

# AUUGN

The Journal of AUUG Inc.

Volume 16, Number 5  
October 1995

## AUUG '95 wrap-up

Tim O'Reilly interview

Photos from the conference

John Lions interview

... plus reviews, Chapter news, and more

AUUG  
Inc.

UNIX & OPEN SYSTEMS USERS



# AUUGN

Volume 16, Number 5  
October, 1995

## **AUUG Membership & General Correspondence**

The AUUG Secretary  
PO Box 366  
Kensington NSW 2033

Tel: (02) 361 5994  
Fax: (02) 332 4066  
Freephone: 1-800-625-655  
E-mail: [auug@auug.org.au](mailto:auug@auug.org.au)

## **AUUG Executive**

President: Michael Paddon  
[mwp@munnari.oz.au](mailto:mwp@munnari.oz.au)

Kodak  
173 Elizabeth St.  
Coburg VIC 3058

Vice President: Glenn Huxtable  
[glenn@fs.com.au](mailto:glenn@fs.com.au)

Functional Software  
PO Box 192  
Leederville WA 6903

Secretary: Peter Wishart  
[pjw@auug.org.au](mailto:pjw@auug.org.au)  
EASAMS/GEC Marconi Systems  
PO Box 4806  
Unit 7, 10 Kennedy St.  
Kingston ACT 2604

Treasurer: Stephen Boucher  
[stephen@mtiame.mtia.oz.au](mailto:stephen@mtiame.mtia.oz.au)  
MTIA  
509 St. Kilda Road  
Melbourne VIC 3004

## **Committee Members:**

Phil McCrea (Past President):  
[pmc@syd.dit.csiro.au](mailto:pmc@syd.dit.csiro.au)  
Division of Information Technology  
CSIRO  
Building E6B  
Macquarie University NSW 2113

Frank Crawford  
[frank@atom.ansto.gov.au](mailto:frank@atom.ansto.gov.au)  
ANSTO  
Private Mail Bag 1  
Menai NSW 2234

Lucy Chubb  
[lucyc@sw.oz.au](mailto:lucyc@sw.oz.au)  
Softway Pty. Ltd.  
PO Box 305  
Strawberry Hills NSW 2021

Chris Maltby  
[chris@sw.oz.au](mailto:chris@sw.oz.au)  
Softway Pty. Ltd.  
PO Box 305  
Strawberry Hills NSW 2021

David Purdue  
[David.Purdue@aus.sun.com](mailto:David.Purdue@aus.sun.com)  
SunSoft  
119 Willoughby Rd.  
Crows Nest NSW 2065

## **AUUGN Business Manager**

Elizabeth Egan  
[auug@auug.org.au](mailto:auug@auug.org.au)  
PO Box 366  
Kensington NSW 2033

# Table of Contents

## **Editorial 3**

### **President's Report 3**

*Michael Paddon*

### **Letters 4**

### **Conferences & Announcements 4**

### **Conference report: APWWW/AUUG '95 The President's view 11**

*Michael Paddon*

### **Traditional fare: Conference photos 12**

*Adrian Booth*

### **Interview: Tim O'Reilly at AUUG '95 13**

*Frank Crawford*

### **Interview: John Lions 15**

*Adrian Booth*

### **Viewpoint: AUUG95-APWWW95 wrapup 16**

*Frank Crawford*

### **Overview: AUUG95 Network 17**

*Frank Crawford*

### **Background: The Cryptography Debate in Australia 18**

*Chris Maltby*

### **Background: Netscape Break-in 19**

*David Purdue*

AUUG Inc. acknowledges the  
generous support of its  
corporate sponsors...

## Opinion: Software Minimalism 20

Jamie Honan

## Just when you thought it was safe... More photos 21

Adrian Booth ... again!

## UNIX Tricks & Traps 22

Janet Jackson (Sub-editor)

## Book Reviews 26

Frank Crawford (Sub-editor)

Reviews by Warren Tomey, Michael Usher, Adrian Booth,  
Kate Lance, John Wright, Michael Haldey

Internet Security Firewalls . . . . .	26
Casting the Net: From ARPANET to INTERNET and beyond... . . . . .	27
The Magic Garden Explained Solutions Manual . . . . .	28
Learning the Bash Shell . . . . .	29
Teach Yourself PERL in 21 Days . . . . .	30
System Programming for UNIX . . . . .	31
OzInternet: A Guide to connecting to the Internet in Australia . . . . .	32

## WAAUG news: From the Western Front 34

Janet Jackson (Sub-editor)

## QAUUG news: Qld. Chapter report 38

## AUUG Canberra news: Canberra chapter 38

## From ;login: 39

## AARNET Mail service details

## AUUG Membership applications & change notification

## Calendar of Events Inside back cover

### AUUGN Correspondence

Please send all correspondence regarding AUUGN to:

AUUGN Editor  
PO Box 366  
Kensington NSW 2033  
E-mail: [auugn@auug.org.au](mailto:auugn@auug.org.au)

### Submission Guidelines

Submission guidelines for AUUGN contributions are regularly posted on the [aus.org.auug](http://aus.org.auug) news group. They are also available from the AUUG World Wide Web site at <http://www.auug.org.au>

Alternately, e-mail to the above correspondence address, requesting a copy.

### AUUGN Back-Issues

A variety of back-issues of AUUGN are still available; for price and availability details, 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 copies of the Conference Proceedings from previous AUUG Conferences are still available, at \$50 each for members, and \$60 for non-members. Contact the AUUG Secretariat for details.

### Mailing Lists

Inquiries regarding the purchase of the AUUGN mailing list should be directed to the AUUG Secretariat. Telephone (02) 361 5994 during business hours, or Fax (02) 332 4066.

### Disclaimer

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

### Copyright Information

Copyright ©1995 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<sup>®</sup> 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(s) must be given. Abstracting with credit is permitted. No other reproduction is permitted without prior permission of AUUG Incorporated.

### Trade marks

UNIX is a registered trade mark of X/Open in the United States and other countries.

# Contribution deadlines for AUUGN in 1995/96

Vol 16, #6 (December): November 20

Vol 17, #1 (February '96): January 25

Vol 17, #2 (April '96): March 22

# Editorial

Phil Anderson <phila@osa.com.au>

I should have been at the polo today, quaffing bubbly with the smart set (they're the information-poor folk we keep hearing about) and hoping to get my photo taken by Rennie Ellis. Instead, I'm chained to my PC and slapping—er, carefully assembling, this issue of AUUGN.

It's not that I really like polo; heaven knows that it's not like the sport makes a great deal of sense. A large bunch of blokey types on horseback who think Kerry's a jolly wheeze, hurtling around a huge rectangle of soon-to-be-mud whacking some innocent ball with golf clubs on steroids? Sport? Performance art okay, but sport?

