUG Inc will invoice my organisation for this service for the calendar year 1991 and for subsequent years unless it receives my organisation's written advice to terminate the Affiliate Membership of AARNet.

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

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

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

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



AARNet

MAIL SERVICE
AFFILIATE MEMBERSHIP
APPLICATION FORM

PLEASE PRINT CLEARLY!

Date: _____

Name of Organisation/Owner: _____

Signed: _____ AUUG Membership No (if known): _____

Name: _____ Position: _____

on behalf of the organisation named above.

Address: _____

Postcode: _____

Administrative Contact: _____ Title: _____

E-Mail: _____ Phone: (_____) _____

Fax: (_____) _____

Technical Contact: _____ Title: _____

E-Mail: _____ Phone: (_____) _____

Fax: (_____) _____

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

Domain Names Requested: _____

Gateway Addresses: _____

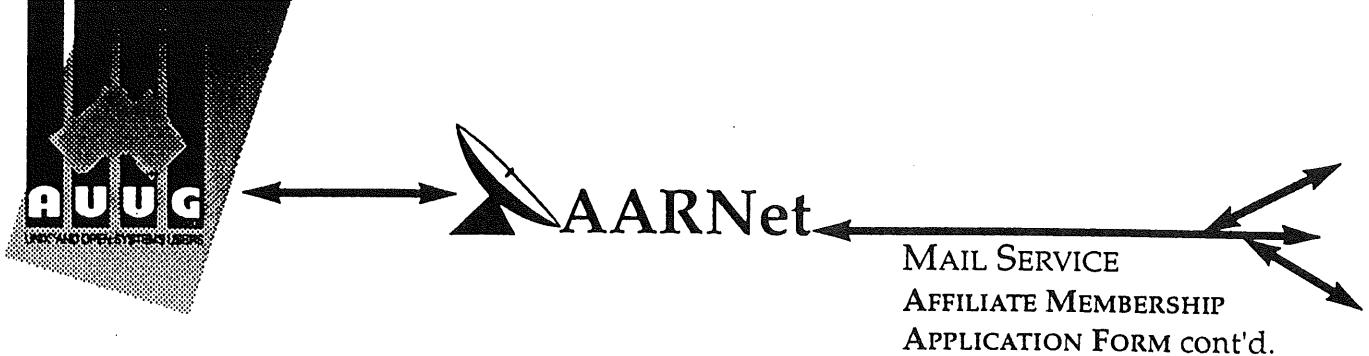
Expected Link Protocol: UUCP SL/IP MHSnet Other: _____

* * * *

Send this page only to:

AUUG Incorporated
PO Box 366
Kensington NSW 2033

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



Note A. Mail Delivery Information

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

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

Example Mail Directory Information request:

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

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

Note B. Getting Connected

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

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

Book Reviews

Welcome to the new year. Over the break I've had people busy doing reviews, with good results. This is probably the most notable set of reviews I've put together yet, with three major new releases, *The Magic Garden Explained*, *Learning Perl* and *Sendmail*, catering for most sections of the Unix market place. Aside from this we have other books from both Prentice-Hall and O'Reilly and Associates, to round out a very informative section.

If you are interested in reviewing books for AUUGN then let me know. The current practice is for me to post a note to the newsgroup *aus.org.auug* when we have new books available. Unfortunately, this disadvantages members without network connections, or on the end of a low speed link. For people in such a position, either mail, via the AUUG PO Box, or fax me on (02) 717 9404, with your contact details and preferences.

Frank Crawford

The Magic Garden Explained: The Internals of UNIX System V Release 4

by Berny Goodheart and James Cox
Prentice Hall
1994, 699 pages, \$59.95
ISBN 0-130981-38-9

Reviewed by
Peter Chubb
Softway Pty Ltd
<peterc@sw.oz.au>

At long last! You've read Bach's book, and Leffler et al's book, heard of the Lions commentary but — you're running SVr4 and the structures seen in the headers don't quite seem the same. Or you're one of those 'mentally contaminated' unfortunates who work with the UNIX source and have to come up to speed quickly on a new (to you) part of the kernel. Or you're a systems administrator with an SVr4 machine or two — just what do all those parameters you can set *really* do?

Now there is a book that goes through all the major data structures and control flow in the SVr4.0 kernel, in great detail. It gathers together in one (moderately large) volume, that is fairly easy to read, what would otherwise require searching through large numbers of Usenix proceedings, program headers, technical reports, other books, patents, etc., and tying it all together by inspired reading of the manual pages and (maybe) the UNIX source code. What does it tell you that you can't find out elsewhere without access to the source? Not a lot, actually. As such, it is surprising to find that USL has prevented publication of this book for around a year, costing Berny much wasted time and money in convincing them that, in fact, all the information in the book is publicly available elsewhere.

ORGANISATION

Part one of the book is introductory matter: what UNIX is, how it got there, and so on. The second part of the book (chapters three to six) describes

the core of the UNIX system: the memory management subsystem, process management, I/O systems and File management. The third part is anything else: Streams, Interprocess Communication, and how to use *crash(1)*.

Each section of each chapter in parts two and three is more-or-less independent, so readers wishing to find out about just one aspect of UNIX can use the fairly comprehensive index to find just the bits they need. This does mean, though, that people who want to read the book sequentially may be annoyed by material included more than once to make the sections able to be read independently.

PART ONE

The first chapter gives an introduction to the history of the UNIX system. Unlike the versions in, say, Bach's book, Berny's version gives due prominence to the contribution Australians made to the early development of UNIX. Australians were amongst the first users, performed the first port, and provided many of the early enhancements that helped make UNIX popular.

Chapter two gives an introduction to UNIX from a user perspective, and introduces standard concepts and terminology that are used throughout the rest of the book. Experienced users of UNIX SVr4 may be able to skim this section, rather than read it thoroughly; it does however discuss some of the differences between SVr4 and previous version of UNIX.

PART TWO

Part two covers the memory management, process control, I/O and file system management subsystems within SVr4.

Important fields in each data structure are named and described. Especially useful are the diagrams that show how data structures are linked together and the general text that explains how they are used.

Pseudo-code shows the algorithms for key functions in the kernel. Commentary on the pseudo-code adds further information about what each function is doing, how and by whom it is called, and any effect it has on global data structures. In some cases the routines shown are simplified, with no indication that this has been done. For instance, in the pseudo-code for *sigtoproc()*, the steps for removing any queued STOP siginfo structures are omitted. Despite this, the pseudo-code does indicate fairly clearly what is going on.

The pseudo-code functions merely list the inputs and outputs, without using the named symbols from the source. I would have preferred the actual names and types — especially now that ANSI C prototypes for many of the functions are in the public header files (although the prototypes do not include the symbol names). However, it was possible to work out which ‘input’ or ‘output’ variable was which in every case I tried.

Algorithms and data structures are described in great detail. This is especially good as a commentary on the source code, because in many places the code is obscure.

There are a few ‘chicken-and-egg’ problems involved in describing the kernel at this level of detail. For instance, interrupt priority levels are mentioned (and used) several sections before they are described. I don’t think that this will matter if the book is used as a reference (driven from the index or from the table of contents) but does reduce its value for reading from beginning to end.

PART THREE

I didn’t have time to read Part three as thoroughly as the rest, but a quick scan though indicated a similar attention to comprehensive detail.

The book ends with a large chapter on using *crash* to explore the system for yourself, and a bibliography for further research.

CONCLUSION

Overall, this is an ideal commentary on the source code, and a very useful reference for the systems administrator or systems programmer who does not have access to the source. I certainly learnt things by a careful reading of Part One together with the source.

The book is based on SVr4.0.3; later versions have not changed substantially in the algorithms they use — the book would be a useful commentary on the basic functionality of any SVr4 system. Of course, new features (like the secure file system *sfs*) are not covered.

If you want high-level description of what UNIX does ‘under the hood’, use Leffler (et al)’s book, or

Bach’s book. If you want a very detailed commentary on SVr4, with information useful for system tuning, debugging and enhancement, this is the book for you.

Highly recommended.

Reference.

Maurice Bach (1986): *Design of the UNIX Operating System*, Prentice Hall

John Lions (1977): *Commentary on the UNIX Operating System*, University of NSW

Samuel Leffler, Kirk McKusick, Michael Karels and John Quarterman (1989): *The Design and Implementation of the 4.3BSD UNIX Operating System*, Addison-Wesley

Learning Perl

by Randal L. Schwartz
O'Reilly and Associates, Inc.
1993, 246 pages.
ISBN 1-56592-042-2

Reviewed by
Jagoda Crawford
ANSTO
<jc@atom.ansto.gov.au>

If you are one of the many people who has heard of Perl, but has been uncomfortable trying to learn it from *The Camel Book*, then, *The Llama Book* is for you. *Learning Perl*, often called *The Llama Book* because of the animal featured on the cover, is intended as a tutorial style introductory to Perl, and in fact, is derived from an introductory Perl course prepared and presented by the author. As such, it is not intended as a comprehensive guide to Perl. For a reference book the author recommends the O'Reilly & Associates book *Programming Perl* (i.e. *The Camel Book* by Larry Wall, Perl's creator, and Randal Schwartz).

For those of you who are unaware of Perl, it is a language sweeping the Unix world (along with other operating systems) for text and file processing, system work and many other general uses (including Poetry!). It was written by Larry Wall, who is continuing the development, and is supported by thousands of people on the net. Randal is widely known on the net by his *Just Another Perl Hacker* signatures.

This book is written in a manner which does not assume knowledge of Unix, working thorough a series of *hands-on* exercises introducing the reader to Perl's many features. At the end of each chapter there is a set of exercises, a possible solution of which is provided at the end of the book. (One comment often made about Perl is that *there is more than one way to do it.*) For those

who are lazy, these solutions can be obtained by FTP, FTPMAIL, BITFTP and UUCP, as described in the book's introduction.

The writing is in a teaching style, not too formal, but set at a level to make it easy to grasp the concepts. There are 18 chapters, and Randal estimates that one should take 2-3 hours to work through a chapter. There are also some lighter sections of the book, especially the forward by Larry Wall and his introduction to the course on the *Magic of Perl* at the *College of Wizardry*, or Randal's poem in reply:

A one L Randal wrote the book,
A two L llama for the look,
But to whom we own it all
Is the three L Larry Wall!

In summary the book is well written and if you are interested in using Perl or you have used it, but are not totally confident with it, then this book is for you. It taught me how to use Perl and I'm sure it can do the same for you.

Sendmail

by Bryan Costales
with Eric Allman and Neil Rickert
O'Reilly & Associates, Inc.
1993, 792 pages
ISBN 0-937175-056-2

Reviewed by
Lucy Chubb
Softway
<lucyc@sw.oz.au>

UNIX and its attendant programs can be unruly beasts. Nutshell handbooks help you tame them says the colophon. They have certainly chosen an unruly beast to grace the cover of the Nutshell handbook on sendmail --- a flying fox. The colophon continues to describe flying foxes foraging for food: *Once a target is located, they are faced with a difficult landing. Sometimes they will simply crash into foliage and grab at what they can; other times they may attempt to catch a branch with their hindfeet as they fly over it and then swing upside-down; some will even attempt a difficult half-roll under a branch in order to grip it in the preferred position.* Awkward, isn't it.

The Nutshell handbook *Sendmail* is a whopping 792 pages long. Sendmail is, in the words used on the back cover, *one of the last great uncharted territories --- and most difficult utilities to learn --- in UNIX system administration.* Those who have met sendmail will not be surprised at this statement. Do you feel that you should have some help before you face sendmail? This book is the one you need.

The book covers the versions of sendmail that are most likely to be in use at any site. It also tells the reader where to obtain the source using ftp, and gives a table of which of the two most common versions are supported on each of a couple of dozen systems. The primary focus of the handbook is on the UIUC IDA version 5.65c (from [ftp.uu.net](ftp://ftp.uu.net)) and BSD V8 sendmail (from [ftp.cs.berkeley.edu](ftp://ftp.cs.berkeley.edu)), with mentions of other versions where the author deems it necessary. This makes the coverage fairly comprehensive.

The book is divided into tutorial, administration, and reference sections. The tutorial system leads the reader through sendmail in reasonable steps, with each section building on the previous ones. For example: what sendmail does in general terms, how to run it, the configuration file, and so on. Each chapter of the tutorial section has a "things to try" section that is based on the material in that chapter. These seem to be well formulated to lead the reader into an understanding of what is happening within sendmail and why. They include such things as experimenting with sendmail to determine its behaviour in the presence of certain types of errors, designing rules, reasoning about the effects of certain actions, and so on.

The administration section starts with how to compile and install sendmail, and then covers topics such as DNS, security, mail queues, aliases, and so on.

The reference section covers, as expected, much of the material in the previous sections but also contains an expanded section on debugging options. Tables are used where appropriate to summarise rule set operators, flags, and so on. They are followed by more detailed descriptions of each entry in the table.

The only additional thing I could have wanted is a quick reference guide consisting of all the tables printed together. However, the list of tables at the beginning of the book makes them fairly easy to find.

This is the book to buy if you want to learn about sendmail or have to administer it. I wish it had been available the first time I had to configure sendmail.

Practical Unix Security

by Simson Garfinkel and Gene Spafford
O'Reilly & Associates, Inc.
March 1993, 482 pages
ISBN 0-937175-72-2

Reviewed by
Chris Dale
University Computing Services
University of Newcastle
<cccd@alinga.newcastle.edu.au>

Another in the ever increasing (and diversifying) range from O'Reilly and Associates, this time an older edition that has somehow slipped through the review net.

Security is a huge topic and "Practical Unix Security" addresses the problem of providing and maintaining security under the Unix operating system. While it certainly doesn't attempt to be a one volume reference on computer security it covers nearly all of the need-to-know's when considering security for a number of common situations. In particular administrators of Unix systems should all be familiar with the contents of this book and the user who is concerned about the security of their work will also find material here to help them maintain a personally secure environment.

The book is split up into logical sections each containing a number of chapters. The sections are;

- I. Unix and Unix Security Basics
- II. Enforcing Security on Your System
- III. Communications and Security
- IV. Handling Security Incidents
- V. Other Security Topics
- VI. Appendices

Quite a bit of ground is covered here but the authors still manage to give a well rounded approach whilst not skimping on detail.

A basic description of the Unix operating system is given as an introduction although I would suggest that the reader be at least slightly familiar with Unix to get the most out of the book. A range of areas of security concern are addressed including denial of service attacks, break-in attempts, physical security, encryption and backup and disaster planning. Numerous examples and "how-to" sections are included thus making the book earn its "practical" title. A small portion of the book is geared towards an American audience (the section on Computer Security and U.S. Law) but this doesn't detract from the overall usefulness and it would be sensible to seek expert advice locally on your particular law/security related needs. A Unix Security Checklist is included as an appendix

summarising all of the security points covered in the book and allows you to consider each in relation to your system and needs.

The authors point out a number of times that the information contained in this book is not intended for those who want to break into computer systems. In particular exact descriptions of some security incidents are omitted and the text is written in such a way as to encourage responsible use. Having said this it is even more important that those concerned with security be familiar with this books contents in order to try and stay ahead of the few who have access to material such as this and also have malicious intent.

"Practical Unix Security" has been around for quite some time now (first printing June 1991) and has not been significantly updated although subsequent reprints have all had minor corrections. It is starting to date slightly and misses out mentioning a few packages that have recently become available, a notable omission being tripwire, Gene Spafford's own Unix file system integrity checker. However despite this, the basics don't really change that much and the book does a good job of making the reader aware of some of the potential problems. Material of particular interest can (and should) be followed up to reveal any recent developments using current computer security publications, CERT/SERT notices, various network newsgroups and mailing lists. Another small feature of the later print runs is the lay-flat binding, it certainly beats balancing a coffee cup on one side of the book to keep it open on your desk.

This book, as the name suggests, deals with aspects of security as they relate to Unix based computer systems. Practical approaches feature prominently making it a useful addition to any security conscious administrators (or users) library. If you want some solid background in theory of computer security try the other security book in the O'Reilly series "Computer Security Basics" by Russell and Gangemi but if you want to get the job done and learn more about security in the process, get this book.