No, I was going to go for the company, the fresh air, the picnic lunch ... okay; I was going for the performance art thing too. Yet here I am in my overcrowded study, bathed in the reassuring glow of my monitor, staring morosely at pages and pages of contributions from loyal contributors – most of it a bit late, but appreciated nonetheless. At least, as appreciated as it can be by one who has been denied that liquid we used to call champagne, but don't any more. \*sigh\*

I should get out more.

•

There's quite a bit in this issue about AUUG '95. Even photos to entertain the membership—yes, most of those featured *do* look like that—and horrify the subjects. Don't laugh too loud; we may be forced to send the official AUUG photographers around to your next function ... then you'll be sorry!

•

This issue's pretty late; sorry 'bout that, but with the help of all you enthusiastic contributors, we'll get the December issue in your letterboxes before the end of the year.❖



# President's Report

Michael Paddon <mwp@munnari.oz.au>

The winter conference has come and gone, marking the effective (if not official) end of another AUUG year. When the conference is approaching it tends to consume the attention of your national and chapter committees and when it is past we tend to stop for breath before plunging into the year ahead. Despite this, there has been some significant activity of which you should be aware.

The last committee meeting saw the official formation of a chapter in Tasmania. I'd like to extend a warm welcome to our newest chapter on behalf of all our members. As some of you may be aware, there has been a *de facto* chapter in operation in our southernmost for several years, however its newfound official status allows us to better support and fund its activities.

AUUG now has chapters in all Australian states and territories, which is an enormous achievement when you consider that three or four years ago we only had two, and those were far more loosely organised than today. The chapter programme has brought enormous benefits to our grass root members—summer conferences, tutorial roadshows, and regular local meetings, to name but a few—and been an extremely effective vehicle in putting AUUG's financial resources to their best use.

I am also pleased to report that AUUG-Vic is on its feet again following a hiatus of several months. The chapter success story has come about due to the hard work and dedication of a host of people, but without the continued involvement of the membership they tend to fall fallow. Indeed, without your continued involvement they have no reason to exist. So get along to a local meeting or two and see what it's all about. I guarantee that you'll, at the least, you'll have fun.

The annual Chapter Council was held in September in Sydney. This is a chance for representatives of every chapter to get together with the AUUG national committee to discuss plans and map out strategies for the coming year. In 1994, we held the council during the AUUG conference, where it turned out to be extremely difficult to find ample time to cover the agenda. This year, therefore, we felt it was much better to schedule it separately.

A lot of useful ground was covered by the council. Tutorial roadshows have proved to be extremely

popular, and there was a definite feeling that we should continue and maybe even increase the number held per year. Talk to us if you know someone who would be a good candidate for such an event. Much of the discussion centered on how the chapters could better cooperate and learn from each other's experiences, and there was also a lively debate on how to best manage the ever more healthy funds held in trust by the chapter committees. A full summary of the meeting will be forthcoming in AUUGN.

Over the past few months, AUUG has been talking to Access One (a national Internet service provider) in order to arrange for preferential pricing for our members. At AUUG 95, we were able to announce that Access One will be offering a flat rate of \$3 an hour—no joining fee, no download fees—for full Internet connectivity. I'd like to stress that we have not entered into an exclusive arrangement (nor shall we) and that we'd like to hear from other ISPs who may be interested in offering similar benefits.❖

---

## Letters

### Looking for old UNIX users/hackers

Warren Toomey <wkt@cs.adfa.oz.au>

This quick note is an attempt to track down all those users and hackers of early versions of UNIX. I've found, while trying to get Seventh Edition UNIX running and helping others to get Sixth and Seventh Edition running, that there isn't a good place to look for help. So I thought that it would be a good idea to compile a list of people who might have knowledge, skills, source code, distributions, documentation, anecdotes etc. so that this information could be saved from the great bit bucket.

Participating in this loose affiliation is not meant to be onerous. If you have anything worth saving, send it to me and I'll archive it. If you want to join a mailing list because you have some skills/experience and can help other people, that would be great. It's entirely up to you.

As old UNIXes were licensed, no distribution of licensed material to non-license-holders will take place. Finally, I'm setting up a WWW home page for this sort of thing, with all the non-licensed stuff available there. My skeletal start is available at <http://minnie.cs.adfa.oz.au/PUPS> ❖

---

## Conferences & Announcements

### Call for Papers and Tutorial Presentations: UniForum NZ '96 Conference - Call

The 1996 UniForum NZ Conference and Exhibition will take place in Rotorua 22-25 May 1996.

Both introductory and in-depth papers, technical and non-technical, are called for.

Themes we would like to have running at this Conference include the following:

- Technology Convergence
- Hypermedia
- The Internet and WWW
- The NZ Marketplace
- Open Networking (including Security)
- User Tools
- Executive Update/Overview
- The Evolution of UNIX

In particular it is intended to have a special stream for beginners, so introductory papers of a mini-tutorial nature are especially welcome.

We are also looking for offers of half- and full-day tutorial presentations. Once again these may be introductory or in-depth in nature, and may cover management, technical, or user-oriented issues.

These do not require a written paper, but pre-prepared notes are expected to be provided to those attending the tutorials.

For each full written paper received in time for publication in the Proceedings, a rebate off the conference registration fee of no more than the value of the early-bird member rate will be provided (joint authors may share the rebate). Tutorial presenters will receive a small remuneration in addition.

### Important dates for presenters

- Registration of interest  
NOW!!
- Commitment to present a paper or tutorial  
31 Oct 1995
- Speaker profile and initial abstract required  
1 Dec 1995
- Notification of Paper Acceptance  
20 Dec 1995

- Full paper required for Proceedings publication  
29 Mar 1996
- Speaker registration form required  
29 Mar 1996
- Presentation abstract and A/V requirements  
29 Mar 1996
- Tutorial notes required for reproduction  
4 May 1996

Contact me to register interest in presenting a paper or tutorial.