DNS and BIND in a Nutshell

by Paul Albitz & Cricket Liu
O'Reilly & Associates, Inc.
March 1993, 381 pages,
ISBN 0-56592-010-4

Reviewed by
Lawrie Brown
<Lawrie.Brown@adfa.oz.au>

The Domain Name System (DNS) is another of those system utilities used by just about everyone on the Internet (for mail, telnet, ftp etc), but which

is maintained (like sendmail) by a handful of benighted sysadmins - this book is for them. The DNS is a globally distributed database with local control of local information, but with global access. It enables the translation of machine names like "ccadfa.cc.adfa.oz.au" into Internet addresses [131.236.1.2], which are subsequently used to route IP packets to that system. This allows humans to remember nice mnemonic names and let the computers do all the hard work. Someone however, has to tell the computers how this is done. The DNS configuration is simpler than sendmail (for which we can all be thankful), but is nonetheless less than intuitively obvious.

I approached this book as a systems programmer and user who has used the services of the DNS many times, and have even done my share of 'dig'ging into the system trying to locate machines. However I'd not previously had to configure and maintain a DNS server.

The book starts with a discussion of the background and theory of the Domain Name System. It then continues with a discussion on whether or not you should run your own domain, and if you decide to, how to set up your name server configuration files and run the server, how to have your parents delegate authority for your name space to you, and how the server interacts with email on your systems. The middle section contains several chapters on maintenance of your domain, configuring hosts to use your name server, planning for growth of your domain, and how to create sub-domains. The final chapters deal with troubleshooting tools, obtaining and interpreting debug output from both the tools and the name server, diagnosing problems, and finally some hints on the lost art of programming with the resolver library routines. The book then includes several appendices including excerpts from the relevant RFCs, compiling and installing BIND on a Sun, a list of top-level domains, and the templates for requesting delegation of a domain from NIC in the US (for those under US top-level domains, and for in-addr.arpa delegations).

The book is written in a fairly light-hearted style (complete with quotes from "Alice in Wonderland" and "Through the Looking-Glass"), whilst still providing a large amount of useful information. The authors include quite a few comparisons with other Unix paradigms to help explain their material, such as relating the domain name space to a file system, emphasising both the similarities and the differences. The authors also include lots of hints and traps for the unwary along the way, which alone would make the book invaluable.

All that being said, the bottom-line must be - does the book assist in establishing and running a name server? Well, the only way to find out was to try it - so after a few hours reading the first third of the

book, I dived in boots and all and set up a new sub-domain here at ADFA (xx.adfa.oz.au for the perennially nosy!). I created both a primary and a secondary server on a couple of 386bsd boxes here, with both forward and reverse lookup data, all correctly delegated. It took only a few hours initially, with more time spent over the next few days fleshing out the information, trying out some of the diagnostic tools, and reading the debug output. I found all the instructions clear and helpful, and had no real problems at all (admittedly I did have access to the files on our existing name servers, however I only needed a couple to help bootstrap myself in). Basically, the book does what it advertises!

In summary, this book attains the high standards we've come to expect from the Nutshell Handbooks. For the topic, its quite an entertaining read, whilst providing a comprehensive guide to creating and managing a domain name server. I would highly recommend it to any systems administrator who has to manage a domain name server, and also to any programmers who need to use the facilities of the domain name system.

**Network Administration
UNIX SVR4.2
(Administration Series)**

Edited by John A. Van Dyk
UNIX Press, Prentice Hall
ISBN 0-130176-33-8

*Reviewed by
Brenda Parsons
UNIX & Open Systems Consulting
<parson@coulomb.pcc.oz.au>*

If you're looking for the definitive book on Network Administration, this isn't it. However, it does fill a void left by other book series which, to date, have been primarily focused on the BSD or Sun variants of UNIX, or at public domain software packages.

As indicated by the "Edited by" phrase on the cover, the chapters in this book have been compiled directly from the AT&T SVR4.2 Manual Set, but unless you are one of the privileged, very rich, or borrow your company's set, then you probably won't own the twenty plus volumes in the manual set.

It is a "how to", book, rather than a "how it works" one, covering in adequate detail how the SVR4.x Networking facilities fit together. Most importantly it covers the features which are unique to SVR4 such as the Service Access Facility (SAF), and the Identification and Authentication Facility (IAF).

A welcome feature of the book is that all fields of control and log files are discussed in great detail and numerous examples are given for all topics in the book.

The topics covered in the book include a host of TLA's (3 letter acronyms): SAF, IAF, BNU, DNS, TLI, TCP, DFS, RFS, NFS, RPC and NIS. For each topic, both the command line interface and the sysadm interface are given, thus the book is suitable for both the novice and the more experienced administrator.

One of the hazards of generating a book such as this from other manual sets is that inevitably there are the reference to other manuals in the series. One such example is: "For more information, see 'Managing Ports' in Advanced System Administration", which coincidentally just happens to be another book in Prentice Hall's "Administration Series".

All in all, Network Administration for SVR4.2 is a good companion book to O'Reilly's TCP/IP Network Administration, UUCP and Usenet, and DNS and Bind books, as it covers the SVR4.2 aspects of the same subjects.

UNIX System V Performance Management

Edited by Phyllis Eve Bergman
and Sally A. Browning
Unix Press, Prentice Hall
ISBN 0-130164-29-1

*Reviewed by
Mathew Lim
ANU Supercomputer Facility
<M.Lim@anu.edu.au>*

I first started grappling with UNIX System V R.4 in 1991. Coming from a SunOS environment, I was immediately struck by how little information there was available back then on SVR4 systems administration. This situation was made even worse because we had a Beta release with manuals written in badly translated English.

Things have improved since then, with many books devoted to this subject. Including the book being reviewed here on performance management.

The book is about 350 pages long, quite a good size for me as my attention span only lasts for about 10. It's written in a fairly concise style which should make it a good reference book. It's broken up into 8 Chapters.

Chapter 1: Improving System Performance. Gives a quick overview of how to go about looking for performance problems.

Chapter 2: Users, Processes, and Workloads. This chapter points out some of the performance

issues surrounding users' profiles. It also has a good section on the SVR4 process scheduler, a subject which was of great interest to me. It then goes on to describe paging behavior and process accounting.

Chapter 3: The File System. Gives an overview of the s5, ufs and bfs filesystem types which are available under SVR4 and their characteristics. It gives some useful tips on the various options available when making each type of filesystem.

Chapter 4: Storage Devices. This chapter doesn't seem to deal very much with performance but is more devoted towards how storage devices are managed under SVR4. Things like disk formatting, device reservation, how to create a bootable disk are discussed.

Chapter 5: Communications. This chapter begins with a useful discussion of STREAMS in SVR4, it then goes on to discuss BNU (Basic Network Utilities a.k.a UUCP). I found this part rather disappointing. The rest of the chapter is a discussion of RFS and NFS which was rather more interesting than UUCP. It has some useful information the differences between these two protocols.

Chapter 6: Monitoring System Activity. I found this chapter to be most useful as it has a good description of the various SVR4 monitoring tools like sadc, sar and sa. It also gives a good description of what the output actually means, how to analyse them and how to fix the problems that they indicate.

Chapter 7: Performance Tools. This chapter gives a good overview of individual tools for monitoring the performance of the kernel as well as of user processes. These include timex and fusage (used for monitoring user processes), several kernel profiling tools, sadp (a disk activity profiler) as well as errdump and sysdump (used for taking memory dumps).

Chapter 8: Configuring UNIX. This is a very useful chapter describing the procedure for rebuilding your kernel and gives a good description of what the various tunable parameters do.

Quick reference. At the end of the book is a very useful Quick Reference Guide to Performance Management which gives you a table of performance management tasks, what tools or procedures to use to perform the task and the chapter in the book which deals with the issue.

In short, a recommended book for those contemplating a move to SVR4. It should make good reference material for both experienced and beginning sys admins.

AUUG BOOK CLUB & PRENTICE HALL AUSTRALIA

20% DISCOUNT TO AUUG MEMBERS

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

**AVAILABLE
NOW!!!**



***Goodheart/The Magic Garden Explained**
ISBN: 0130981389, Paper, 1994 , RRP \$59.95



***USL/Network Administration for Unix SVR4.2**
ISBN: 0130176338, Paper, 1993 , RRP \$65.95



***Bregman/Unix System V Performance Management**
ISBN: 0130164291, Paper, 1993, RRP \$61.95

*Deduct 20% from listed retail price

Name: _____ Organisation: _____

Address: _____
(Street address only) Telephone: _____

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

Enclosed cheque for \$ _____ (Payable to 'Prentice Hall Australia')

Please charge my: Bankcard Visa MasterCard

Credit Card No:

Expiry Date: _____ Signature: _____

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



OR Use our FAST PHONE SERVICE by calling Liz Guthrie.

SYDNEY (02) 939 1333

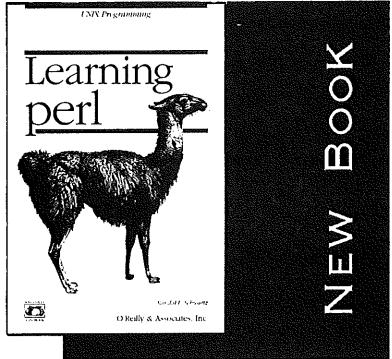
A.C.N. 000 383 406



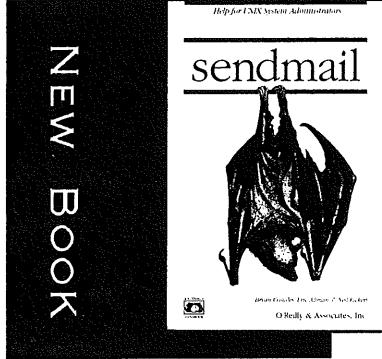
Prentice Hall Pty. Ltd.
7 Grosvenor Place, Brookvale NSW 2100.
Tel: (02) 939 1333 Fax: (02) 905 7934

A Paramount Communications Company

O'REILLY & ASSOCIATES, INC.



NEW BOOK



LEARNING PERL

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

PRICE: \$49.95

ISBN: 1-56592-042-2

SENDMAIL

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

PRICE: \$65.00

ISBN: 1-56592-056-2

AVAILABLE FROM AUSTRALIA'S LEADING TECHNICAL BOOKSELLERS

NSW: Dymocks George St Sydney 224 0411
Dymocks Pocket & Technical
Wynard Sydney 224 0444
Hotline Chatswood Sydney 415 2265
Dymocks North Sydney 956 6771

VIC: Tech Books Melbourne (03) 663 3951
McGills Melbourne (03) 602 5566
Dymocks Melbourne (03) 663 1344

QLD: Dymocks Edward St Brisbane (07) 220 0152

SA: The Bookshelf Adelaide (08) 223 3899
Dymocks Adelaide (08) 223 5381

WA: Rellim Booksellers Perth (09) 321 5684
Boffin Bookshop Perth (09) 321 5755

A.C.T. Uni Co-Op Bkshp ANU Canberra (06) 257 2673

Or call WoodsLane for the stockist nearest you.

Distributed by WoodsLane Pty Ltd
Phone: (02) 979 5944 Fax: (02) 997 3348 Email: woods@tmx.mhs.oz.au

A.C.N. 003 667 549

!AUUGN

The following appeared in AUUGN Vol 4 Number 3. It is interesting to see some of the forecasts. Has this happened yet?

Perkin-Elmer Supports Unix
Or
A Philosophical View of Unix
in the Commercial Environment
Or
Will Success Spoil Unix?
Joseph Longo
Perkin-Elmer, Melbourne

The recent announcement by Perkin-Elmer Australia to offer Unix as a standard supported product has been received with mixed sentiments. From the PE viewpoint, Unix offers a dual-pronged attraction :-

- it offers a sophisticated time-sharing development environment for PE's full range of superminis, an environment not offered by PE software until now;
- it also acts as a catalyst with the ability to bring PE's traditionally low profile out into the lime-light.

Relative to the outside world, the PE/Unix coupling is seen as :

- legitimizing Unix as a viable operating system to be taken seriously;
- a chance for a large cross-section of computer users to combine the fabled Unix development environment with low cost, high performance 32 bit hardware. (on a PE 3210 : 8 megb/sec DMA throughput plus 0.94 Mips, concurrently for around \$60k.)

But there is the negative view as well :-

- elements of the old guard within PE argue that the Unix offering is unnecessary. The argument is almost true in that there does not appear to be an overwhelming demand for Unix locally. But compare this to the U.S. where hardware vendors not able to offer Unix are out in the cold.
- a typical commercial Cobol-bound user will give you a million reasons why he doesn't need Unix. But have him work on it for a while and he'll give you two million why he can't live without it ! (Is there bliss in ignorance ?).

Very few people would seriously suggest that Australia will not follow trends in the U.S. where the adoption of Unix has snowballed. Even the worst of the Unix critics/cynics who last year were claiming Unix was useless because its I/O was not record-oriented, are this year sheepishly admitting that Unix will probably gain a foothold locally.

There are some aspects to the commercial use of Unix that are worth mentioning. Firstly, it brings the University computer departments closer to the commercial users in the real world. The two may not sleep together, but they will at least live together. The common criticisms that fly between universities and industry will become less frequent. Both of them, for the first time, will be talking the same language : Unix.

As a direct consequence, graduates will be immediately productive in their first jobs. They won't need to be 'de-tuned' or 're-trained' (brain-washed) before becoming acceptable to their employers. Job adverts may look something like :-

WANTED
PROGRAMMER/ANALYST
Must have Unix expertise,
preferably with tertiary qualifications;
Hardware experience irrelevant;
Promotion to SU plus access to
"/usr/games" after qualifying period.
Cobol programmers need not apply.

The flow of software between universities and industry will become a two-way street. Universities will make available a host of Unix-related software which until now was mainly academic, while industry will offer its own packages to the universities giving undergrads first-hand experience with the type of environment in which they are likely to work.

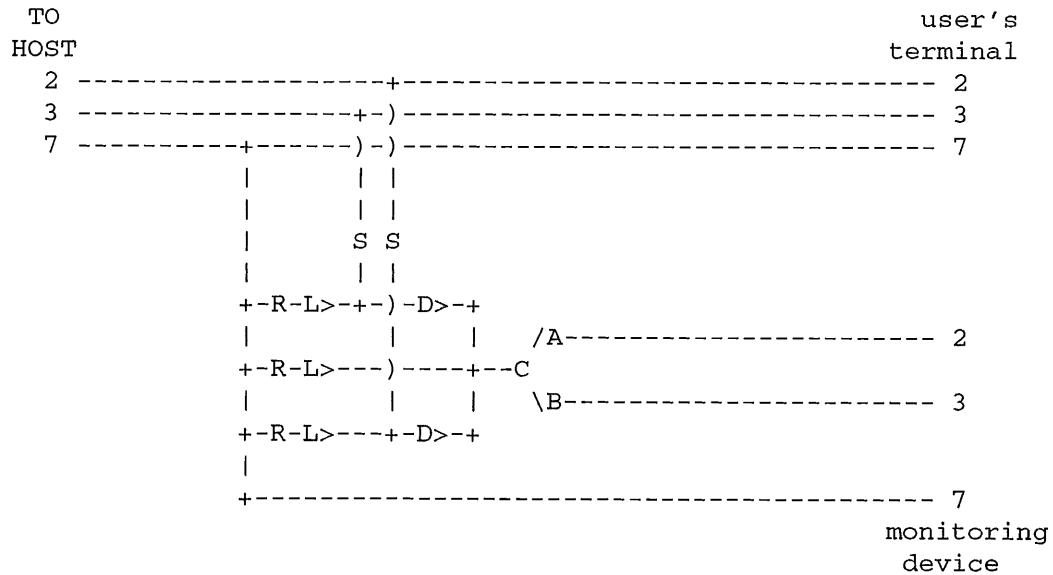
As far as PE is concerned, this software exchange will be something to look forward to. Hopefully, some of the PE commercial packages will be ported to Unix. The PE optimizing Fortran compiler, recognized as being one of industry's best, should be the first candidate, followed by other compilers. The PE transaction-processing/data-base package, Reliance, itself a highly successful development tool, would be an ideal Unix application. Take, then, some of the local and U.S. Unix enhancements (e.g. Melbourne U. mods, Berkeley mods, etc.) and plug them into PE's Unix. The result of all this commercial-academic interchange is a formidable operating system and development environment. Run it on a high powered PE supermini and you get a lot of people sitting-up to attention.

The concept of packaging Unix to sell in the commercial environment is an abhorrence to many Unix followers. At the same time, it creates the groundwork needed to bring the commercial users forward from 1956 and academic users down out of orbit. Acceptance of Unix by Perkin-Elmer sets a precedent which makes all of these discussions adopt a touch of realism. Undoubtedly, a lot of these forecasts have already taken place in the U.S.. With a bit of encouragement the same environment can be created in Australia within the immediate future.

Sam's Serial Listening Device (Mark 3)

This article describes a (relatively) simple hardware tool which was designed to permit serial traffic passing between a user's terminal and a host system to be observed on a separate screen or printer. This all occurs without interfering in any way with hardware or software on the host or user's terminal.

I hope that you find this Serial T-piece as useful in debugging hardware and software as I have.



LEGEND:

2 = Pin 2 of a 25 pin RS232 D shell (=pin 3 for 9pin) [Tx/D]
3 = Pin 3 of a 25 pin RS232 D shell (=pin 2 for 9pin) [Rx/D]
7 = Pin 7 of a 25 pin RS232 D shell (=pin 5 for 9pin) [Gnd]
+ = Wires join (or turn)
) = Wires cross but do not join.
R = 100 Ohm Resistors, 0.25 watt.
L> = Light Emitting diode (LED), arrow toward Cathode.
D> = Small signal diode, arrow toward Cathode. (eg IN914)
S = Simple on/off miniature switch.
C = Centre post (common) of a miniature two position switch.
A = One of the possible positions of the 2 position switch.
B = The other possible position of the 2 position switch.

Construction

Much of the construction detail really depends upon the size of the switches and resistors as well as your ability/agility with a soldering iron. However, as a guide, here is a description of mine.

The leads that pass from the host to terminal are essentially a DB-25 plug and DB-25 socket soldered back to back. From that, three leads about 15 centimeters (6 inches) long go to another DB-25 plug. Inside this plug is *all* of the electronics. Being built this way enables the Serial T-piece to be quite portable.

All pins from the host should be passed through to the users' terminal, that is pins 1 through 25. However, as only pins 2, 3 and 7 are tapped, only those pins/lines are shown in the diagram.

Hopefully I've drawn the polarity of the diodes (& LEDs) correctly. If it doesn't work as expected, just turn the diode or LED around.

Operation

The two small signal diodes prevent the Monitoring Device from transmitting, and the transmit data becoming receive data, and vice versa. This maintains the (illusion of the) integrity of the user's activity.

The two simple switches determine which directions are monitored. One can observe the user pressing keys, or watch the host's transmissions, or both, or neither (not all that useful!). The LED of each line being monitored lights up.

As the monitoring device may require pins 2 and 3 swapped, the two-position switch will direct the data to be sampled to the appropriate pin for the Monitoring Device. The LED associated with this switch illuminates when this switch is positioned *incorrectly*. Actually, all LEDs will illuminate when this switch is positioned *incorrectly*.

Configure your monitoring device so that "Functions" or ASCII codes are displayed to observe all data, and listen away. Depending upon the reliability and speed of your monitoring device, and the line speed, some data may in fact not appear on your monitor. This is of course a result of the monitor not being permitted to transmit xoff/xon or use hardware handshaking to slow the data stream (nor should it!). This minor inconvenience is usually outweighed by the value of the data that is observed.

Ethics

As this device does not discriminate about the data sampled, any user being observed should be made aware of the activity -- firstly because passwords or other unechoed activity will appear on the monitor when the user's keyboard activity is being sampled; secondly because this is an eavesdropping device!

It is not the object of this design to eavesdrop illicitly, but everything being equal, I'm sure it will be used to that end. However, the legitimate uses to which it can be put have proved to be valuable in determining a myriad of problems since Mark 1 was put into operation about 10 years ago.

Postscript

I tried to invent a catchy name for this serial T-piece, but all I could come up with was YABA (Yet Another Bloody Acronym). So I gave up there and then:-)

Happy New Year to everyone.

Sam Pascoe <samp@DIALix.oz.au>, 22 Dec 93
From WAUG, the WA Chapter of AUUG

A Dataflow Shell

or

A Workbench for a Tools-Based Mail System

Abstract

At the school of CS&E-UNSW we have replaced the traditional mail handler (*sendmail*) with a more flexible and more understandable tools-based mail system.

Any tools-based system requires a workbench to join the tools together, just as the "shell" does for traditional UNIX tools. This paper discusses a new command language interpreter which was developed to integrate the mail tools. It is different from the traditional shell in that it focuses on the flow of data rather than the flow of control.

1 Introduction

A *Dataflow shell* is a command language interpreter that focuses on the flow of data rather than the flow of control. It takes a collection of data and passes it to commands for processing, allowing for the processing to both change the data in the collection, and direct the further processing of the collection. For analogy with control flow programming, we will call such a collection of data a *Data Thread* or just a *thread*.

The term *shell* doesn't seem very meaningful in this context and is used only to draw on similarity with the Unix shell. *dataflow command language interpreter* would be more appropriate, but is also more wordy.

The motivation for creating this shell was the desire to write a tools based mail system. The author was not very impressed with the standard mail handling system supplied with Unix (*sendmail*) and decided to write a new one. Being a great fan of the tools-based approach to programming, it was decided to develop a suite of tools for doing different things with mail, and to integrate them into a complete system. This integration required more than the standard Unix workbench — the shell — and so the dataflow shell was developed.

The remainder of this paper describes the overall architecture of the dataflow shell, and the specifics of handling data and controlling flow. Examples, where used, are taken from the mail system currently in use at the School Of Computer Science and Engineering, UNSW.

2 Architecture

The shape, or architecture, of a dataflow shell turns out to be quite different from that of a traditional shell.

While there are a number of similarities such as the need for a script to tell the dataflow shell what to do and the fact that most real work is actually done by other programs, the underlying mechanisms are quite different.

The main reason for this is that the state of computation for a particular thread is better defined in a dataflow shell.

2.1 The State Of Computation

In a traditional process, the state of computation at any time involves not only the value of variables and the program counter, it can also include the state of a number of operating system services such as open file descriptors, child processes and signal handlers.

In a dataflow processes, there are a number of discrete instances when the state is completely described by the data in the thread, and the stage in the process that the thread is up to. As this state is so simple, it is quite practical to save the state in a file (or at least, in the filesystem). This provides for easy control over both the lifetime and the location of a thread.

2.1.1 Lifetime

As the state of a thread can be stored in a file, it is sensible to make the lifetime of a thread correspond to the life time of a file. That is, it will survive reboots or any similar interruption to the execution of any processes.

2.1.2 Location

As with files, there should be no difficulty moving threads from one machine to another. Providing that there is an appropriate dataflow "program" on another machine that will accept a given thread, it is quite possible to run part of a thread on one machine and part on another.

The thread could be transferred as a whole to the new machine (e.g. over a tcp/ip connection) or it could be accessed from the different machines via a network-transparent file system (such as NFS). In either case, the thread should be able to easily moved to where it can run most effectively.

2.1.3 Thread Creation

Creating a new thread will be done simply by creating a new file. Naturally the file will have to be in an appropriate place and have an appropriate format so that it will be noticed and understood by the dataflow shell. To create threads there is auxiliary program which will be described in some more detail later. It collects data from files or command line arguments, creates a data thread, and stores in somewhere. It also makes sure that the dataflow shell is running.

2.1.4 Implications

To support this flexibility in lifestyle, we need to be able to store and schedule a number of concurrent threads running in the one dataflow program. While the operating system is good as scheduling processes, it has no facility for scheduling files, so this has to be provided by the "shell".

This implies the need for a simple but flexible queuing system to manage multiple threads.

Indeed, it was out of a simple system for doing just this — storing jobs, and either forwarding them over the network or processing them locally — that the current dataflow shell concept arose. The system was and is called "Store and Forward" or SaF: a name that seemed appropriate at the inception but that unfortunately does not reflect the current state of the project.

2.2 Implementation Specifics

In the system as described, a given dataflow program requires more than just a file (dataflow script) to contain the code of the program. There must also

be a directory to contain currently active threads. The current implementation expects a dataflow program to be a directory containing a number of threads, one per file; the program code in a file with a distinguished name; and a directory for subsidiary files (as described below) with another distinguished name.

The current system also allows only one such dataflow program per host. This restriction is an historic rather than a design restriction and may well be relaxed later.

There are two main programs used to implement this dataflow shell, they are a program which creates a data thread, called *saf* and a program which interprets the program and schedules the threads, called *saf_daemon*.

When *saf* creates a thread it includes in it the name of the user who created it, the machine on which it was created, the time of creation, and a few other details that may be of use during the processing of the thread.

The *saf_daemon* is the dataflow shell. It periodically scans the program directory looking for threads that need to be scheduled. When such a thread is found, it is run through the next stage of processing. This involves extracting particular items of data from the thread, as described in the script, and passing them to a program (or pipeline of programs) for processing. The output of these programs, along with some of the data in the original thread, are then used to create the new state of the thread. This last step is actually done with the help of *saf*. Once the new state is successfully created, the old is discarded.

3 Data and Flow

When considering a dataflow shell, two concepts must be dealt with: Data and Flow. That is, how the data in a thread should be structured, and how the flow is to be controlled.

3.1 Data

The thread of data which is being moved around by the dataflow shell needs to have a structure which is restricted enough to be accessible by the shell, and yet free enough to be able to store whatever may be required.

3.1.1 Name/Value Pairing

The most obvious approach is a list of name/value pairs, not unlike the environment passed to Unix processes. This is easy to implement and provides for quite straight forward creation and decomposition of threads. Whether or not it is sufficiently expressive depends primarily on the type of values that can be associated with each name.

The current implementation only allows single character names.

3.1.2 Strings or Files?

The primary use that the shell will put the components of a thread to is as inputs to some program which processes the data in the thread. There are two ways that data can be given to a program — as arguments or as a file. It could be argued from this that there should be at least two value types, strings and files where strings are used for the values that would be passed as command line arguments, and files would be used for values that programs would expect to read from files. It turns out that this distinction is not really necessary and just serves to complicate things. It is quite easy for the shell to convert a string to

a file or vice-versa and so both of these data types could best be stored as strings which are converted to files when necessary.

3.1.3 Arrays!

It is common for programs to take a list of arguments which all refer to the same sort of thing (a list of files or of users etc.). This corresponds to the concept of an array so it seems that it would be useful to support arrays in the dataflow shell. This is quite easily done by allowing one name to appear multiple times in the data thread, each time with a (potentially) different value. This list of values makes up the values in the array. This has proved very useful in the mail system for storing lists of recipient addresses.

It is necessary for the values in a data thread to be extracted from the output of a program, so that threads can be changed by processing. For string type values, they can simply be read from some output file descriptor such as stdout. For arrays, the extraction is not so obvious. The current system allows an array to be read from a file/pipe using newlines to separate the elements of the array. While it might be desirable to have more control over how elements are extracted, this approach serves most situations well.

3.1.4 Integers?

Another type of data which comes to mind is the integer. However the only way to pass an integer to a program is by first converting it to a string. This means that an integer type would only be useful if the shell could do some arithmetic itself. As yet, no need for arithmetic has been found.

3.2 Flow

The traditional way of describing dataflow is with a dataflow diagram, with bubbles for the processors and arrows indicating the flow of data between them. Though attractive, a diagram like this is a bit awkward to type in and difficult for a program to process.

The language currently supported by the dataflow shell is at the other extreme of the spectrum and is very low level. Each bubble in the corresponding diagram is a named stage that a thread may be at, and with each stage is a pipeline to be run when a thread reaches that stage. Thus the dataflow program script is a list of stages, each of which has a name and one (or more) pipelines associated with it. The ordering of the stages is unimportant.

It is planned that a more aesthetic and easier to manage language will be developed in the near future to describe what is wanted rather than exactly how to do it.

Experience with the current language, as described below, has illuminated some of the matters that must be considered.

3.2.1 Commands

The centre-piece of the language is the simple command which describes how to pass parts of a thread to a program, and how to create a new copy of the thread based on the output of that program.

Passing parts of a thread to a program is described with a syntax similar to the Bourne shell and with substitutions reminiscent of printf(3).

A substitution is introduced by a percent sign % and includes the next two characters. This first character is the name of the value to substitute. The second is how to substitute that value. There are three options for the how:

- ‘s’ meaning simple string substitution,
- ‘f’ meaning substitute the name of a newly created file containing the string, and
- ‘L’ meaning substitute the name of a newly created file containing the named array.

If the name doesn’t exist in the thread, the substitution and any adjacent non-white-space text is deleted. If the name appears multiple times, then (except for with the ‘L’ option) the substitution and adjacent text is replicated appropriately.

Thus, the command which takes a new article (from the value named ‘a’) which has arrived via SMTP from the machine named in the ‘h’ value, with return address given in the ‘s’ value, and for recipients listed in the ‘r’ values, and checks that the headers are OK, would be

```
tidier -r%ss -h%hs -tSsmtp %rs <%af
```

If there are multiple ‘r’ values, they get substituted as multiple arguments.

Collecting the output into a new object is achieved by piping the outputs of the program into the *saf* program which builds new objects. This program can read strings and arrays from file descriptors (-d and -L flags), can set strings from command line arguments (-s flag) and can include in the thread the name of the next stage of processing (-H flag). Collecting the output of the above program into a new thread to be passed to the address rewriting stage would be:

```
tidier ... | saf -HAddress -da0 -ss%ss -sr%rs
```

This creates a new copy of the thread with the sender and recipients unchanged, and with a suitably modified article. Once this pipeline completes successfully (creating a new thread), the original thread is removed. If a command needs to generate more than one value, it sends them to different file descriptors. Multiple pipes can then be used to connect the processing program to *saf*. e.g.

```
processor ... | 2|4 3|3 saf ...
```

creates three pipes connecting: file 1 of processor to file 0 of *saf* (the default), file 2 to file 4 and file 3 to file 3.

3.2.2 Gotos

The only way to direct the flow of a thread once it has left a particular stage by indicating to the *saf* command which stage should come next. This is effectively equivalent to a *goto*. A distinguished name in each thread is used to indicate the name of the next stage of processing, and its value is set appropriately when a new copy of a thread gets created.

It is not clear whether a more structured approach would be possible or useful.

3.2.3 Forks

It is often useful, even essential, to cause a data thread to fork and follow each of a number of paths. This is effected quite easily by providing a list of processors for a given stage. The thread is passed to the first processor, creating a new copy of the thread headed somewhere, and then rather than deleting the original, it is passed to the next processor.

This is used a number of times in the mail system script. For example, when a mail article is to be delivered locally to some number of users three different forks must be followed.

- The article must be placed in the users' mail box.
- The *comsat* daemon must be informed of the arrival of new mail so that it can in turn inform the user.
- The article must be checked for a **Return-Receipt-To** header and a receipt may need to be generated.

3.2.4 Conditionals

Different threads may need to take different paths though a script, so it is necessary to provide conditional routing. In line with the Bourne shell approach, the choice of which of two possibilities is based on the exit status of a processor.

When the dataflow shell runs a pipeline, it makes the last process (which will always be an instance of *saf*) the parent of preceding processes. Thus *saf* can detect and act upon the exit status of the processor. Normally if the processor exits with a non-zero exit code, it is assumed that something went wrong and that the processing should be tried again later. If, however, *saf* is given an alternate next stage (-A flag), then an exit code of *one* will cause an object destined for that stage to be created, instead of for the normal (-H) next stage.

A good example of this from the mail system is after a mail article has been passed to ACSnet via the *sendfile* program. If *sendfile* returns a non-zero exit status, the thread must follow a path which returns an error report to the sender. Otherwise the thread can safely terminate.

3.2.5 Termination

A thread is terminated by supplying a processing command that does not run a *saf* to create a new copy of the thread.

3.3 Builtins

As with the Bourne shell, it is useful to provide a few commonly used pieces of functionality as builtins.

There are currently two functions that are built in for efficiency. One is a test to see if an object contains a particular name. The other is a processor to log passage of threads though particular stages. The syntax for these is neither appealing nor interesting and so will not be discussed here.

4 Further Work

SaF -- the dataflow shell is very much still under development. Many changes need to be made, not the least of which is choosing a better name.

Other changes have been hinted at earlier, but for completeness, they are all listed here.

- Names of values in a thread should be able to be longer than a single character.
- The syntax for writing a script needs a complete overhaul, making it easier to manage and at a higher level.

- It would be good if there could be more than one *saf* script per system. Currently there is only one in /usr/spool/SaF.
- As threads can easily move between machine, it would be useful to be able to monitor the status of the thread queue on a remote machine.
- There must be other possible uses for a dataflow shell than for processing mail, but no other good examples have surfaced.

UniForum NZ'94

The Open Advantage

11th Annual Conference

18-21 May 1994

Quality Hotel, Rotorua

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

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

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

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

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

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

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

Stashbase

Scott Merrilees
Sm@bhpese.oz.au

BHP Information Technology

ABSTRACT

Stashbase is a simple keyword based text retrieval system, implemented using the UNIX† file system as a DBMS. Each item in the database is stored as an individual file linked to multiple path names generated from the supplied keywords. Various shell scripts provide insertion, removal, keyword maintenance and lookup facilities.