Ray Brownrigg  
UniForum NZ '96 Conference Program Coordinator  
Phone: +64 4 472-1000  
FAX: +64 4 495-5118  
ray@isor.vuw.ac.nz

## Upcoming USENIX events

**January 22-26, 1996**

### USENIX 1996 TECHNICAL CONFERENCE

San Diego Marriott Hotel and Marina, San Diego, California Sponsored by the USENIX Association

Program Chair: Bob Gray, US WEST  
Tutorial Program: January 22-23, 1996  
Keynote Address: 9 am, January 24

"Nature and Nurture: The Interplay of UNIX and Networking" by Van Jacobsen, Lawrence Berkeley National Laboratory Technical Sessions, including Refereed Papers and Invited Talks tracks:  
January 24-26, 1996

Birds-of-a-Feather Sessions: evenings, January 23-25,  
1996 USENIX Reception: evening, January 24, 1996  
Vendor Display: January 24-25, 1996

**May 13-17, 1996**

### 5th UNIX SYSTEM ADMINISTRATION, NETWORKING, AND SECURITY SYMPOSIUM (SANS V) Washington, DC

Sponsored by the Open Systems Conference Board, in cooperation with SAGE, the System Administrators Guild, FedUNIX, and the Escal Institute Program

Chairs: Rob Kolsatd and Michelle Crabb

**June 17-21, 1996**

### 2ND CONFERENCE ON OBJECT-ORIENTED TECHNOLOGIES AND SYSTEMS (COOTS) Marriott Eaton Centre, Toronto, Canada

Sponsored by the USENIX Association

Program Chair: Douglas C. Schmidt, Washington University

Tutorial Program Chair: Doug Lea, SUNY Oswego  
Submissions due: February 13, 1996  
Final papers due: May 17, 1996

**July 10-13, 1996**

### 4TH TCL/TK WORKSHOP (TCL/TK 96)

Doubletree Hotel, Monterey, California

Sponsored by the USENIX Association

Program Chairs: Mark Diekhans, Santa Cruz Operation and Mark Roseman, University of Calgary

Paper and demo submissions due: March 5, 1996; copy due: May 28, 1996

**July 22-25, 1996**

### 6TH USENIX SECURITY SYMPOSIUM

Fairmont Hotel, San Jose, California

Sponsored by the USENIX Association. Co-sponsored by UniForum (pending) in cooperation with the Computer Emergency Response Team (CERT), and IFIP WG 11.4

Program Chair: Greg Rose, Sterling Software

Extended abstract submissions due: March 19, 1996  
Final papers due: June 10, 1996

**September 13-17, 1996**

### 10TH USENIX SYSTEMS ADMINISTRATION CONFERENCE (LISA '96) Marriott Hotel, Chicago, Illinois

Sponsored by the USENIX Association and SAGE, the System Administrators Guild Program Chairs: Helen Harrison, SAS Institute and Amy Kreiling, University of North Carolina

Extended abstract submissions due: May 7, 1996  
Final Papers due: August 15, 1996

**October 29-November 1, 1996**

### 2ND USENIX SYMPOSIUM ON OPERATING SYSTEMS DESIGN AND IMPLEMENTATION (OSDI II) Westin Hotel, Seattle, Washington

Sponsored by the USENIX Association

Co-sponsored by ACM SIGOPS (pending) and IEEE TCOS

Program Chairs: Karin Petersen, Xerox PARC and Willy Zwaenepoel, Rice University

Full paper submissions due: May 7, 1996  
Final papers due: September 16, 1996

## Privacy, the Web, Firewalls, Java Among Topics in Technical Conference Program Announced by USENIX Association

The USENIX Association and Program Chair, Robert Gray of US West Advanced Technologies, have just released the tutorial and technical program for its annual Technical Conference to be held at the San Diego Marriott Hotel & Marina, January 22-26, 1996.

Top technologists, engineers, and systems administrators meet at the critically important yet highly informal USENIX Technical Conference to discuss the state-of-the-art developments unveiled there. Many of the high-tech breakthroughs that change the way we work and live have been announced at USENIX Conferences. The January conference is expected to attract over 1400 attendees from the UNIX and advanced computing systems community around the world.

USENIX will offer twenty-one day-long tutorials on January 22-23. Among the twelve brand-new topics offered will be: Java, UNIX Security Tools, Advanced Tcl/Tk, Creating Web Sites, Windows 95 for UNIX Jocks, Your Rights on the Internet, and What's New in Networking. Back by popular demand are tutorials on firewalls, Sendmail, Perl, advanced UNIX security, introduction to Tcl/Tk, and system administration.

The Technical Program, January 24-26, begins with the Keynote Address by Van Jacobson of Lawrence Berkeley National Laboratory on "Nature and Nurture: The Interplay of UNIX and Networking." Mr. Jacobson was one of the creators of TCP/IP, arguably the world's most successful networking protocol. During the Technical Program, refereed papers will present new research in file systems, operating system extensions, multimedia, networks, distributed systems, and protocols.

Of particular interest will be papers comparing various UNIX flavours you can run on your Pentium PC, and presentations on how to conserve bandwidth on the World Wide Web.

Among the Invited Talks are Linus Torvalds on "Linux: Architecture, Experiences, and Future," "Cryptography in the 21st Century," and "Forming a More Perfect Net Governance," by Carey E. Heckman, Co-Director of Stanford Law and Technology Policy Center.

Two panels will explore "Recent Legal Decisions" and "Selling Stuff that's Free: the Commercial Side of Free Software."

The Technical Executive Summary will feature eight experts summarizing the state-of-the-art in their field of specialization.

A lively, informal Vendor Display will take place on January 24-25, with demonstrations of the latest hardware, software, and networking products.

### Announcement & Call for Papers:

## 4TH ANNUAL TCL/TK WORKSHOP (TCL/TK 96)

July 10-13, 1996

Doubletree Hotel, Monterey, California

Sponsored by the USENIX Association

The fourth annual Tcl/Tk workshop is a forum to:

- Bring together Tcl/Tk researchers and practitioners.
- Publish and present current work.
- Plan for future Tcl/Tk related developments.

The workshop program will include formal presentations of papers and demonstrations, as well as informal demonstrations, work-in-progress sessions, birds-of-a-feather sessions, and tutorials.

Papers and demonstrations should report on original Tcl/Tk research. Example topics include system extensions, novel Tcl/Tk based applications, reports on experiences building particular applications, use of different programming paradigms within Tcl, and proposals for new directions. The audience for the workshop is researchers and practitioners who are expert users of Tcl/Tk.

There are three types of submissions: applications papers, general papers, and demonstrations. Paper authors will be given a twenty-minute time slot at the workshop. Demonstrations are intended as a forum to highlight and describe innovative techniques having a highly visual or interactive component; they are not intended as a forum for marketing-oriented presentations. Live, non-commercial demonstrations of software will be given a thirty-minute time slot. A paper of up to four pages must accompany the demonstration. Detailed instructions on submission format appear below.

Applications papers have typically proven difficult to write. Authors considering submission of these types of papers are encouraged to consider the following common causes of rejection:

- Paper is blatant marketing material for commercial product.



- Insufficient background on application domain so that the audience cannot follow; excessive use of domain-specific buzzwords.
- Too much information on the application, but not enough on the relevance of Tcl/Tk to the application.
- Too little consideration of how the Tcl/Tk community could benefit from experiences; limited generalizability.
- Application illustrates only a routine usage of Tcl/Tk.