The problem: How to organise and index all the useful snippets of information *stashed* over the years such that the information is easily accessible?

A solution: Stashbase.

Stashbase is a form of text retrieval system. It isn't very smart, it isn't very big (currently 8 shell scripts, totaling 514 lines of code), but it IS useful.

Stashbase is built around the idea of using the UNIX file system as a DBMS and using the UNIX shell to build some simple interface tools. The basic unit of storage is a file, the basic index a directory entry. Since UNIX allows an individual file to have multiple directory entries, or links, all that is necessary to insert an entry into Stashbase is *mkdir* and *cat*. Similarly, *ls* and *cat* can be used to search Stashbase. In practise, interface scripts such as *dbi* for insertion, and *dbv* for viewing are usually used.

Implementation

The Stashbase database is implemented as a hierarchical tree in the UNIX file system. The root of the tree is defined by the environment variable \$DB.

Each item inserted into Stashbase is assigned a unique monotonically increasing item number, which is zero-filled to 8 digits and used as the file name of the item within the database. The last assigned item number is simply kept in

the file *\$DB/item*. Each item is initially created in the *\$DB/item* directory, to allow lookup by item number.

Item indices are simply directories with an extra level, composed of the first character of the keyword, to keep the size of the *\$DB* directory manageable. Thus, the index directory for the keyword *keyword* is *\$DB/k/keyword*. When indexing a new item, any necessary index directories are created, then the item is simply linked from *\$DB/item* into the appropriate index directories.

This method of storage is very similar to that used by the various USENET news systems implemented under UNIX, except that a unique item number will be assigned for each article. Instead of a newsgroup hierarchy, all indices are at one level.

If an item is added to the database using both *ftp* and *mail* as keywords and *\$DB/item* contained 931 before the addition, then after the addition *\$DB/item* would contain 932, and the following directories would exist:

```
$DB
$DB/item
$DB/f
$DB/ftp
$DB/m
$DB/m/mail
```

also, a file would be created named by the follow-

† UNIX is a trademark of Bell Laboratories.

ing path names:

```
$DB/item/00000932  
$DB/ftp/00000932  
$DB/mail/00000932
```

Interfaces

Initially *dbi* was the only interface script written. *Dbi* is used to insert items into the database and index the item to the specified keywords. Lookup was initially via *ls* and *less*.

After **Stashbase** grew to about 50 items, doing a lookup using *ls \$DB/?* lost some of its appeal. The increasing complexity of the database required the creation of *dbf* and *dbv*.

Dbv views the items in the database that match the keywords specified as arguments. *Dbv* is basically a front-end that paginates the filenames returned by *dbf*. *Dbf* basically accepts the same arguments as *dbv* and returns the absolute path names of the matching items.

Next to be written was *dbl*, which lists the items or keywords in the database. If keywords are specified, each keyword and the associated items are listed. If items are specified, each item and the associated keywords are listed.

After manually adding a series of keywords and deleting others, it became apparent that a program was needed to automate the picky details of keyword addition and deletion. Hence *dbk* was created to allow keywords to be added or deleted from an item already in the database.

Next *dbr* for removal and *dbe* for editing were written. When *dbe* was written a potential problem was revealed. As each item is inserted into the database, by *dbi*, it is created with mode 444. Most editors can not edit read-only files. This problem was solved using a combination of *ls*, *sed*, *sh* and *m4*. These were used to calculate the new protection relative to the current *umask*.

At the time of writing the last interface to be written was *dbc*, which is a cross-reference tool. Given a list of keywords *dbc* will return all items referenced by those keywords, including keywords referencing those items. This may take a while, especially with larger databases on slower machines.

Design Issues

Given the simplicity of the design requirements for **Stashbase**, it would have been of similar complexity to implement the indices as a file

rather than use the file system. The file system has the advantage that UNIX does the maintenance and consistency checking. While it is easier to do simple searches on a file, using a file means that it must be updated atomically, this is not too much of a problem when doing an append, but more of a problem when wanting to add and delete indices.

A potential problem was how to lock the *\$DB/item* file while updating it. This was an issue as initially the database was on one machine, and most of the items destined for it were on another. The items to be sent to the database machine were encoded as electronic mail. On receipt by the database machine the items were automatically inserted into the database. Thus it was possible for 2 items to be simultaneously updating the *\$DB/item* file.

This was solved by linking the *\$DB/item* file to a lock file, doing the update, then unlinking the lock file. Unfortunately, this uncovered what may be considered a bug in the *System V UNIX* system used: *ln* has the wonderful property that if the specified target path name already exists, *ln* will first unlink the target path name and then create the specified link. This meant that it was necessary to resort to */etc/link* to do the linking, which meant that the protection on */etc/link* had to be changed from 550 to 555. Why is */etc/link* so dangerous that it is necessary to protect it from general users with a 550 protection mode?

Another general gripe is there doesn't seem to be any utility that will attempt to remove a file, and if it fails, report the failure and exit with a failure status. *Rm* doesn't have this property, it prompts if the file doesn't have write access enabled, and the *-f* option causes it to exit with success status. This was highlighted during the writing of *dbr*.

Metrics

The current **Stashbase** uses 11789 k of disk space, contains 898 items under 738 distinct keywords using 2287 keyword entries giving averages of 3.1 keywords per item of 13.12 k, including indices, created at a rate of 1.55 days per item.

History

Stashbase initially had only keyword indices. This became annoying as often a single item could not be selected by keywords alone. When *dbk* was written, it created the need to specify a particular item, and it was simple to add

an item index, *\$DB/item*, to satisfy this need. The addition of the item index required minimal modification to existing interfaces. The modifications to *dbi* to handle the item index actually made the code simpler.

Most of the items in the database originated from *USENET*, and as such still contain most of their news headers. A significant part of *dbi* was and is to remove extraneous news headers.

The earliest entry in *Stashbase* is a news article from March 1989:

From: ajw@ditmela.oz (Andrew Waugh)
Newsgroups: aus.general
Subject: Public Domain ISO Protocol Stack
Message-ID: <4607@ditmela.oz>
Date: 30 Mar 89 23:49:31 GMT

Problems

Keyword Selection

The major problem with *Stashbase* is keyword selection. When inserting an item it is difficult to select consistent, descriptive and sufficient keywords. In some ways it would be nice if the database itself automatically select keywords describing each item, but this may to have it's own problems:

- algorithm design
- computational load
- index storage requirements

If multiple people were choosing keywords for a shared database then these problems would be exacerbated.

The following heuristic is used in an attempt to solve some of the problems of keyword selection:

- Where possible, keywords are singular, not plural.
- Hyphenated keywords are discouraged.
- Lower case is preferred, even for acronyms.
- Leading dashes are prohibited.
- Slashes are prohibited.

Leading dashes in keywords make it difficult to use the keyword as an initial argument to a program. Allowing a slash in a keyword is an obvious problem as slash is

the UNIX path name separator.

The Garbage Factor

A problem *Stashbase* shares with normal user filing systems is expiry of unwanted items. This could be partially solved by including an optional lifetime when an item is initially inserted into the database.

Inode Exhaustion

A large *Stashbase* uses lots of inodes. A database with 1000 item uses more than 1000 inodes. Where multiple databases are used on a single disk partition *Stashbase* may cause problems, even though most UNIX disk partitions have inodes to spare.

Possible Enhancements

It should be reasonably easy to add an extra non-keyword index, to allow item lifetime to be specified, either by specifying an expiry date or a modification time limit. This, when associated with a script run from the user's personal *crontab* would remove dead items.

Some news articles would be good to regularly save in the database, but would need to have something akin to the news system's *Supersedes:* header to purge unnecessary items from the database.

Documentation in the form of manual entries would probably be a good idea, however, with the largest shell script being 120 lines, *less* makes a good manual command.

Conclusion

Stashbase is simple, easy to use and modify, and fits in well with the rest of the UNIX environment. It provides a useful, if limited, text retrieval function.



Moving to OPEN Systems?

Softway is Australia's largest open systems software house.

We understand the needs of the open systems marketplace, and have extensive expertise in the following areas:

- Client/Server architectures
- TCP/IP based networks
- Security auditing
- Network integration
- Benchmarking and performance tuning
- Software Quality Assurance
- UNIX Training
- Contract software development

For more information contact us by phone on (02) 698 2322, by fax on (02) 699 9174, or by email on enquiries@sw.oz.au

79 Myrtle Street
Chippendale NSW 2008

PO Box 305
Strawberry Hills NSW 2012

Security Tool Review: TCP Wrappers †

by Mark A. Monroe

<monroe@mercator.rtpnc.epa.gov>

[SAGE editor's note: Chistine Quinn of Stanford University has been diligently working to create a regular column to review software tools for systems administrators. I am pleased to present the first of many such reviews. Mark Monroe and Dan Farmer have teamed up to review security tools for SAGE, and their first offering is a good one. If there are tools that you think Christine should try to have reviewed in future columns, please contact her at <quinn@ee-cf.stanford.edu>.]

"May I see some identification, please?"

The security guard asks this question of each person entering the lobby of our building, and the

presence of the guard keeps all but the most determined intruders outside the foyer.

If you imagine that your computer is an office building, the TCP wrapper program becomes a security guard at the entry desk, recording the comings and goings of everyone who accesses the "building." In addition, the wrapper acts like a security gate, checking IDs before allowing entry. Wrappers are not foolproof, but they do go a long way towards closing the door on unwanted visitors.

The wrapper program, add-on software written by Wietse Venema, does not require changes to existing daemons or configuration files. It does not exchange data with the client program, which makes it invisible to the authorized user and

† This is a re-print from *login*, the USENIX Association Newsletter, Volume 18 Number 6

insensitive to bad data from the requestor. It is also independent of application (*telnet*, *rlogin* etc), so the same protection can be used to cover many services. It is available via anonymous *ftp*. (See below.)

How Does It Work?

TCP wrapper checks each request your computer receives for TCP/IP services like *ftp*, *telnet*, Berkeley *r* commands, etc. It compares the source address of incoming requests to a list of hosts that are allowed or denied access to services on your machine. Every request, successful or not, is logged by *syslog* so you can see who is requesting access to your computer.

The concept is simple: *inetd* is configured to call TCP wrapper in the place of the daemon for each service you wrap. This gives the wrapper a chance to check two files, *hosts.allow* and *hosts.deny*, for the hostname or Internet address of the incoming request. The allow/deny files are queried each time a connection is requested, so changes to the files are picked up without a SIGHUP to an existing process.

If the TCP wrapper can match the incoming request with a line in the *hosts.allow* file, the request is logged, the connection is passed to the appropriate daemon program and the session continues as if there were no wrapper. If the wrapper finds a match in the *hosts.deny* file, the request is logged and the connection is dropped. Keywords (e.g., ALL) and wildcards in the allow and deny files make configuration easy. The entry ".edu" refers to all connections from host addresses ending in that string.

TCP wrappers can be configured to compare the address inside the IP packet to the host name that is returned when the remote site is queried with *gethostbyaddr()*. The wrapper disconnects if there is any discrepancy, suspecting that the requestor is trying to spoof your machine.

You can even do counterintelligence of a sort. The allow and deny files can be configured so that the remote site is identified using a reverse *finger* (or any arbitrary command) to try and identify the real user making the request. The RFC 931 protocol is also supported by TCP wrappers.

The README file contained in the TCP wrapper package is excellent and has a more thorough explanation of these and other features.

How Long Does It Take To Install?

I retrieved the *shar* files and had the program built and installed two hours later on my Data General AViiON. Most of the time was used reading the documentation, changing configuration files, etc. The build takes only a couple of minutes. I was able to build and install on a Sun 4c machine in about thirty minutes, start to finish. We tested four other machines and had similar results. The README file lists 12 machines that have been tested successfully, including SunOS, DEC Ultrix and OSF/1, HP-UX, AIX, Apollo, Sony, NeXT, SCO UNIX, DG/UX, and UNICOS on a Cray.

Shortcomings

Version 5.1 of TCP wrappers doesn't support the System V TLI protocol yet. I tried an alpha test version that had been modified to accept TLI connections, and expect to be able to wrap TLI on DG/UX and Solaris very soon.

In the README file, other shortcomings are discussed, including the difficulty in detecting host name and host address spoofing. The current version of TCP wrappers tries to detect host name spoofs by checking the name-address combination from two sources. Wrappers also only check the initial connection request, making it unusable on NFS mount daemons or daemons that linger after the initial session is complete, such as *sprayd*.

Where To Get It

Many sites have copies of the latest version of TCP wrapper. An *archie* search revealed 96 sites with some available version. I retrieved my copy from *ftp.uu.net* in the */usenet/comp.sources.misc/volume36/log_tcp* directory. It's also available on *cert.org* and the author's archive on *ftp.win.tue.nl*. The *patchlevel.h* file should say it is version 5.1 after you apply *patch01*. The three *shar* files and the *patch01* only take up 154KB on your disk when uncompressed, and the final program is only about 25KB on the Sun.

Recommendation

Overall, TCP wrappers are an excellent tool for improving your network security. The logging features alone add a level of integrity that few UNIX vendors provide out of the box, and the ability to screen requests makes it harder to abuse services from outside your domain.

Hierarchies in System Administration

by Wendy Nather

<wendy@il.us.swissbank.com>

How many of these system administration myths do you still hold near and dear to your heart?

- PCs aren't real computers; therefore no one who administers them can be a real system administrator.
- Real system administrators are male; females just aren't technical enough.
- Documentation writing isn't really technical work.
- If you can write device drivers, you should never have to change a toner cartridge again.
- Real system administrators don't have to deal with users directly; that's the Help Desk's job.
- Real system administrators are Hackers. Hackers are brilliant, eccentric hippies or nerds who don't have to have social skills because they're so smart.

The caste system is alive and well in system administration. The Untouchables at the bottom of the heap are the documentation writers, cable pullers, and customer support people who do all the work nobody else wants to do. At the top of the food chain are the old macho hackers who are pampered and indulged as thoroughly as any prima donna.

And if this admittedly sweeping generalization annoys you, stop to think about why it gets to you. Are you still seeing vestiges of this attitude in your own workplace? Does J. Random Hacker still oversleep and miss the group meetings without reprimand, or throw temper tantrums and slam-dunk his beeper into the toilet without reproach because he's the guru of the group? Do your coworkers somehow manage to let backup tapes go unchanged because they were busy writing poetry in Perl? Does all the documentation work end up piled on the desk of the female coworkers because "they're better at that kind of thing," while all the men shrug and say "I'm technical; don't expect me to be able to spell"?

I believe that in the realm of system administration, we cannot afford to drop the "dirty work" to the bottom of the list when defining a Real Sysadmin. A Real System Administrator should be able and willing to document when needed, pull cables with the best of them, explain something to a naive user with tact and diplomacy, manage time and projects responsibly, work cooperatively with others, and above all, treat other technical people as equals instead of perpetuating stereotypes.

For one thing, UNIX administration has opened up considerably in the past few years. Granted, it still requires a higher amount of technical expertise than most nontechnical management realizes, but it is also not the secret society it used to be. Management who didn't understand the inner workings of UNIX were heavily dependent on their gurus and indulged their idiosyncrasies since the technical pickings were very slim. But the workforce has grown larger, tools are making it easier and easier for lesser-experienced people to administer their own systems, and the industry is expecting more and better customer service from its system administrators. We can no longer afford to act as a privileged class, or make class distinctions among ourselves. We must concentrate on giving the best *generalized* service we can — user consulting, design, documentation, training, hands-on work with all other technical people, and an integrated system overview that only technical knowledge can provide. As Heinlein pointed out, specialization is for insects.

And as the old saying goes, if you want something done right, you need to do it yourself. Maybe if the most adept technical people wrote the documentation, we'd have good documentation we wouldn't have to complain about. If hot-shot hackers would devote time to installing and maintaining PC networks, the glamour and mystique would transfer itself to that job and would then be better compensated and appreciated. I have often seen higher-level problems noticed and solved by having a "guru" do the most mundane of jobs. There are very few tasks in system administration that cannot benefit from the expert's touch.

And finally, rather than getting yourself stuck forever doing the dirty work you're afraid to touch, you will increase your own value and marketability if you are willing and competent at every aspect of your job. I have seen several gurus passed over for opportunities they wanted because they were missing the requisite people skills, couldn't be trusted around customers, or had left a bad impression on management by their unprofessional behavior. If you work yourself into such a specialized, rarefied

† This is a re-print from ;login, the USENIX Association Newsletter, Volume 18 Number 6

position that you can't be replaced by someone else, then you also can't be promoted either. It benefits both you and your group to pass on what you know, rather than keep it to yourself to preserve your elite status.