### DETAILED SUBMISSION INSTRUCTIONS

Deadlines for paper and demo submissions: March 5, 1996  
 Notification of acceptance: April 16, 1996  
 Camera-ready copy due: May 28, 1996

We are accepting workshop submissions via email. Submissions should be sent as both plain text (with no extra mark-up), and as Postscript (formatted for an 8.5 x 11 inch page). When submitting Postscript, please strive to ensure that your file can be printed on a variety of printers. If accepted, both electronic and camera-ready hard copy of the final version will be required.

Submissions for both applications papers and general papers must be full-length versions, and not just abstracts. Papers may be a maximum of ten pages in length. If accepted, we would encourage use of brief video clips or demos during the presentation. If you think you may use AV equipment other than standard overheads or 35mm slides, please make a note of it on the cover sheet described below. (This information is not used to judge your submission, but will assist in organizing the program.)

Submitters for demonstrations must provide a paper of up to four pages in length describing the demonstration and providing further background; this will appear in the workshop proceedings. You are encouraged to submit additional material supporting your demonstration for review, such as a storyboard or outline. If you would like to submit other material, such as videotapes, contact the conference chairs for more information.

Each submission should include a cover letter containing the following information:

- Name of all authors
- Primary contact
- Full postal address
- Telephone number
- Email address (very important)

- Category (application paper, general paper, or demonstration)
- Anticipated AV needs (used only for planning purposes)

Submissions should consist of a uuencoded, compressed tar file (compress or gzip) containing both the plain text and Postscript versions (filenames should be based on your last name, e.g. smith.txt and smith.ps). The tar file should be e-mailed, along with the cover letter to tcl96@sco.com. Receipt of submissions will be acknowledged by return email within the week. If an acknowledgement is not received, please contact the co-chairs.

This workshop, like most conferences and journals, requires that papers not be submitted simultaneously to another conference or publication and that submitted papers not be previously or subsequently published elsewhere. Papers accompanied by "nondisclosure agreement" forms are not acceptable and will be returned to the author unread. All submissions are held in the highest confidentiality prior to publication in the Proceedings, both as a matter of policy and in accord with the U.S. Copyright Act of 1976.

### Program committee

Co-chair: Mark Diekhans, SCO <markd@sco.com>  
 Co-chair: Mark Roseman, University of Calgary <roseman@cpsc.ucalgary.ca>  
 Ben Bederson, University of New Mexico  
 Wayne Christopher, ICEM CFD Engineering  
 Joe Konstan, University of Minnesota  
 Don Libes, NIST  
 Michael McLennan, AT&T  
 Larry Rowe, University of California Berkeley  
 Brent Welch, Sun  
 David Young, SCO  
 Will Wilbrink, Unisys Canada

### Registration materials

Materials containing all details of the technical and tutorial programs, registration fees and forms, and hotel information will be available in April 1996. If you wish to receive the registration materials, please contact:

USENIX Conference Office  
 22672 Lambert Street, Suite 613  
 Lake Forest CA 92630

**Announcement and Preliminary Call for Papers:  
6TH USENIX UNIX SECURITY  
SYMPOSIUM**

Focusing on Applications of Cryptography

July 22-25, 1996

Fairmont Hotel, San Jose, California

Sponsored by the USENIX Association, the UNIX and Advanced Computing Systems Professional and Technical Association

Co-sponsored by UniForum (pending), in cooperation with: The Computer Emergency Response Team (CERT), and IFIP WG 11.4

The goal of this symposium is to bring together security and cryptography practitioners, researchers, system administrators, systems programmers, and others with an interest in applying cryptography, network and computer security, and especially the area where these overlap. The focus on applications of cryptography is intended to attract papers in the fields of electronic commerce and information processing, as well as security. This will be a four-day, single-track symposium with tutorials, refereed and technical presentations, and panel discussions.

**Tutorial program-July 22-23, 1995**

Tutorials for both technical staff and managers will provide immediately useful, practical information on topics such as local and network security precautions, what cryptography can and cannot do, security mechanisms and policies, firewalls and monitoring systems.

**Technical sessions-July 24-25, 1995**

In addition to the keynote presentation, the technical program includes refereed papers and invited talks. There may be panel sessions. There will be Birds-of-a-Feather sessions and Works-in Progress Reports on two evenings. You are invited to make suggestions to the program committee via email to [security@usenix.org](mailto:security@usenix.org).

Papers that have been formally reviewed and accepted will be presented during the symposium and published in the proceedings. Proceedings will be provided free to technical session attendees; additional copies will be available for purchase from USENIX.

**Symposium topics**

Presentations are being solicited especially but not exclusively in the following areas. Please note: papers

about new cryptographic algorithms are not solicited; however, those about new applications are. This symposium is not about new codes or ciphers, or cryptanalysis for its own sake.

- Anonymous transactions
- Applications of cryptographic techniques
- Attacks against secure networks/machines
- Cryptanalysis and codebreaking as attacks
- Cryptographic tools
- Electronic commerce security
- Firewalls and firewall toolkits
- Legislative and legal issues
- Case studies
- Computer misuse and anomaly detection
- File and File system security
- Network security
- Security and system management
- Security in heterogeneous environments
- Security incident investigation and response
- Security tools
- User/system authentication
- Penetration testing
- Malicious code analysis

**Program committee**

Greg Rose, Sterling Software, Program Chair  
Fred Avolio, Trusted Information Systems, Inc.  
Steve Bellovin, AT&T Bell Laboratories  
Brent Chapman, Great Circle Associates  
Diane Coe, MITRE Corporation  
Ed DeHart, CERT  
Dan Geer, Open Market Inc.  
Peter Gutmann, University of Auckland  
Kent Landfield, Sterling Software  
Clifford Neuman, Information Sciences Institute  
Avi Rubin, Bellcore  
Eugene Spafford, COAST Lab., Purdue University  
Ken van Wyk, Defense Information Systems Agency  
Karen Worstell, The Boeing Company

Readers:

Matt Bishop, U.C. Davis  
Phil Karn, Qualcomm

**Dates for refereed paper submission**

Extended abstracts due: March 19, 1996

Authors notified of acceptance: April 15, 1996 Camera-ready, final papers due: June 10, 1996

## How to submit a refereed paper

Submissions must be received by March 19, 1996. Authors are encouraged to submit an extended abstract which discusses key ideas and demonstrates the structure of the finished paper. Extended abstracts should be 3 to 5 pages long (about 1500-2500 words), not counting references and figures. The body of the extended abstract should be in complete paragraphs. The object of an extended abstract is to convince the reviewers that a good paper and presentation will result. Full papers can be submitted if they are complete in advance of the due date. Full papers should be 8 to 15 typeset pages.

If you would like to receive detailed guidelines for submission and examples of an extended abstracts, you may send email to [securityauthors@usenix.org](mailto:securityauthors@usenix.org) or telephone the USENIX Association office at +1 510 528 8649.

Please accompany each submission by a cover letter stating the paper title and authors along with the name of the person who will act as the contact to the program committee. Please include a surface mail address, daytime and evening telephone number, and, if available, an email address and fax number for the contact person.

The UNIX Security Symposium, like most conferences and journals, requires that papers not be submitted simultaneously to another conference or publication and that submitted papers not be previously or subsequently published elsewhere.

Papers accompanied by "nondisclosure agreement" forms are not acceptable and will be returned to the author unread. All submissions are held in the highest confidentiality prior to publication in the Proceedings, both as a matter of policy and in accord with the U.S. Copyright Act of 1976.

All submissions will be judged on originality, relevance, and correctness. Each accepted submission will be assigned a member of the program committee to act as its shepherd through the preparation of the final paper. The assigned member will act as a conduit for feedback from the committee to the authors.

## Where to submit

Please send one copy of an extended abstract or a full paper to the program committee via two of the following methods. All submissions will be acknowledged.

## Preferred Method

Email (Postscript or ASCII) to:  
[securitypapers@usenix.org](mailto:securitypapers@usenix.org)

## Additional methods

Postal delivery to:

Security Symposium  
USENIX Association  
2560 Ninth St.  
Berkeley CA 94710 U.S.A.

Telephone: +1 510 528 8649

Fax to:

USENIX Association  
+1 510 548 5738

## Registration information

The complete program and registration information will be available at the end of April 1996. If you would like to receive registration materials, please contact:

USENIX Conference Office  
22672 Lambert Street, Suite 613  
Lake Forest, CA 92630  
Telephone: +1 714 588 8649  
Fax: +1 714 588 9706.  
Email: [conference@usenix.org](mailto:conference@usenix.org)  
URL: <http://www.usenix.org>

USENIX is the UNIX and Advanced Computing Systems Technical and Professional Association. Since 1975 the USENIX Association has brought together the community of engineers, scientists, and technicians working on the cutting edge of the computing world. The USENIX conferences have become the essential meeting grounds for the presentation and discussion of the most advanced information on the developments of all aspects of computing systems.

For more information about USENIX events, please: -  
Contact USENIX Conference Office, 22672 Lambert Street, Suite 613, Lake Forest, CA USA 92630,  
Telephone +1 714 588 8649, Fax +1 714 588 9706

Email: [conference@usenix.org](mailto:conference@usenix.org)

Access the USENIX Resource Center on the World Wide Web--the URL is <http://www.usenix.org>

Email to our mailserver at [info@usenix.org](mailto:info@usenix.org). In your message include the line "send conferences catalog." ❖

# AUUG NSW Chapter Summer Conference 1996

## "The Commercial Internet Comes Of Age"

### Blue Mountains, Feb 28-March 1, 1996

#### **CALL FOR PAPERS**

We all now know what a wonderful, if not essential, gestalt entity the Internet is. We read every day about how we must get connected or perish.

But, after the Internet has found a cure for cancer, rediscovered Lasseter's Reef and invented a perpetual motion machine, some serious questions will remain: What, in my business, will I use the Internet for?

Do I really want to be connected, and why? If I regard a World Wide Web page as a good form of advertising, should I be surprised when my employees spend their whole day surfing the Web? If I do want to connect, how should I go about it?

If you can answer any of these questions, or if you have more questions of your own, you are invited to submit a paper for the AUUG NSW Chapter Summer Conference 1996.

We also seek papers on topics related to UNIX and Open Systems.

#### **TUTORIALS**

We seek tutorial presentations on topics related to the Internet, UNIX and Open Systems. A day of tutorials will precede the conference, and half day or full day tutorials are invited.

#### **LOCATION**

In order to better serve those members of AUUG NSW Chapter who live outside Sydney, we will be holding the Summer 96 conference in the Blue Mountains.

The Conference will consist of 1 day of tutorials and two days of paper presentations.

#### **SUBMISSION GUIDELINES**

Those proposing to present papers should submit an extended abstract (1-3 pages) and a brief biography.

Those submitting tutorial proposals should submit a plan for the tutorial and a brief biography.

Please indicate on your proposal whether it is a paper or tutorial, and for tutorials whether they are half day or full day.

#### **PROPOSALS SHOULD BE SENT TO:**

AUUG NSW Summer 96 Program Committee  
PO Box 6425  
North Sydney, NSW, 2060  
Phone: David Purdue on (02) 906 3766  
Email: [auug-nsw-exec@auug.org.au](mailto:auug-nsw-exec@auug.org.au)

#### **IMPORTANT DATES**

Abstracts/Proposals due: 1 October 1995  
Authors notified: 1 November 1995  
Final copy due: 1 February 1996  
Conference: 28 Feb-1 Mar 1996

Conference report:

# APWWW/AUUG '95

## The President's view

Michael Paddon <mwp@munnari.oz.au>

An undertaking as large as the AUUG Conference and trade show will never be without blemish, but I found the week enjoyable, informative and fun. From a numerical viewpoint the event was certainly a tremendous success, with the greatest number of attendees and the largest exhibition ever.

However, growth is not an end in itself, and AUUG places far more importance on how the conference serves the needs of the delegates. I have received several criticisms on the content, especially on the heavy emphasis shown to the World Wide Web by the programme. This was always inevitable, given that our chosen theme was "The Internet Means Business" and that we later joined the event with the Asia-Pacific World Wide Web Conference.

---

*...both parties gained enormously  
by cooperating...*

---

Sharing the programme and the organisation with Charles Sturt University was a radical experience for AUUG, and I'd strongly contend that both parties gained enormously by cooperating: our experience versus their enthusiasm and fresh ideas, our broad focus versus their special interest. I don't think we'll ever again co-organise an event in exactly the same way, but it's been a learning experience and we certainly may try something similar in the future. I'm interested to hear what you think of the results with the all the benefits of hindsight.

Despite all this, there were still several strong threads running throughout AUUG 95 that had little or nothing to do with the web. I personally found the most difficult problem was choosing between four streams and, in effect, only being able to see a quarter of the proceedings. Next year, we will almost certainly reduce the number of concurrent sessions and widen

the focus, especially since the web's "flavour of the month" hysteria will have ebbed.

In the final analysis, however, the content of our conferences is decided by the feedback we receive from members (which guides us on themes and who to invite as keynotes) and the papers we receive. We'd especially like to see more presenters from our home turf. There's a lot of excellent work being done in Australia and the AUUG conference is an ideal way of getting exposure... Perhaps you'll think about sending in your abstract when you see our upcoming call for papers.

Speaking of next year, we have decided to run a "themeless" conference. While we have always encouraged material covering the gamut of the open systems world, we felt that choosing a topic made some people believe they were constrained. Conversely, AUUG is now known and respected widely enough to attract papers on reputation.