Just as there are different types of intelligence, there are different types of technical expertise. Let's acknowledge and appreciate each sort because they all belong under the umbrella of system administration.

Why System Administrators Are So Short Tempered: Power Switches

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

I just spent 2.5 hours turning off computers. Admittedly, we have over 200 computers (and actually, I think our previous numbers were off, because I know we never counted any Amigas, and I know we turned off two). Furthermore, they are relatively evenly distributed over a floor-and-a-half of a very long building, which at the time was rather poorly lit. (The security guards turn off a lot of lights from the circuit breakers on the weekend.) On the other hand, there were three of us, which averages out to a little under 3 minutes per computer. As our waiting friends and relatives said as we struggled out of the building, weary but unbeaten, "How can it possibly take TWO AND A HALF HOURS to turn computers off????!!!!?"

Well, to start with, no two models of anything have power switches in the same place. Within minutes, you develop switch-finding heuristics, which allow you to find most power switches rapidly while only looking moderately stupid. You grope for the power cord, and feel for a switch or switch-like object in its vicinity. Failing that, you push anything in that general area that feels button-like. (This has the happy side-effect of firmly seating all the fuses.) If the machine is still running, you glance at the front to see if there is a switch or button there. Still no luck? Try the right side to see if there is a big red switch. After this, you're down to feeling under things, cautiously feeling the rest of the back of machine, and then looking very carefully at the whole machine.

It's a good thing that the users aren't generally there to watch when we turn off machines. First, our heuristic mechanism generally takes longer on a machine with a big obvious power switch on the front, since you don't look there until you've tried the back. This looks stupid to bystanders, who can't understand why you don't just look at the machine. Looking at the front takes more effort than you'd expect:

- Nobody makes front-mounted power switches conspicuous on machines that sit in offices, because it would either annoy people or encourage them to push them at inopportune times. (This is usually a good thing; small children with a passion for buttons and switches do occur in offices.)

- Since most machines don't have front-mounted switches anyway, the sole purpose of such a survey would be to keep you from looking silly. It wouldn't work, either: after you turn off all the machines you can find, you stand in the middle of the office and listen intently to see if you can hear anything that's still running. This looks pretty silly any way you slice it, and even sillier when you start searching down the remaining noises.

Second, we don't exude that air of confidence and surety that users like to see. Users like to feel that you know what you're doing, particularly when it involves power, and discovering that you don't in fact know how to do something as basic as turning off the computer shakes their confidence. In point of fact, however, actually flipping the power switches is the only part of shutting down the facility that requires no expertise beyond a thorough acquaintance with the tricks designers use to hide the switch. The main trick is in knowing when you can flip the switch, and that we're very good at indeed.

Third, after 2.5 hours, you develop a sort of casual brutality about the entire enterprise. Can't find the power switch? Shove it around until you can see what you're doing. Still no switch? Turn off the power strip it's attached to. No power strip? Pull the cord off the wall and go on to the next. People think you ought to take the whole thing more seriously and slowly.

It's another one of these enterprises that causes you to develop firm and passionately-held opinions about things that other people regard as fundamentally trivial. For instance: power switches belong next to power cords. Power switches never, ever belong *under* things, even overhangs you can reach under easily. Power switches should be *switches*, with clearly marked on and off positions. Toggles are barely acceptable; rocker switches and flat plates that otherwise work like switches are unacceptable. Buttons are completely, totally unacceptable. Putting little diagrams next to them to show which position is on and which is off does not make them acceptable, particularly if on and off are distinguished by how far they stick up (How can you tell which picture matches the current state of the button?)

Slide switches are out. Knobs are beneath contempt, and a special circle in hell is reserved for people who design equipment which has a power switch but can actually only be turned off from software.

Furthermore, power switches should be the only switches next to the power cord. The Diag/Norm switch on Sun 3 machines is a cruel trick, particularly since it's much easier to find than the real power switch, and does nothing perceptible until the machine is rebooted, or not rebooted, as the case usually is. Any given piece of equipment should have only one power switch; putting one on the front that turns off most things, and then putting another on the back that turns off the power supply itself is not fair at all. Hiding power switches behind panels is not fair unless the panel is clearly marked as where all the switches are. Combining the two can only be ascribed to sadism. We had a tape drive at one point with a "Logic Power On" button on the front panel with all the other buttons, and the real power switch hidden behind the door you open to load the tape, all by itself. One of our hardware people disassembled the drive, figuring that it was broken, because the switch on the front panel was on but it wasn't getting any power. (Admittedly, this was the same drive that displayed error conditions in binary from right to left using the status lights, where blinking was one and solid on or off was zero, so it probably was designed by a sadist.)

For other reasons, I dislike lighted power switches — they're great when the little lights work, but what do you do when the light in the power switch burns out? To avoid this, our power

strips use some neon-like bulb, which I assume doesn't burn out often. Instead, it flickers, thereby convincing the unwary that either the power or the power strip is intermittent. On the other hand, it is nice to be able to tell whether the object is turned on or not, without feeling for warmth or vibration. People look at you funny when you put your hand on pieces of equipment and then concentrate intently as if trying to commune with the machine. You know that you're trying to figure out whether it's vibrating because of its own fan, or whether it's transmitted from something else, or if it's not vibrating perhaps it's heating up, and if so is that because its fan is broken, and it's going to catch fire, or because it isn't supposed to have a fan and is just supposed to radiate heat gently, or maybe because the piece of equipment nearest it is blowing hot air on it and it's really turned off.

A few vendors have decided to remove the whole power switch concept by simply not providing the power switch, another unfortunate decision. The rumor says that in the initial design of the Kinetics FastPath (yes, when it really was Kinetics), they analyzed the mean time between failures of all the parts, and the ones with the shortest times were the power switch and the power lights, so they designed it out. They declined to recalculate the mean time between failures at this point, or they would have discovered that the power cord was now the most likely to fail, with a lower mean time between failures than the power switch had originally. This may all be fiction, but the fact was that the next hardware release had a power switch (and a power light), much to the relief of their customers.

Perl 5.0 Overview †

by Tom Christiansen
<tchrist@cs.colorado.edu>

The last major release of Larry Wall's freely redistributable Perl programming language was three years ago. Since then, Larry's released a few updates, but these have been almost entirely bug fixes and portability enhancements: no significant new functionality was introduced. During this time, Perl has spread from hard-core UNIX shops into regions never dreamt of by its author, from trading firms to chip designers to heavy industry.

Although its origins are firmly rooted in UNIX, Perl now runs more or less happily on operating systems as diverse as MS-DOS, Windows/NT, VMS, the Apple Macintosh, and many others. It has become the language of choice for harried systems administrators who haven't the time for slow, awkward, and unportable shell scripts, nor for inscrutable, tedious, and unportable C programs. Increasing numbers of companies are now shipping Perl and using it internally for their install programs, test suites, database interfaces, and end-user applications.

With its 5.0 release, Perl promises to address an even broader range of systems and applications programs than ever before. For certain applications, programmers will still labor long and painful hours writing in C so that they might squeeze that last bit of efficiency. Still, many of us find that a good interpreted language is plenty fast enough, especially on today's blazing hardware, which can only get faster quickly.

The new features in Perl 5.0 will not break existing code except in a few extremely rare cases where the writer was being, to quote one of C's fathers, unreasonably chummy with the compiler. This is because the internals have changed drastically, yielding a leaner, cleaner, faster, and more predictable program. The grammar is much smaller and simpler, and the language is more forgiving of mistakes and more informative when you do something questionable, instead of silently doing something you might not expect.

For example, you no longer have to remember which things must have, might have, or can't have parentheses, although the cautious programmer will continue to use them in all circumstances to make life easier for software

maintainers. Even better, most of the surprise factors causing difficulty for new users, such as context sensitivity, now generate warnings (under the *-w* switch) when you're doing something that doesn't really make much sense, like multiplying two non-numeric strings together, using a list when you want a single element, and so forth. Running perl without *-w* would be like running classic C without using *lint*.

The single most important area which Perl now addresses is that of nested data structures and references. A given element of a list (linear array) or table (associative array) can itself be a reference to another list or table. This allows you to construct all the complex data types which you were longing for, such as binary (or n-way) trees, C-style structures and jump tables, or lists of lists of lists. For example, here's a list containing sublists:

```
[2, 4, [2, 9], 8,  
 [8, 12, [2, 5, 0], 9], 8]
```

And here's a table containing other tables. (The *=>* is just a glorified comma that visually distinguishes when you're constructing tables.)

```
{  
    RED => { CRIMSON => 1, SCARLET => 2 },  
    BLUE => { AZURE => 1, INDIGO => 2 },  
}
```

Furthermore, these can be mixed and matched; here's a table, indexed by someone's name, whose values are each lists of names:

```
{  
    John => [ Mary, Pat, Blanch ]  
    Paul => [ Sally, Jill, Jane ]  
    Mark => [ Ann, Bob, Dawn ]  
}
```

We now have references and referencing and dereferencing operators to go with them. References in Perl are type-safe and type-checked, unlike C's pointers. Furthermore, variables have reference counts on them to control when storage is released.

For postfix dereferencing, you have C's *->* operator, allowing you to write things like

```
$r -> [3]                                # list element  
$r -> [3] -> [4] -> [17] = 3; # 3-dim array  
$r -> {"John"}                            # table element  
$r -> {"ru_utime"} -> {"tv_sec"} # nested table
```

The last demonstrates the most straightforward way to represent C-style structures. More elaborate methods are also possible. Multiple level constructs need not be declared — each level is created on the fly if needed, just as assigning

† This is a re-print from *login*, the USENIX Association Newsletter, Volume 18 Number 6

something to a scalar variable makes it spring into existence without any previously declaration.

To create a reference to a named variable, we use a backslash where C used an ampersand:

```
$sref = \$some_var;
$1ref = \@some_list;
$trref = \%some_table;
$frref = \&some_func;
```

For prefix dereferencing for which C uses a *, you can use any of the four previous type specifiers: \$, @, %, or &. You can use more than one of them, in fact, because you could have refs to refs, or functions returning refs.

```
$$sref = 3;           # assign 3 to $some_var
pop(@$1ref);        # pop @some_list
@key_list = keys(%$trref); # keys \%some_table
&$frref($parms);    # some_func()
```

As in the shells, you may always use braces to clarify:

```
pop( @{ $1ref } );
&{ $jump_table{$name} }($keystroke);
${${$refref}} = 1;      # like $$refref = 1
```

Finally, there's a ref() built-in function that returns the type of reference you've got, allowing you to write functions to print out nested data structures without knowing quite what's in them beforehand. Possible return values from the ref() are "" if it's not a reference, or SCALAR, ARRAY, HASH, CODE, or REF.

```
if (ref($r) eq "HASH") {
    dump_hashtable()
}
```

That's right: you don't need to use the ampersand on a function call anymore if you don't want to.

And yes, you may have user-defined data types as well! Using Perl's package system, you can write code using object-oriented programming strategies. Per-class and per-instance user-defined constructors and destructors are supported (reference counts are essential for knowing when to call the destructor), as are multiple inheritance and class-specific functions and data. For example:

```
$r->next_seq()
```

would call the next_seq() method of whatever class (package) to which \$r belongs (next_seq() could get at \$r, it's "this" pointer, in C++ parlance). If there isn't such a method, then a runtime search up the inheritance chain will ensue. If none is found, then an exception would be raised using Perl's exception handling mechanism. If one is found, then it's cached so that the search

can be shorter the next time.

Here's another interesting possibility. These two are identical, making it slightly nicer to read and write certain constructs:

```
$new_ob = $old_ob->new();
$new_ob = new $old_ob;

$count = $ob->sizeof();
$count = sizeof $ob;
```

Nifty, eh? Those are user-defined methods, not built-ins. This isn't just gratuitous syntactic sugar: it falls out of the "indirect object" slot such as is found in output statements, sort routines, etc. It's really just like:

```
print $fh "string\n";
```

The other major set of functionality that will greatly aid programmers is in the area of scoping. Perl now supports both static and dynamic scoping of variables. Originally only dynamic scoping was supported, but so many C and Pascal programmers were unused to it that they found themselves making strange mistakes.

It used to be that variables always sprang into existence when you first used them whether you wanted them to or not. Now, you may optionally enforce variable "declarations" on a per-block basis. For such blocks, all variable references must be either a statically-scoped local or a fully-qualified global. Any stray variable references will be flagged at compile time. This makes it much easier to write safe code that doesn't accidentally alter or create a global variable.

There's quite a bit more we don't have time to go over in detail. Here's a partial list of what's already been completed that we haven't gone over already in this article:

- Nestability of quoted strings
- Improved exception mechanism
- Support for BEGIN and END subs on a per-package basis
- Various bug fixes

And here's a list of some of what's anticipated to be there, but not strictly guaranteed:

- Embeddability into C and C++: cc prog.c -lperl
- File handle objects: \$STDOUT->flush(1)
- Separate man pages for all library functions and built-ins
- Very easy GUI Perl applications using high-

- level X bindings
- Many more libraries, including *class* and *struct* libs
- More example code in the *eg/* directory
- A Perl profiler
- Debugger enhancements
- Access to POSIX 1003.1 functions
- Mnemonics for all the "funny" variables, (e.g.,

`$ERRNO for $!`

- Easy extensibility using C functions
- Various Perl development tools

If you'd like to play with an alpha release of the Perl 5.0 release, you can retrieve it from `ftp. netlabs. com [192.94.48.152]` in `pub/outgoing/perl5.0/perl5a3. tar.Z`. It already contains all the functionality detailed in the long exposition above. It also has interesting files with more information: check out *Changes*, *Todo*, and *Wishlist*.

Automatic Information Server †

The Association has a mail server that will answer requests received through electronic mail. Information is available about USENIX and SAGE membership, conferences, and publications.

Please send all requests to: <info@usenix.org>

To receive the *Primary Services Catalog*, the body of your mail message should contain only the line:

```
send catalog
```

This catalog outlines accessible information about the USENIX Association's services. You can get more detailed listings for subtopics in each service category by sending the line:

```
send service catalog
```

where *service* is one of the primary topics listed below. Thus:

```
send conferences catalog
```

will result in your receiving a listing of available announcements for upcoming USENIX Conferences and Symposia.

All files are in plain text format. The following subtopics exist currently:

Service: conferences

Description: The *conferences* catalog contains all the current announcements for USENIX Conferences and Symposia. This includes all available calls-for-papers and registration information (technical sessions, tutorials, fees, and accommodations) for upcoming events.

To get the *conferences* catalog:

```
send conferences catalog
```

Service: publications

Description: The *publications* catalog contains ordering information for our conference proceedings, back issues of our quarterly technical journal, *Computing Systems*, as well as publications, announcements, and calls-for-papers for special issues of *Computing Systems*.

To get the *publications* catalog:

```
send publications catalog
```

Service: membership

Description: The *membership* catalog contains information about the USENIX Association, the System Administrators' Guild (SAGE), and an application form.

To get the *membership* catalog:

```
send membership catalog
```

Service: papers

Description: The *papers* file contains information about submitting papers to upcoming USENIX conferences and symposia, including a sample abstract and advice on how to write a good paper.

To get the *papers* catalog:

```
send papers catalog
```

Questions, comments, suggestions: please contact <info-admin@usenix.org>.

† This is a re-print from ;login, the USENIX Association Newsletter, Volume 18 Number 6

An Update on UNIX-related Standards Activities †

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

IEEE Standards Board

Mary Lynne Nielsen *<m.nielsen@ieee.org>* reports on the June, 1993 meeting in Montreal, Canada:

This time around, the IEEE Standards Board meeting was held in Montreal, Canada. Not only is it the home town of the current vice-president of the Standards Board, Wally Read, but also having a meeting in this location continues a tradition of trying to hold IEEE Standards Board meetings outside of the United States and the East Coast in particular.

Electronic PARs

NesCom, the New Standards Committee and the group in charge of reviewing Project Authorization Requests (PARs) for the Board, has slowly begun to nudge itself into the electronic arena by working on the development of an electronic PAR form. There has been a call for this for some years now. As such, Jim Isaak and Steve Diamond, two members of NesCom who represent the Computer Society, have been working with IEEE staff to develop this. A first draft was submitted for review and comment at this meeting. With any luck, standards developers will be able to submit PARs electronically sometime in 1994; some sticky issues, such as having real signatures on the form, are the reasons for the time delay.