With all the fuss about the conference, it's easy to lose sight of the exhibition which generates the majority of funding for the entire event. The most notable success story this year was the conference network, boasting a 2 Mbit link to the USA and a 768 Kbit link to AARNet direct from the exhibition hall. Most of the vendors participated in connecting to the our network, it was heavily used by presenters and the sessions in the main auditorium were broadcast (video and audio) back onto the Internet. As you can imagine, there was an incredible amount of hard work behind the scenes to create and support a network of this size, and I'd like to particularly thank all the people from Access One for their professional and superlative efforts.

The web room turned out to be a great crowd-pleaser, and is an idea I think we'll repeat in future.

Unfortunately the terminal room, providing remote access and email facilities was plagued with problems. I'd like to apologise to anyone who was inconvenienced by this downtime. The root of all our woes was simply that there was not physically enough time to get a stable environment configured for the limited hardware resources we had available. Next

year we plan to stick to the KISS principle and aim for a simplified but robust service.

---

*...we had to evacuate the official dinner...*

---

When people think back to AUUG 95, however, they will almost certainly remember the bomb scare before all else. Yes, we had to evacuate the official dinner to the hall next door, after a (spurious, as it turned out) threat was anonymously phoned in. Luckily, several bright sparks managed to bring along a supply of the (excellent) Charles Sturt University wine which, continuing the lucky streak, turned out not to be wired to explode! At least, not outside of our heads and not before the next morning.

Several people approached me and suggested that it was a marketing exercise! No, AUUG doesn't stoop to that kind of behaviour. Several others suggested—jokingly—that Bill Gates might be behind it all. How ridiculous! After all, what has Windows 95 got to fear from UNIX?

Just in passing, check out NetBSD at some point. Version 1.1 is just about to be released and this has to be the cleanest, best designed and just plain elegant UNIX-like system (after all, UNIX is a trademark) I've ever had the pleasure of using. It runs on a half dozen different architectures, including PCs, and it would be my operating system of choice in a perfect world. Best of all it's free. Watch out Microsoft!

As usual, along with the conference comes the AGM, which in the short time allocated can only pretend to address AUUG's doings, plans, finances and other business. As usual, it was scheduled before the dinner which at least gives everyone a reason to move through the agenda smartly; for those who couldn't make the meeting, the minutes will be published in AUUGN. Chatting with some people afterwards, I was a little disturbed to hear that they had some issues to discuss but didn't get the chance. Unfortunately, the AGM will always be constrained by time and structure, although I think we could make more discussion time available.

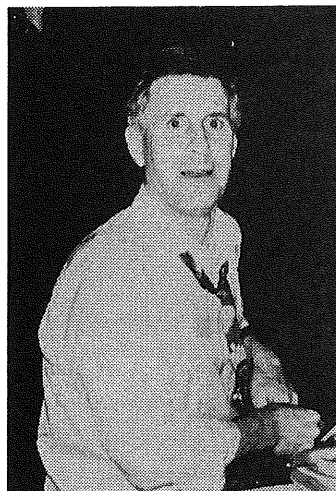
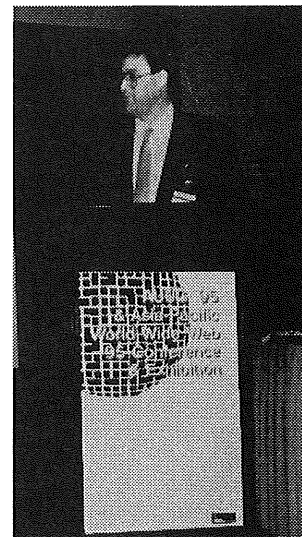
More importantly, you don't have to wait for the AGM to share an idea or take up a grievance with your committee. We meet six times a year, and most of our work is driven by member input. So please don't hesitate to pick up the phone or to drop us a line... we want to hear your thoughts on what we do, and what you'd like to see. I trust we'll be hearing from you. ❖

---

## Traditional fare: Conference photos

*Perpetrated by Adrian Booth  
whose comments accompany...*

Our new prez, Michael Paddon,  
at the AGM



A more relaxed Phil McCrea  
*(This is relaxed? You gotta wonder what being AUUG President does to you! – Ed.)*

PANIC strikes during  
the bomb scare!  
Brenda Lobb  
(UniForum NZ) & fellow  
diners help remove  
sensitive equipment...



Interview:

## Tim O'Reilly at AUUG '95

Frank Crawford <frank@ansto.gov.au>

During the AUUG'95 Conference I had the pleasure of interviewing Tim O'Reilly (TOR) for AUUGN. As one of the principles of the most active UNIX, and more recently Internet, publishing companies, their current and future plans are of interest to AUUG. Tim turned out to be friendly and very willing to answer questions about the future of O'Reilly and Associates (ORA).

**AUUGN:** What directions do you see ORA taking in the future?

**TOR:** ORA is undertaking activities on lots of different fronts, for example we are expanding our publishing "sideways", with plans to do more books on PC topics, such as the recent book on *PC Networking*, and other related interoperability issues. We also plan to publish books on such topics as publishing on the Web and CGI script development:

However, we plan to explore the area of the intersection of book publishing and online services. The *Global Network Navigator* was our first move in this area. It was based around a magazine model of publishing, and was a way of testing the model, as nothing like it had previously existed. We have now sold GNN, however, we are still interested in the area, both with GNN and with our new venture of *Songline Studio*.

A different model we are looking at is the online technical book area. We are looking at the online book as a user interface. From a historical perspective, the book has always been a part of the user interface, now, with new technologies and interfaces, this can be literally true. As an example of this new area we are working on a "toolbox", which could be considered an update of the *Power Users* book, but taken further. As an example, consider how you currently find free software. It will have a web interface, and should be out in the next six months.

As well, we are looking at the organisational structure of technical books. Many of these are really bodies of information referring to other such bodies. The second edition of *Graphical File Formats* is an example of this, it will be published online, with links to cheat sheets for the necessary information. A further example is as a

source code explorer, for example for Linux. You could create links from the book to the code.

ORA is changing its directions to also become more of a software company. Our first example was with *Internet in a Box* developed with Spry, and now with *WebSite*, and there are more to follow. Two of these are *WebBoard* a conferencing system, and *Cold Fusion* a database front end. The difference we bring to the software area is the "publishing house model", where we have outside developers—authors—and still with in-house technical support. We see the software business as an extension to our traditional publishing, not as a replacement. They deal with different markets and have different costs.

**AUUGN:** *WebSite* is a move away from UNIX as it is available as a Windows NT product. Are you "deserting" your original market?

**TOR:** Not at all. Today, operating systems matter less and less. To many people the Web is the platform and we are taking the chance to extend the technology to the new space of the Web, much as how CGI was a boost to perl. We see the goal of our business is to fill holes.