Other NesCom news involved the approval of four new PARs to expand and rearrange the 1238 work (which also involved the earlier withdrawal of one 1238 PAR). Another PAR, 1003.11, was also withdrawn. This action was the result from an initial ballot on this standards profile on transaction processing, which demonstrated that the work was premature. As such, PASC asked that the PAR be withdrawn, and NesCom approved that request at this meeting. Details of all of this are listed below.

IntCom JTC1 Guide

As mentioned in earlier reports, the IEEE Standards Board International Committee (IntCom) recently approved a document to help IEEE working groups coordinate their standards development with ISO/IEC JTC1 (the international committee in charge of developing infor-

mation technology standards) if working groups wanted to move their standards into the international arena. This guide was disseminated to a wide mailing list, and some concerns were raised about how the role of national bodies was represented. Because of this, IntCom agreed to prepare a revision to address this issue, and a revised guide will be prepared for the fall.

ProCom Actions

The Process Committee, ProCom, approved some new language to identify more clearly the role of organizational representatives, and this will be included in the ballot of new material for the *IEEE Standards Board Bylaws* and *IEEE Standards Operations Manual* for next year.

ProCom also had a long discussion about the merits of company membership and whether the IEEE should be examining this as a possibility. As most of you know, the IEEE is an individual membership organization. However, some groups have approached the IEEE about using its system of standards development, only to be turned down because they wanted to have company members in addition to individual members. (Remember, this is distinct from organizational representatives, which cannot be individual companies.) This, in turn, has led to some discussion over whether some working groups should be allowed to have company membership in their balloting, and if that should be restricted to all company membership, partial company membership, or some other mixture.

Needless to say, ProCom felt it had to gain a great deal of input before it could make a decision on this matter. As such, they are soliciting opinions and white papers on this subject. If you are interested, you can relay your concerns to the IEEE Standards Department, which will pass them on to ProCom. Some of the material presented to ProCom is also available (such as a summary of an earlier IEEE forum on this subject).

New Horizons

Finally, the IEEE heard reports on two areas of potential growth. One has been in the works for a couple of years now, the IEEE SPAsystem—the electronic bulletin board, delivery, and standards development system. At the June IEEE Board of Directors meeting (this is the board that runs *everything* in the IEEE, and that oversees the IEEE Standards Board, among other things), a loan

† This is a re-print from ;login, the USENIX Association Newsletter, Volume 18 Number 6

was granted to the IEEE Standards Department to support the implementation of the SPAsystem. This will go a long way towards helping get this large and ambitious project off the ground.

In addition, the IEEE Standards Board approved work on developing pilot programs in the area of conformity assessment. Unlike the SPAsystem, this is a brand-new area of work for the IEEE, which will hopefully increase the visibility and build the value of IEEE standards while also allowing easier entry of products that conform to these standards into international markets. Work is currently underway to pinpoint the pilot projects and to develop these programs further.

One other minor but interesting point: this was the first meeting in over a year in which PASC did not have at least one project approved at RevCom!

New PARs

P1351: Draft Standard for Information Technology—OSE Application Program Interfaces (APIs)—ACSE and Presentation Layer Application Program Interface [Language Independent]

P1352: Draft Standard for Information Technology—Test Methods for OSE Application Program Interfaces (APIs)—ACSE and Presentation Layer Application Program Interface [Language Independent]

P1353: Draft Standard for Information Technology—OSE Application Program Interfaces (APIs) C Language Interfaces-Binding for ACSE and Presentation Layer Application Program Interface

P1354: Draft Standard for Information Technology—Test Methods for OSE Application Program Interfaces (APIs) C Language Interfaces-Binding for ACSE and Presentation Layer Application Program Interface

Withdrawn PAR

P1003.11: POSIX based Transaction Processing Applications Environment Profile

Report on POSIX.0: Guide to OSE

Kevin Lewis <klewis@gucci.enet.dec.com> reports on the October 18 - 22, 1993 meeting in Bethesda, MD:

This was a week of shifting expectations. We were hoping to start resolution of the recirculation ballots, but, since the deadline was set after the ending of the meeting (and, by the way, was subsequently extended to November 8), the group shifted its attention to two other questions: "Where do we go from here, if anywhere?" and "How can we be sure that this document moves smoothly through the international standards community?"

We spent a fair amount of time talking about where we might go from here. Member of two groups presented their ideas in the form of sample PARs. One discussed ODP and the other focused on profiling from the user perspective. If you want my opinion (and you must if you're reading this), ODP seems to be the way that many think we should go. That doesn't mean that other work would not also be pursued. The problem is that old classic conflict of too big a plate for too few people, particularly given the state of the industry and the reduced number of participants.

One member was quite vociferous about having the group pursue ODP. The group's consensus was that it is not ready to pursue any other work as a group until the current effort is close to completion. But that doesn't mean that someone can't go off and champion such an effort alone, or with the help of others, and my belief is that someone may do this. I must also admit that the current reduction in participants along with the possibility of POSIX, or rather the Portable Applications Standards Committee (PASC), being restructured does, in my view, have a psychological effect on how new work might be accomplished. As you can see, there are many unanswered questions.

As for the question about getting the guide through the international standards community, the first hiccup occurred when SC22 omitted the POSIX.0 guide from its list of documents to be forwarded for Committee Document (CD) registration and letter ballot. Most of what I heard about this was in the halls and over lunch. It appears that SC22 did not like what it perceived to be a change in scope in Draft 16, the recirculation draft. The WG15 Technical Advisory Group (TAG) agreed to back up a step and forward draft 15 for SC22 CD registration and letter ballot. Yes, we are regressing ...) but, the plan, or should I say hope, is that all concerns raised by SC22 will be addressed during resolution of the recirculation ballots. I think they will be addressed ultimately.

The plan is to come into the January meeting at Irvine prepared to commence the recirculation resolution. I will be sending out ballots to the section leaders as I receive them so they can get started as soon as possible.

Now, for that \$64 question: when do I think POSIX.0 will be done? I have been burned more than once trying to answer that. But I will try again: look for a completion, i.e. submittal to the IEEE standards board for approval, sometime during late summer or early fall of 1994.

Unix Tricks & Traps

Working in two places at once

I usually work at an X terminal on my desk in the Centre for Water Research (CWR), at UWA. Until recently, I also liked to run a few X clients on one of the systems at my old site, the Department of Computer Science, and have them display on my X terminal automatically, as part of my usual working environment. You may find it useful to adapt my automation to your own needs.

The following script is meant to be sourced by a C-shell running at Computer Science. Once this script is sourced, the shell can then run X clients and they will be displayed on my X terminal back at CWR. The script sets up the DISPLAY variable and the necessary magic-cookie authentication.

```
#!/bin/csh
# cwrdis
# Source this file to set up environment on CWR display
#
setenv DISPLAY xterm.cwr.uwa.edu.au:0
set KEY='rsh xdmhost.cwr.uwa.edu.au -l jackson \
          xauth list $DISPLAY | cut -c28-
xauth <<End
remove $DISPLAY
add $DISPLAY MIT-MAGIC-COOKIE-1 $KEY
exit
End
unset KEY
```

Things to note about *cwrdis*:

- *xterm* and *xdmhost* aren't the real hostnames.
- *xterm*'s display is managed by *xdm* running on *xdmhost*.
- The CWR hostnames are hardcoded in this script.
- For this to work, my .rhosts at CWR has to allow access from my account on the Computer Science host.
- Matters are complicated because my account name at CWR is *jackson*, whereas at Computer Science it is *janet*.

So now I can telnet or rlogin to the Computer Science host, source *cwrdis*, and run my Computer Science X clients. The next step is a script to do that, running a standard set of clients.

```
#!/bin/csh -f
# cwrrun
# Run usual commands on CWR display
#
source ${HOME}/bin/cwrdis
# Now run whatever clients you want, eg xterm & xpostit
set hostname='hostname'
xterm -xrm "*iconGeometry: 62x17-85+240" \
      -iconic -cu -j -ls -rw -sb \
      -geometry =80x24+100+100 -font 10x20 \
      -T $hostname -n $hostname &
xpostit -geometry 105x64-92+1 &
```

Now I can achieve my ends by simply typing *cwrrun* from a Computer Science shell. But this is still not good enough: I want to be able to do it without first having to remotely log in to the Computer Science host. For this I created a C-shell alias (in *.cshrc* or *.login* in my CWR home directory):

```
alias csrun "rsh -l janet -n cshost.cs.uwa.edu.au \
             'cwrrun </dev/null >&/dev/null &'"
```

(*cshost* isn't a real hostname.)

This only works if my .rhosts on *cshost* allows access from my account on *xdmhost*.

The I/O redirections and the -n option to rsh allow the remote command to be backgrounded: I don't want to wait around for the Computer Science X clients to run before I get my commandline back.

I can now type csrun from a shell running on *xdmhost* in a window on my X terminal. All that remains is to put the same command in my X startup file (.xsession in my CWR home directory):

```
rsh -l janet -n cshost.cs.uwa.edu.au \
    "cwrrun </dev/null >&/dev/null &" &
```

and I now get my Computer Science X clients onscreen each morning without any manual intervention. By keeping the csrun alias, I am still able to run them manually if I need to.

Perl to the rescue again

One of my users recently asked if I could help him fix an ASCII data file. It had been munged somehow, and he wanted to get it back to its proper format for processing. The problems were:

- The file was supposed to contain columns of data, with a "D" beginning each line. However, it had been joined together into one long line.
- Strings of null characters had replaced the spaces between some of the data fields.
- The file (and hence, the line) was about 2Mb long.

A 2Mb line is beyond the capacity of sed, but if you've got the memory, Perl doesn't bat an eyelid.

The following script does more than was needed: it contains a loop in case there is actually more than one line.

```
#!/usr/bin/perl
# fixfile -- one-off script to fix someone's file
# Janet Jackson <jackson@cwr.uwa.edu.au>, 1994-01-14
#
# Usage: fixfile <oldfile >newfile

while (<>)      # for each line in the file
{
    # Get rid of the NULL characters,
    # replacing each string of them with 2 spaces
    #
    s/\000\000*/  /g;

    # Insert a newline before each 'D' except the first.
    #
    s/(.)D/\1\
D/g;

    # print the line, new newlines and all
    #
    print;
}
```

Unfortunately I didn't time it. It took a noticeable length of time to run on a SPARCstation 2 — but it only had to be done once.

Janet Jackson <jackson@cwr.uwa.edu.au>, (09) 380 2408

PS If you have a Unix trick or trap to share via this column, please send it to my email address, or if you're not on email, phone me to make other arrangements.

AUUG MANAGEMENT COMMITTEE

SUMMARY OF MINUTES OF MEETING 10th December 1993

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

Apologies: Greg Birnie, Peter Wishart.

Guests: Liz Fraumann, Wael Foda, Lachie Hill, Piers Lauder.

1. New Member

Welcome to Rick Stevenson as a new Management Committee member, and thanks to Chris Schoettle for his contributions.

2. President's Report

Received a letter from Cliff Stoll - he is happy to come back. EF to put answer to Morris' question to Stoll in AUUGN.

It was agreed we would seek close relationships with the following organisations for support/endorsement for the 94 conference and other projects:

AIIA - Open Systems Group

Mike Duffy, ATO, heads Govt IESC body.

3. Secretary's Report

Secretary not present.

Framemaker distribution of AUUG93 proceedings. There are issues of copyright of proceedings and licence ambiguity and legality. EF will clarify copyright issue and will endeavour to get separate bundling of material from viewer software. More open format. AUUG covering letter to be included.

4. Treasurer's Report

Final AUUG'93 payment yet to come - \$20K recently.

\$5K in chequeing account, as of meeting date. \$63K in Cash Management, and \$36K in Term Deposit. AARNet bill for \$25K due January. Net worth \$100K.

Final AUUG'93 figures are done income - \$251,820, expenses - ?, profit \$27-30K including exhibition. Still some bookkeeping to be resolved.

5. AUUG94

IH has "Call for Papers" in draft form. Issues: speaker conference package? Resolved:

- (1) only for general speakers and technical, management streams.
- (2) WIP, Panelists - single day with option to top up.
- (3) tutorials - full conference package or 25% of net.

It was resolved to form a conference committee of: EF, PMc, FC, IH, WF et al.

Possible speakers: Novell, X/Open, Microsoft (Cutler), Tanenbaum, PJ Plauger, Ousterhout.

Co-conferences:

- (1) SCO developers: they should make proposal.
- (2) PICK group: submit papers to CFP for referee. May provide session if enough papers.
- (3) Supercomputer: seek proposal from them and may run a separate stream. Separate CFP.

Resolved: that the conference committee to prepare a conference budget seeking \$20K positive - for next committee meeting. Noted: that \$450 registration be considered maximum.

Suggestion:

\$350	early bird	members only < 31/5/94
\$425	normal	members only > 1/6/94
\$525	late	members only > 1/8/94
+\$100	for non-members	

6. Other Business

6.1. Australian Articles

Only missed one week but there have been no articles since the last meeting. Trouble with articles, improvements are needed. PMc to do one. Possible topics: Unix in India, Free Software issues.

6.2. Summer conference - McKusick Roadshow

Pricing yet to be determined. P.R. program underway. Diary dates have gone out. Discussions on what chapters get out of AUUG:

- (1) Paying for Kirk's travel.
- (2) Promotion - PR (LH)
- (3) Insurance policies
- (4) Financial underwriting
- (5) Mailings
- (6) Secretarial administration (EF)
- (7) Accounting facilities (Purchase Orders, Credit Cards)

6.3. NZ Conference

EF has secured a bulk travel programme for discounts. Will promote in next AUUGN issue:

A\$668 SYD-AKL-SYD Rotorua May 18-21

6.4. Discount Agreements

Discount agreements have been renewed. Netcomm ,ComTech, Prentice Hall, Connect.com, MHS, DIALix. Woodslane, Dymocks is forthcoming after the new year.

6.5. FTP Server

MP, FC to investigate provision of a 2GB disk to some convenient ftp archive site for AUUG faces, software, reviews, proceedings, AUUGN etc. For approval next meeting.

6.6. Public Liability Insurance

Public Liability insurance - for all AUUG events (including chapters). MMI \$5M, \$500 excess for \$1,150 p/a. Resolved to proceed with policy.

6.7. Membership Drive

Target areas: Student members, Linux user groups (encourage Linux speakers at AUUG), AIIA members, MIS 3000, AUUG '93 Exhibitors , ... GH will contact chapters to offer support for membership drive based on committee ideas.

7. Next Meeting

The next meeting will be on Fri 11th February 1994 at MTIA in Melbourne (date and venue to be confirmed).

Peter Wishart
AUUG Inc - Secretary

AUUG Membership Categories

Once again a reminder for all "members" of AUUG to check that you are, in fact, a member, and that you still will be for the next two months.

There are 4 membership types, plus a newsletter subscription, any of which might be just right for you.

The membership categories are:

Institutional Member
Ordinary Member
Student Member
Honorary Life Member

Institutional memberships are primarily intended for university departments, companies, etc. This is a voting membership (one vote), which receives two copies of the newsletter. Institutional members can also delegate 2 representatives to attend AUUG meetings at members rates. AUUG is also keeping track of the licence status of institutional members. If, at some future date, we are able to offer a software tape distribution service, this would be available only to institutional members, whose relevant licences can be verified.

If your institution is not an institutional member, isn't it about time it became one?

Ordinary memberships are for individuals. This is also a voting membership (one vote), which receives a single copy of the newsletter. A primary difference from Institutional Membership is that the benefits of Ordinary Membership apply to the named member only. That is, only the member can obtain discounts an attendance at AUUG meetings, etc. Sending a representative isn't permitted.

Are you an AUUG member?

Student Memberships are for full time students at recognised academic institutions. This is a non voting membership which receives a single copy of the newsletter. Otherwise the benefits are as for Ordinary Members.

Honorary Life Membership is not a membership you can apply for, you must be elected to it. What's more, you must have been a member for at least 5 years before being elected.

It's also possible to subscribe to the newsletter without being an AUUG member. This saves you nothing financially, that is, the subscription price is greater than the membership dues. However, it might be appropriate for libraries, etc, which simply want copies of AUUGN to help fill their shelves, and have no actual interest in the contents, or the association.

Subscriptions are also available to members who have a need for more copies of AUUGN than their membership provides.

To find out your membership type, examine your membership card or the mailing label of this AUUGN. Both of these contain information about your current membership status. The first letter is your membership type code, M for regular members, S for students, and I for institutions, or R for newsletter subscription. Membership falls due in January or July, as appropriate. You will be invoiced prior to the expiry of your membership.