The Internet started out as peer-to-peer, however, at the moment people browse the Web with many low cost—that is, free—browsers, but there are not many servers, in comparison, so there is a desire for a low cost commercial product to fill the gap. We see this as good for the "Internet culture" of the sharing of information, but goes against the pricing model of such organisations as Netscape Corporation. Our distribution of *WebSite*, we believe, has caused cheaper access to the Web for all people.

We also see it important to move with our market. People are moving from UNIX to the Web and onto some future product. We see that the "network is the platform", and this is rapidly becoming true. Compare this with what Sun Microsystems has been saying for some time: "the network is the computer".

As with the Web, we are pursuing other areas. The Web is going to different markets and so are we. Today it is strong in the UNIX area and now moving to other areas. In the future more of "mix and match" of systems and products will be seen.

What we will see, is that the spirit of UNIX will migrate to other platforms.

**AUUGN:** Just to close on a more personal note, how do you find Australia?

**TOR:** I like Australia, I've have had one previous visit. It reminds me a lot of California, my home. The people here know how to enjoy life.❖



Right now, Herb Johnson is also doing laundry, making dinner and talking to prospective customers with WebSite.™

Meet Herb. He's busy promoting his company on the World Wide Web, one of the fastest-growing networks on the Internet. And thanks to WebSite, a new server software package for NT 3.5 and Windows 95 users, he's getting lots of other important things done, too.

So can you. Think of WebSite as your opportunity to become a publisher on the Internet. What you communicate—and who it's for—is up to you. Maybe a progress report for people in your company, a research draft for reviewers, even recipes to share with anyone on the Internet. With our password and security system, you can easily target your message to the group of people you're most interested in reaching.

Fortunately, as Herb well knows, you don't



have to be a rocket scientist to use WebSite. Its intuitive, graphical interface, online Help, and complete documentation will show you step-by-step how to publish on the Internet.

You simply won't find this power and flexibility in a Web publisher anywhere else.

WebSite's 32-bit server offers remote administration, multiple domain names, a graphical document manager, application or service capabilities, and the

ability to include data from other programs such as Excel in Web documents using CGI. For more details, send email to [website@ora.com](mailto:website@ora.com)

Or call Emma at woodsLane Pty. Ltd, on 1 800 803 443.

Look what it's doing for Herb.



**O'REILLY & ASSOCIATES, INC.**

Distributed in Australia by WoodsLane Pty. Ltd,

7/5 Vuko Place Warriewood NSW 2102 • Phone (02) 970 5111 • Fax (02) 970 5002

WebSite is a trademark of O'Reilly & Associates. All other brands are trademarks or registered trademarks of their respective holders.



Interview:

# John Lions

*Adrian Booth <booth\_adrian@tandem.com>*

John Lions was the "founding father" of UNIX in Australia, best remembered for his 1975 *A Commentary on the UNIX Operating System*. This book was the first describing the UNIX kernel. Distribution was later taken over by Bell Laboratories, who included the volume with every copy of UNIX they shipped for a number of years.

John Lions retired recently, an event which has inspired me to finally write up an e-mail interview I did with him oh so many moons ago.

John was being bombarded with questions from all over, including ones by Peter H. Salus (author of *A Quarter Century of UNIX* and Ronda Hauben (writing her own history of the Internet and of UNIX). Instead of answering each of my questions directly, in many cases John has simply supplied the answer he gave to a previous similar question. I have attempted to mark such questions with either a PS or RH prefix. I have also included some material from John's 1985 interview in UNIX Review.

**AB:** Could you describe your initial involvement with AUUG?

**JL:** I convened the first meeting. I was the first president, and am (so far) the only Honorary Life Member (this was written before Robert Elz became our second life member).

**AB:** Can you name other people active in AUUG/UNIX in the early days?

**JL:** The original group of enthusiasts was led by Ian Johnstone with Greg Rose, Chris Maltby and Greg James in hot pursuit. Other such as Peter Ivanov were not far behind.

**AB:** What other user groups have you been involved with?

**JL:** I initiated the Australian Computer Science Conferences, of which the seventeenth has just been held in Christchurch, New Zealand! Apart from this, AUUG is enough for me...

*John described some background to UNSW acquiring UNIX in his 1985 UNIX Review interview:*

**JL:** In 1974, [UNSW] decided to buy a Cyber 72... but since the Cyber only recognised User2000 terminals which already were completely obsolete, the university ended up buying some PDP-11s that could emulate User2000s... the Department of Computer Science was to have a share of one of the larger PDP-11/40s, but most of any given day it was to do User2000 stuff, leaving us to salvage whatever capacity was left over.

**PS:** When were you first exposed to UNIX?

**JL:** In 1974 when we saw the 1974 CACM (Communications of the ACM) paper, and were about to be given a PDP11/40, with all of three RK05 cartridge disk drives (2.5 megabytes each) and 208K bytes of core storage.

Our license agreement was signed in mid-December, 1974, and the Edition 5 tape and manuals arrived as a late Christmas present. When the system arrived, Ian Johnstone, Peter Ivanov, John Wainwright and Greg James put it up, tried it, and decided that this was the way to go...

[Ian Johnstone with assistance from others wrote a User2000 emulator] that ran under UNIX. That became the first application of UNIX to be written in Australia...this exercise proved to be extremely important. With the PDP-11 completely to ourselves, we most likely would have run vanilla UNIX on it and been happy. But because we had to provide the User2000 emulator, we had to learn a lot about the system and pay a lot of attention to performance issues. We needed help, but we couldn't get any from outside sources. So we ended up generating our own expertise...

**PS:** What motivated you to write and annotate the code?

**JL:** I was teaching Operating Systems. I was in (friendly) competition with a colleague (Ken Robinson) who was teaching Compilers by getting students to write real compilers... so a code reading exercise seemed like a good idea. Also our UNIX license did not explicitly forbid such activity. Later license agreements were modified to change this, but somehow I never learned of this (officially)...

After I started distributing copies of my notes on UNIX (Source Code and Commentary), I sent more than two hundred copies to BTL. One night (sometime in 1978?), I had a phone call from Doug McIlroy saying BTL would like to assume responsibility for distributing these documents, and would I agree? I did. It saved me much work.

*Continued on page 18*

Viewpoint:

# AUUG95-APWWW95 wrapup

Frank Crawford <frank@ansto.gov.au>

Well, AUUG95/APWWW95 has come and gone, and finally we can see if all the planning, claims and marketing hype has come true. The answer is yes, both the conference and exhibition lived up to expectations.

Starting with the conference with over 800 delegates, it was by far the largest conference that AUUG has staged and the largest conference on the Internet and the World-Wide-Web ever held in Australia. In fact the actual breakdown was 742 delegates to the conference, 333 tutorial attendees and 842 attendees in all (there were obviously a large overlap of people attending both the tutorials and conference).