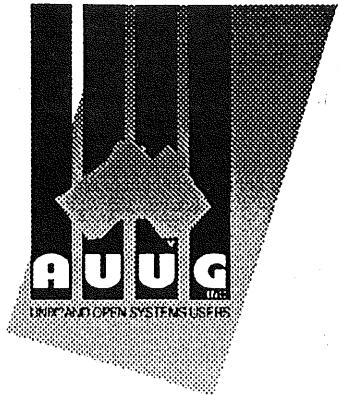
Check that your membership isn't about to expire and always keep your address up-to-date. Ask your colleagues if they received this issue of AUUGN, tell them that if not, it probably means that their membership has lapsed, or perhaps, they were never a member at all! Feel free to copy the membership forms, give one to everyone that you know.

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

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

MEMBERSHIP APPLICATION

INDIVIDUAL



To apply for AUUG membership, complete this form and return it with payment in Australian Dollars to:

**REPLY PAID 66, AUUG MEMBERSHIP SECRETARY,
P.O. BOX 366, KENSINGTON, NSW 2033, AUSTRALIA
Tel: +61 2 361-5994 Fax: +61 2 332-4066**

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

NOTE: Please do not send purchase orders - perhaps your purchasing department will consider this form to be an invoice. Foreign applicants please send a bank draft drawn on an Australian bank, and remember to select either surface or air mail.

INDIVIDUAL OR STUDENT MEMBERSHIP:

I, _____ do hereby apply for:

Renewal/New membership of AUUG

\$ 90.00

Renewal/New Student membership

\$ 25.00 (please complete certification portion)

International surface mail

\$ 20.00

International air mail

\$ 60.00

UniForum affiliate membership

\$ 20.00

TOTAL REMITTED:

AUD\$ _____ (Cheque, or money order, or credit card)

LOCAL CHAPTER DESIGNATE:

You can participate in the activities of a local AUUG Chapter. Part of your fee will be given to the chapter to support those activities. By default a regional chapter will be selected for you. If you would rather nominate a chapter, please specify here _____ (indicate NONE for no chapter).

TO BETTER SERVE YOU, PLEASE PRINT YOUR CONTACT INFORMATION:

Name/Contact: _____

Position>Title: _____

Company: _____

Address: _____ Postcode _____

Tel: BH _____ AH _____

Fax: BH _____ AH _____

email address: _____

Over for Institutional Membership

STUDENT MEMBER CERTIFICATION: (to be completed by a member of the academic staff)

I, _____ certify that

(administrator)

is a full time student at

(name)

and is expected to

(institution)

graduate approximately _____

(date)

Please charge \$ _____ to my

Bankcard, Visa, Mastercard

Account number: _____

Expiry date: _____

Name on card: _____

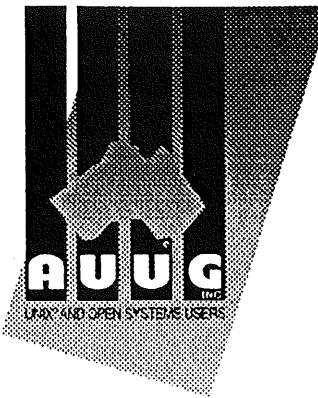
Signature: _____

AUUG Secretariat Use

Chq: bank _____	bsb _____
a/c _____	# _____
Date: _____	\$ _____
Initial: _____	
Date processed: _____	
Membership # _____	

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

MEMBERSHIP APPLICATION INSTITUTIONAL



To apply for AUUG membership, complete this form and return it with payment in Australian Dollars to:

**REPLY PAID 66, AUUG MEMBERSHIP SECRETARY,
P.O. BOX 366, KENSINGTON, NSW 2033, AUSTRALIA
Tel: +61 2 361-5994 Fax: +61 2 332-4066**

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

NOTE: Please do not send purchase orders - perhaps your purchasing department will consider this form to be an invoice. Foreign applicants please send a bank draft drawn on an Australian bank, and remember to select either surface or air mail.

We, _____
(Company Name)
do hereby apply for:
Renewal/New* Inst. membership of AUUG \$350.00
International surface mail \$ 40.00
International air mail \$120.00

TOTAL REMITTED AUD\$ _____
(Cheque, money order, or credit card)

I/We agree that this membership will be subject to the rules and by-laws of AUUG as in force from time to time, and that this membership will run from time of joining/renewal until the end of the calendar or financial year.

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

Signed _____ Date _____
Title _____

INSTITUTIONAL MEMBER DETAILS:

To be completed by institutional members only.

Following are our specified contacts. The primary contact holds the full member voting rights. The two designated reps will also be given membership rates to AUUG activities including chapter activities. By default a regional chapter will be selected for you. If you would rather nominate a chapter, please specify in space provided (indicate NONE for no chapter). (Please print clearly or type)

Primary Contact _____
Position/Title _____
Address _____ Postcode _____
Bus. Tel: _____ Bus. Fax: _____
e-mail address _____
Local Chapter Pref. _____

1st Rep. _____

Position/Title _____

Address _____

Bus. Tel: _____ Bus. Fax: _____

e-mail Address _____

Local Chapter Pref. _____

2nd Rep. _____

Position/Title _____

Address _____

Bus. Tel: _____ Bus. Fax: _____

e-mail address _____

Local Chapter Pref. _____

Please charge \$ _____ to my

Bankcard, Visa, Mastercard

Account number: _____

Expiry date: _____

Name on card: _____

Signature: _____

AUUG Secretariat Use

Chq: bank _____	bsb _____
a/c _____	# _____
Date: _____	\$ _____
Initial: _____	
Date processed: _____	
Membership # _____	

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

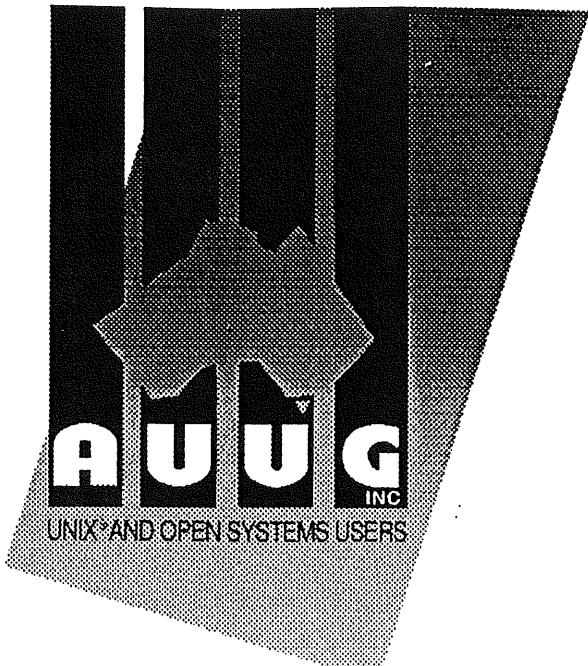
Notification of Change

You can help us! If you have changed your mailing address, phone, title, or any other contact information, please keep us updated. Complete the following information and either fax it to the AUUG Membership Secretary on (02) 361-5994 or post 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..)

- The following changes are for my personal details, member #: _____
- The following changes are for our Institutional Member, primary contact.
- The following changes are for our Institutional Member, representative 1.
- The following changes are for our Institutional Member, representative 2.



PLEASE PRINT YOUR OLD CONTACT INFORMATION (OR ATTACH A MAILING LABEL):	
Name/Contact:	_____
Position/Title:	_____
Company:	_____
Address:	_____
Tel:	BH _____ AH _____ Postcode _____
Fax:	BH _____ AH _____
email address: _____	

PLEASE PRINT YOUR NEW CONTACT INFORMATION:	
Name/Contact:	_____
Position/Title:	_____
Company:	_____
Address:	_____
Tel:	BH _____ AH _____ Postcode _____
Fax:	BH _____ AH _____
email address: _____	

AUUG Secretariat Use

Date:	_____
Initial:	_____
Date processed:	_____
Membership #	_____



AUUG Incorporated

Application for Newsletter Subscription

AUUG Inc.

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

AUUG Membership Secretary
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, 1994

Please *enter / renew* my subscription for the Australian UNIX systems User Group Newsletter, as follows:

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

..... Net Address:

.....
.....
.....

*Write "Unchanged" if address has
not altered and this is a renewal.*

For each copy requested, I enclose:

<input type="checkbox"/> Subscription to AUUGN	\$ 90.00
<input type="checkbox"/> International Surface Mail	\$ 20.00
<input type="checkbox"/> International Air Mail	\$ 60.00

Copies requested (to above address) _____

Total remitted

AUD\$ _____

(cheque, money order, credit card)

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

Please charge \$ ____ to my Bankcard Visa Mastercard.

Account number: _____ Expiry date: ____ / ____ .

Name on card: _____ Signed: _____

Office use only:

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

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

Who: _____ Subscr# _____

AUUGN Volume 14 Index – Papers (Sorted by title)

McRae, Andrew, "386BSD: A Look Under The Hood," *AUUGN*, vol. 14, no. 2, pp. 46-53, AUUG Inc., Sydney, NSW, April 1993.

386BSD is a freely available port of the BSD Networking Release 2 software to a 386 PC architecture. It is fully functional kernel with associated user programs and tools, albeit not yet a commercially stable release. Initially the history and background of this system will be explored, then some machine specific aspects of 386BSD will be discussed, such as system startup, I/O configuration, and the virtual memory subsystem. Some discussion of various performance and porting issues will follow, and some conclusions drawn concerning the strengths and weaknesses of this particular UNIX port, and how well it fits onto the 386 and the PC hardware architecture.

Billam, Peter J., "386BSD in the Shrink-Wrap World," *AUUGN*, vol. 14, no. 4, pp. 50-54, AUUG Inc., Sydney, NSW, August 1993.

386BSD offers to the commercial user significant advantages over traditional LAN products: low cost, legality, connectivity, and routable networking protocols. It also has various resources which can be used to increase ease of administration and security, resources which are less available to users of bought software. The author also discusses and administerable user-friendly Unix login shell, and recommends system servers to improve the administerability of networked computers.

Phillips, Andrew, "The ABCs of Unix," *AUUGN*, vol. 14, no. 1, p. 11, AUUG Inc., Sydney, NSW, February 1993.

Striplin, Patricia Dewar, "Australian Users' Views on Open Systems," *AUUGN*, vol. 14, no. 3, pp. 39-42, AUUG Inc., Sydney, NSW, June 1993.

The second of a series of informal end-user discussions organized by Datapro International took place in Sydney in mid-November 1992. Five members of the Australian UNIX Users Group (AUUG) and the Australian User Alliance for Open Systems (AUAOS) sparked frank conversation about the implementation of open systems in a production environment. Mediated by Dr. Philip McCrea, president of AUUG, the closed door session revealed the realities of an "open" choice. While "open systems" now exist for hardware choices, it is applications software and specifically the database management system and development tools that actually drive the decision.

Paddon, Michael, "The AUUG 93 Face Saver," *AUUGN*, vol. 14, no. 5, pp. 17-25, AUUG Inc., Sydney, NSW, October 1993. Email: mwp@iconix.oz.au

Johnstone, Ian, Chris Maltby, and Greg Rose, "A brief note on UNIX System Performance," *AUUGN*, vol. 14, no. 4, pp. 43-44, AUUG Inc., Sydney, NSW, August 1993. ** Reprinted from AUUGN v1n4 **

Naylor, Richard, "Community Computing," *AUUGN*, vol. 14, no. 5, pp. 52-55, AUUG Inc., Sydney, NSW, October 1993. ** Reprinted from UniForum NZ 93 Conference **

Technology is now in all our homes. Modems are cheap and reasonably common. But how is this technology used in our communities, should it be used? This paper discusses experience in making information and Internet access available to the public.

Jester, Rolf, "Cost Savings through Standards," *AUUGN*, vol. 14, no. 1, pp. 32-53, AUUG Inc., Sydney, NSW, February 1993. ** Paper presented at AUUG92 **

Reiss, Harold, "Distributed Object Management," *AUUGN*, vol. 14, no. 6, pp. 78-87, AUUG Inc., Sydney, NSW, December 1993. Email: hr@rdeint.dialix.oz.au

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

Gray, Peter, "Distributing C Compiles," *AUUGN*, vol. 14, no. 3, pp. 60-69, AUUG Inc., Sydney, NSW, June 1993. Email: pdg@cs.uow.edu.au

This paper describes a system written at Wollongong to distribute C compiles off the heavily loaded servers onto the lightly loaded workstations in such a way as to be totally transparent to the user. This is feasible because the compilers involved always read and write files and may not be used as filters.

Booth, Adrian, "The Electronic Interviews - Greg Rose," *AUUGN*, vol. 14, no. 4, pp. 38-42, AUUG Inc., Sydney, NSW, August 1993.

McGrath, Tony, "The First Port is the Deepest," *AUUGN*, vol. 14, no. 4, pp. 45-50, AUUG Inc., Sydney, NSW, August 1993.

There are many events that happen to each of us that help define exactly who we are and why we are the way we are. For me, one of those seminal events took place fifteen years ago while I was an undergraduate student at the University of Wollongong, and the event was the first UNIX port. Looking back over those fifteen years, it has become difficult to accept just what it was that I was involved in, and what effect it would have on myself and the whole UNIX world. This is a short, somewhat biased look at those events, and the people that made them happen.

Biddle, Robert, "Graphic User Interfaces Made Easy? - A Tcl/Tk Tutorial," *AUUGN*, vol. 14, no. 1, pp. 54-70, AUUG Inc., Sydney, NSW, February 1993.

Tcl is a small programming language designed to be embedded into other programs to enable extensibility and customisation. Tk is an X window system toolkit providing graphic user interface facilities in connection with Tcl. Both were developed by John Ousterhout at the University of California at Berkeley, and are available free. Tcl and Tk together offer an easy route to graphic user interface programming in the X environment. This document is a practical introduction to using Tk and Tcl, concentrating on how they can be used to quickly develop flexible graphic user interface facilities. Some simple examples are worked through, and suggestions made as to how to proceed further.

Schulze, Mike, George Benko, and Craig Farrell, "Homebrew Network Monitoring: A Prelude to Network Management," *AUUGN*, vol. 14, no. 3, pp. 70-79, AUUG Inc., Sydney, NSW, June 1993.

A wide variety of public domain and commercial tools exist to help network administrators manage networks. Few of these tools achieve a satisfactory price/performance ratio for organisations with small budgets. To date, we have implemented a number of tools which allow us to examine and visualise network communications with a simple, intuitive, X based graphical user interface. These tools acquire knowledge on network communications via passive network monitoring, with a minimal amount of user intervention. This allows users who do not have an indepth understanding of network architectures to gain an immediate, intuitive understanding of their network and its performance.

<nameless>, "Humour - The Story of a Degenerate," *AUUGN*, vol. 14, no. 3, pp. 54-55, AUUG Inc., Sydney, NSW, June 1993.

Lions, John, "Letter from Bell Labs," *AUUGN*, vol. 14, no. 5, pp. 48-51, AUUG Inc., Sydney, NSW, October 1993. ** Reprinted from AUUGN v1n2 **

Chubb, Peter, "LINUX - An Open System Opened," *AUUGN*, vol. 14, no. 6, pp. 70-77, AUUG Inc., Sydney, NSW, December 1993.

The question to ask with all the hype about open systems is: "Open to whom?" Most proprietary "open" systems are frozen to input from ordinary people. Linux is a Posix conformant operating system that runs on 386 and 486 machines. It has almost all the features one could ask for in an open system: comes with full source code, free (in both the GPL and monetary senses), wide support via the Internet, many utilities and applications; it has multiple file systems (including /proc and MS-DOS file systems), sockets and TCP/IP, SCSI disc, tape and CD-ROM support, virtual memory, a MS-DOS emulator, etc., etc., etc. Linux has been developed cooperatively by many people world-wide. It isn't too late for your contribution to make it into the system - you can't get more open than that!

Bond, Andy, "Load Sharing in a Distributed Environment," *AUUGN*, vol. 14, no. 2, pp. 54-65, AUUG Inc., Sydney, NSW, April 1993.

A network of UNIX workstations is becoming a common sight in modern computing environments. Each workstation provides powerful computing resources which are periodically in strong demand by the local user. However even in busy environments, a significant proportion of these machines will be idle or underutilized at any one time. By supplementing the local computing resources through offloading tasks to idle workstations, better utilization can be made of the distributed computing environment. Such a strategy is commonly known as load sharing or load balancing. This paper discusses and experimental task allocation system called STARS. Using a distributed database of resource usage measurements, STARS allocates tasks to systems based on the availability of resources within the network. We will look at how such a system can be integrated into a users computing environment and some results based on our experience with its use.

Booth, Adrian, "Mining for Gold in the UNIX Kernel," *AUUGN*, vol. 14, no. 3, pp. 22-25, AUUG Inc., Sydney, NSW, June 1993.

Brown, Lawrie, "MUDs - Serious Research Tool or Just Another Game," *AUUGN*, vol. 14, no. 5, pp. 56-62, AUUG Inc., Sydney, NSW, October 1993.

In this talk I intend to introduce the concepts and history of the MUDs, and then provide an overview of some uses for MUDs. These range from an on-line conference room in my own MUD, to testing garbage collection algorithms, experimenting with economic models, simulating a Mars colony, and investigating the psychology of user interactions on MUDs. I'll try to conclude with an idea of where they are going, and some hints for responsible MUD management.

Templeman, Paul, "Network Backup and Archival Strategies," *AUUGN*, vol. 14, no. 5, pp. 66-76, AUUG Inc., Sydney, NSW, October 1993.

This paper looks at backup and archiving strategies that can be employed by system administrators, and how an administrator can assess the strategy that best suits their own environment. Available utilities, both standard UNIX ones, and third-party utilities, are covered. Hardware options, such as tape, optical, and jukebox facilities are also reviewed.

Booth, Adrian, "New COSEy alliance formed," *AUUGN*, vol. 14, no. 2, p. 10, AUUG Inc., Sydney, NSW, April 1993.

Hooten, Anthony D., "Open, Distributed OnLine Transaction Processing," *AUUGN*, vol. 14, no. 4, pp. 55-79, AUUG Inc., Sydney, NSW, August 1993.

Online Transaction Processing (OLTP) is the backbone of businesses throughout the world. Advances in distributed computing and transaction processing software, standards-based computing, and powerful workstations have led to a migration towards open, distributed transaction processing. This paper provides an overview of the evolution to open OLTP and of IBM's open, distributed OLTP structure. This open OLTP structure is built on the Open Software Foundation's Distributed Computing Environment (DCE), utilizes the Encina transaction processing technology from Transarc, and offers a modern, state-of-the-art implementation of IBM's CICS OLTP monitor.

Lauder, Piers, "Share Scheduling Works," *AUUGN*, vol. 14, no. 3, pp. 56-59, AUUG Inc., Sydney, NSW, June 1993. ** Reprinted from AUUGN v2n6 **

The Share Scheduling Algorithm on Unix provides a per-user Scheduler on top of the standard per-process scheduler. It acts to provide equitable allocation of the machine between classes of users, and between users of the same class, according to their allocation of "shares" of the machine.

Trevallyn-Jones, Nik, "Technical Issues with Object Databases," *AUUGN*, vol. 14, no. 3, pp. 80-89, AUUG Inc., Sydney, NSW, June 1993. Email: oda@extro.ucc.su.oz.au

Today's complex applications are relying on software design and programming facilities such as Object Orientation. These systems require a database substrate powerful and flexible enough to cope with the complexity of the data structures, fast enough to provide real-time performance, and an interface simple enough to be usable in such complex applications. The modern ODBMSs are meeting these criteria in most aspects. This paper describes their major abilities, and discusses issues in their implementation.

Ivanov, Peter, "UNIX in a Hostile Environment -or- Mouse Watching by a Cat," *AUUGN*, vol. 14, no. 1, pp. 29-31, AUUG Inc., Sydney, NSW, February 1993. ** Reprinted from AUUGN v1n1 **

Booth, Adrian, "Unix Magic," *AUUGN*, vol. 14, no. 4, pp. 89-90, AUUG Inc., Sydney, NSW, August 1993.

Jackson, Janet, "Unix Traps - endlessly recursive directories," *AUUGN*, vol. 14, no. 4, p. 92, AUUG Inc., Sydney, NSW, August 1993.

Felsche, Bernd, "Unix Tricks & Traps," *AUUGN*, vol. 14, no. 5, pp. 83-85, AUUG Inc., Sydney, NSW, October 1993. Email: bernie@metapro.DIALix.oz.au

Jackson, Janet, "Unix Tricks & Traps: form letters with MH," *AUUGN*, vol. 14, no. 6, p. 107, AUUG Inc., Sydney, NSW, December 1993. Email: jackson@cwr.uwa.edu.au

Booth, Adrian, "Unix Tricks 'n' Traps," *AUUGN*, vol. 14, no. 2, pp. 86-87, AUUG Inc., Sydney, NSW, April 1993.

Huxtable, Glenn, "Unix Tricks - forcing NFS cache flushes," *AUUGN*, vol. 14, no. 4, p. 91, AUUG Inc., Sydney, NSW, August 1993.

Jackson, Janet, "User Support Mailbox," *AUUGN*, vol. 14, no. 2, pp. 87-88, AUUG Inc., Sydney, NSW, April 1993.

Crawford, Frank, "vpcheck - A Daemon to Balance Vector and Scalar Usage," *AUUGN*, vol. 14, no. 2, pp. 40-45, AUUG Inc., Sydney, NSW, April 1993.

Lions, John, "What Ken and Dennis Did Right," *AUUGN*, vol. 14, no. 5, pp. 63-65, AUUG Inc., Sydney, NSW, October 1993.

The UNIX system was originally developed by Ken Thompson and Dennis Ritchie who were later supported by a cast of thousands. The original underlying philosophy was that UNIX would run on a single computer. Such systems were initially isolated. The idea of networking has put an expiry date on many standard UNIX features. Many things will be different in Plan 9. This paper attempts a retrospective assessment of the features of the UNIX system, especially those that are starting to appear a little frayed at the edges.

AUUGN Volume 14 (1993) Index – Papers (Sorted by author)

Author	Title	Issue	Page	Month
<nameless>,	Humour - The Story of a Degenerate	3	54-55	June
Biddle, Robert	Graphic User Interfaces Made Easy? - A Tcl/Tk Tutorial	1	54-70	February
Billam, Peter J.	386BSD in the Shrink-Wrap World	4	50-54	August
Bond, Andy	Load Sharing in a Distributed Environment	2	54-65	April
Booth, Adrian	New COSEy alliance formed	2	10	April
Booth, Adrian	Unix Tricks 'n' Traps	2	86-87	April
Booth, Adrian	Unix Magic	4	89-90	August
Booth, Adrian	The Electronic Interviews - Greg Rose	4	38-42	August
Booth, Adrian	Mining for Gold in the UNIX Kernel	3	22-25	June
Brown, Lawrie	MUDs - Serious Research Tool or Just Another Game	5	56-62	October
Chubb, Peter	LINUX - An Open System Opened	6	70-77	December
Crawford, Frank	vpcheck - A Daemon to Balance Vector and Scalar Usage	2	40-45	April
Felsche, Bernd	Unix Tricks & Traps	5	83-85	October
Gray, Peter	Distributing C Compiles	3	60-69	June
Hooten, Anthony D.	Open, Distributed OnLine Transaction Processing	4	55-79	August
Huxtable, Glenn	Unix Tricks - forcing NFS cache flushes	4	91	August
Ivanov, Peter	UNIX in a Hostile Environment -or- Mouse Watching by a Cat	1	29-31	February
Jackson, Janet	User Support Mailbox	2	87-88	April
Jackson, Janet	Unix Traps - endlessly recursive directories	4	92	August
Jackson, Janet	Unix Tricks & Traps: form letters with MH	6	107	December
Jester, Rolf	Cost Savings through Standards	1	32-53	February
Johnstone, Ian, Chris	A brief note on UNIX System Performance	4	43-44	August
Maltby, and Greg Rose				
Lauder, Piers	Share Scheduling Works	3	56-59	June
Lions, John	Letter from Bell Labs	5	48-51	October
Lions, John	What Ken and Dennis Did Right	5	63-65	October
McGrath, Tony	The First Port is the Deepest	4	45-50	August
McRae, Andrew	386BSD: A Look Under The Hood	2	46-53	April
Naylor, Richard	Community Computing	5	52-55	October
Paddon, Michael	The AUUG 93 Face Saver	5	17-25	October
Phillips, Andrew	The ABCs of Unix	1	11	February
Reiss, Harold	Distributed Object Management	6	78-87	December
Schulze, Mike, George	Homebrew Network Monitoring: A Prelude to Network	3	70-79	June
Benko, and Craig	Management			
Farrell				
Striplin, Patricia Dewar	Australian Users' Views on Open Systems	3	39-42	June
Templeman, Paul	Network Backup and Archival Strategies	5	66-76	October
Trevallyn-Jones, Nik	Technical Issues with Object Databases	3	80-89	June

AUUGN Volume 14 (1993) Index – Reports (Sorted by author)

Author	Title	Issue	Page	Month
Booth, Adrian	WAUG Meeting Reviews	1	17-18	February
Booth, Adrian	Overview of AUUG Summer Technical Conference 1993 - Perth	3	21	June
Booth, Adrian	Ramblings about the AUUG93 Conference and Exhibition	5	12-16	October
Croft, Maxine	WAUG Meeting Reviews	6	57-59	December
Dryden, Julian	AUUG Inc. NSW Chapter	5	27	October
Fraumann, Liz	So, how did AUUG spend the monies at AUUG '93?? or Results through Open Communication...	6	22-24	December
Fraumann, Liz	Mind-Share vs. Market-share - A Perspective	3	10-12	June
Huxtable, Glenn	WAUG Meeting Review - September	5	33	October
Jackson, Janet	WAUG and Perth News	4	21-22	August
Jackson, Janet	Impressions of SAGE-AU 93	4	23-26	August
Jackson, Janet	From the Western Front	6	56	December
Jackson, Janet	WAUG and Perth News	3	20	June
Jackson, Janet	WAUG and Perth News	5	30	October
Major,	WAUG Meeting Review	2	16	April
McCrea, Phil	Report on Fiji Computer Society Conference, March 25-27	3	14	June
Paddon, Michael	AUUG Inc - Victorian Chapter, A Review of the Last Few Months	3	17-19	June
Prince, Stephen	Update on AUUG Inc. - Victorian Chapter Activities	4	19-20	August
Prince, Stephen	Update on AUUG Inc. - Victorian Chapter Activities	6	55	December
Prince, Stephen	Update on AUUG Inc. - Victorian Chapter Activities	5	29	October
Rose, Greg	Report on Usenix, San Diego 25th-29th January 1993	1	24-28	February
Schulze, Mike	AARNet General Manager's Perth Visit	5	31-32	October
Wagner, Michael	AUUG South Australian Chapter Information	4	17	August
White, Mark	AUUG Queensland Chapter Update	4	17	August

AUUGN Volume 14 (1993) Index – Book Reviews (Sorted by author)

Author	Title	Reviewer	Issue
(Ed.), Sally A. Browning	UNIX System V Print Service Administration	Frank Crawford	3
Comer, Douglas E. and David L. Stevens	Internetworking with TCP/IP, Volume III - Client-Server Programming and Applications	Michael Paddon	5
Dowd, Kevin	High Performance Computing	Ian Hoyle	4
Ferbache, D and Shearer, G.	UNIX Installation security and integrity	Tim McGrath	3
Glines, Steve, Peter S. Spicer, Benjamin A. Hunsberger, and Karen Lynn White	Inside SCO Unix	Taso Hatzi	4
Goldstein, Fred R.	ISDN in Perspective	Scott Colwell	4
Hardie, Edward T.L. and Vivian Neou	INTERNET: Mailing Lists	Geoff Huston	5
Newmarch, Jan D.	The X Window System and MOTIF: A Fast Track Approach	Michelle Ritchie	4
Oram, Andrew and Steve Talbott	Managing Projects with make	Frank Crawford	3
Padovano, Michael	Networking Applications on UNIX System V Release 4	Adrian Booth	5
Parker, Tim	Unix User's Handbook	Frank Crawford	2
Peek, Jerry, Tim O'Reilly, and Mike Loukides	UNIX Power Tools	Ian Hoyle	2
Peek, Jerry, Tim O'Reilly, and Mike Loukides	UNIX Power Tools	Frank Crawford	2
Press, Motorola University	The Basic Book of X.25 Packet Switching	Rick Stevenson	4
Press, Unix	Unix System V Release 4.2 Documentation Series	Zoltan Somogyi	5
Rose, Marshall T.	The Internet Message	Nick Frisina	3
Rosenblatt, Bill	Learning the Korn Shell	Frank Crawford	5
Rosenblatt, Bill	Learning the Korn Shell	Gorden Turner	6
Sanderson, David	Smileys	Ian Hoyle	4
Stern, Hal	Managing NFS and NIS	Greg Rose	2
Welsh, Matt	Linux Installation and Getting Started	Dave Watson	5
X/Open Company Limited	(January 1992): CAE Specification X/Open Transport Interface	Keith Lugsden	6
X/Open Company Limited	X/Open Guide - Distributed Transaction Processing Reference Model	Michael Fazzolare	6
X/Open Company Limited	Preliminary Specification of: Distributed Transaction Processing: The TX (Transaction Demarcation) Specification	Michael Fazzolare	6
X/Open Company Limited	Systems Management: Management Protocols API (XMP)	David Hughes	6

Calendar of Events

<i>Date</i>	<i>Org</i>	<i>Event</i>	<i>City</i>
1994			
Feb 3-5	AUUG	McKusick Workshop,	Melbourne
Feb 9	AUUG	McKusick Workshop,	Darwin
Feb 10-11	AUUG	Summer Conference Series	Darwin
Feb 15	AUUG/ACS	Summer Conference Series	Hobart
Feb 14	AUUG	McKusick Workshop	Canberra
Feb 15-16	AUUG	Summer Conference Series	Canberra
Feb 21-22	AUUG	McKusick Workshop	Perth
Feb 23	AUUG	Summer Conference Series	Perth
Feb 28	AUUG	McKusick Workshop	Adelaide
Mar 1	AUUG	Summer Conference Series	Adelaide
Mar 3	AUUG	McKusick Workshop	Sydney
Mar 4	AUUG	Summer Conference Series	Sydney
Mar 9-10	AUUG	McKusick Workshop	Hobart
Mar 15-16	AUUG	Summer Conference Series	Melbourne
Mar 21-25	UniForum	Conference & Exhibition	San Francisco, CA,
Mar 24	AUUG	McKusick Workshop	Brisbane
Mar 25	AUUG	Summer Conference Series	Brisbane
Apr 5-9	Usenix	SANS II	Arlington, VA
Apr 11-14	Usenix	C++	Cambridge, MA
Apr 18-22	IEEE	1003	Lake Tahoe, CA
Apr 25-28	Usenix	Unix Applications Dev.	Toronto, Canada
May 2-6			
May 7-13	DECUS	Networld + INTEROP '94	Las Vegas, NV
May 18-21	UniForum NZ	DECUS - Summer '94	New Orleans, LA
		UniForum NZ Conference	Rotorua, New Zealand
Jun 6-10	Usenix	USENIX	Boston, MA
Jun 6-10		Networld + INTEROP	Berlin, Germany
Jul 11-15	IEEE	IEEE 1003	
Sep 6-9	AUUG	AUUG '94	Melbourne
Sep 12-14		Networld + INTEROP	Atlanta, GA
Sep 19-23	Usenix	LISA VIII	San Diego, CA
Oct 23-27	ACM	OOPSLA	Portland, OR
Oct 26-28	Usenix	Very High Level Languages	Sante Fe, NM
Nov 12-18	DECUS	DECUS - Winter '94	Anaheim, CA

<i>Date</i>	<i>Org</i>	<i>Event</i>	<i>City</i>
1995			
Jan 16-20	Usenix	USENIX	New Orleans, LA
Feb	AUUG	Summer Conference Series	All States in Australia
Feb 21-23	UniForum	UniForum	Dallas, TX
May 13-19	DECUS	DECUS - Summer '95	Atlanta, GA.
Jun 19-23	Usenix	USENIX	San Francisco, CA
Sep 19-22	AUUG	AUUG '95	Sydney
Nov 2-8	DECUS	DECUS - Winter '95	San Francisco, CA
1996			
Jan 22-26	Usenix	USENIX	San Diego, CA
Feb	AUUG	Summer Conference Series	All States in Australia
Mar 12-14	UniForum	UniForum	San Francisco, CA
May 18-24	DECUS	DECUS - Summer '96	Orlando, FL
Nov 16-22	DECUS	DECUS - Winter '96	Anaheim, CA

AUUG Inc. is pleased to provide the following worldwide calendar of events, in cooperation with other open systems bodies. For updated information please contact the sponsoring organisation directly.