While the numbers were down on some initial predictions of over 1000 delegates it is nearly 200 more than any previous AUUG conference. Given the heavy involvement by AUUG members in the Internet in Australia, it is not surprising there was a big overlap between those attending the AUUG conference, and those attending the Asia Pacific World Wide Web conference hosted by Charles Sturt University.

The delegates attending the conference came not only from Australia, but from around the world with over 50 people coming from Europe, America, Asia and the Pacific. Some of these visitors were presenters, but many were delegates here to see the latest in both WWW and Unix technologies.

A first at this conference was the staging of two days of tutorials, on the Sunday and Monday prior to the conference, previously they had been restricted to only one day. These tutorials ranged from introductory courses on free versions of Unix to advanced topics in network design and implementation, from configuration of complex system software or WWW security to the ethics of the World-Wide-Web.

If the tutorials covered a wide range of topics, then the three days of the conference took that even further. There were keynote speakers from such places as CERN, where the WWW protocol was first developed, to NCSA, where Mosaic was developed, from those involved in the development of the future network technologies, to the most innovative book publishers in business today. As well, there were experts from

Australia, including Geoff Huston, the manager of Telstra Internet, to David Green from ANU and the Australian National Library.

As well as the keynote speakers there were presentations on such topics as Unix Developments, Internet Applications, the use of the Internet for Business, Developments of the WWW and the use of the WWW for Education. In all there were 72 presentations, demonstrating both current and future technologies in both Unix, networking and the WWW. Many of these presentations are available in the conference proceedings, which are available from the AUUG secretariat, and at some time in the future will be available for online.

For those who weren't able to attend in person, many of these presentations were broadcast directly over the Internet, using the MBone. While no accurate count was kept of these viewers, there were times when an additional 25-30 people were known to be logged on via this mechanism.

While much of the conference was about the future of the Internet, the current network technology played a crucial role in both the conference and the exhibition. Darling Harbour was linked to the Access One (the Internet Services Provider selected for the conference) hub by a 2 Megabit link which was then connected directly to the USA over a 2 Megabit link, and to AARNet by a 768 Kilobit link. For the duration of the conference the best connected place in Australia was at conference venue, and many experienced Australian users began to understand what the Internet feels like to many US users, high speed links, no delays and instant access to the world.

Aside from just connectivity, which reached into all the conference rooms, a first for Darling Harbour, there were many services provided within the exhibition area. One of those services was a Web proxy, i.e. a service which stores recently accessed web pages, etc, to allow faster response for the next time. This service recorded 95354 operations of which 93889 were GET commands. This translates to one transaction per second for the duration of the conference!

Another interesting statistic from the network was the rate that it was rolled out. The entire conference and exhibition network was rolled out in a period of only three days, connecting all the conference centre, an email centre of twenty X-terminals, a Web centre of more than ten of Apple Power Macintoshes and over 230 identifiable hosts, on one of the fastest Internet connections ever seen in Australia. This network

*Continued on page 21*

## Overview:

# AUUG95 Network

Frank Crawford <frank@ansto.gov.au>

There is considerable hype in the media about computer networks, the Internet and the World-Wide Web, but there is generally no way for the general public to experiment with these facilities, without signing up and paying an Internet Service Provider (ISP).

One venue that offered people the opportunity was at the AUUG95/APWWW95 Conference and Exhibition. In fact, the conference and exhibition had one of the biggest networks ever setup at an Australian conference.

Starting with the exhibition area, all stands were networked by Cisco Systems Australia (one of the worlds leading suppliers of network equipment), allowing access to other stands or to the Internet directly. This allowed vendors to demonstrate their connectivity both to a standard ethernet-based local area network and to the Internet, and further allowed any vendor to run applications on other cooperating stands.

Further, having the connection directly to the Internet permitted other vendors to directly access their headquarters and display a wider range of demonstrations than would normally be available for an exhibitor's demonstration. While this has been available in a more limited way at previous conferences, this is the first time it involved all stands and vendors.

The Exhibition Centre was connected to the rest of the world via a 2 Megabit link from Darling Harbour to AccessOne, the ISP providing the network connection for the conference. This connection was configured to supply the bandwidth on demand, with additional connections automatically coming online as required by the demand within Darling Harbour, and then being freed when no longer needed.

The connection from Darling Harbour to AccessOne was only a small step in the link to the Internet, from there a direct circuit to the USA provided up to 2 Megabits. This circuit was dedicated to the Conference and Exhibition, and when one considers that only a little over one year ago the link for the whole of Australia was only 1.5 Megabits, one begins to understand the size of this connection.

The other important link from Darling Harbour was the connection into AARNet by Telstra Internet Services.

However, this network is only the infrastructure on which more interesting activities were implemented. Within the Exhibition area a section was set aside for members of the public to come and explore the Internet, and in particular, the World-Wide Web. There were facilities to "surf the net", by using Web browsers such as Netscape, as well as the ability to allow them to set up their own temporary "home-page" on the Web.

Charles Sturt University arranged for a number of Power Macintoshes to be configured with multimedia production equipment and had an experienced team on hand to assist people in capturing images, writing HTML (the language of the Web) and setting up links, so they could temporarily have a presence on the Web. As the Exhibition area was connected to the Internet, these "web-pages" were also accessible from anywhere in the world during the conference.

The Conference itself made extensive use of Internet facilities. Many of the presentations used data taken directly from the Internet, often allowing the presenters to run their presentations from their home machines. Many of the tutorials presented at the conference were on Internet and Web related topics, and the instructors (acknowledged experts from around the world) used the Conference's Internet connection to demonstrate and display the basis of their tutorials.

Finally, information not only flowed into the Conference, but was also transmitted. Some parts of the Conference were broadcast directly to the Internet using the "mbone". This is a facility that allows appropriately configured workstations on the Internet to display images and sound that has been transmitted from some other site.

The conferences theme being "The Internet Means Bu\$iness", it became a showcase of the various technologies and facilities that are available on the Internet. It also gave the public a chance to see and play with these facilities, so as to provide a better understanding of what they do and how they may be used in the future. ❖

Background:

# The Cryptography Debate in Australia

Chris Maltby <chris@sw.oz.au>

Recent weeks have seen a burst of discussion on the future of encryption technology in Australia, following the presentation of a paper by Steve Orłowski of the Attorney-General's Department to the Cryptography Policy and Algorithms Conference this July ([http://commerce.anu.edu.au/comm/staff/RogerC/Info\\_Infrastructure/Orłowski.html](http://commerce.anu.edu.au/comm/staff/RogerC/Info_Infrastructure/Orłowski.html)).

In spite of the usual disclaimer that his views do not necessarily represent those of the Australian Government, the paper has been received as an attempt to express thinking in Commonwealth law enforcement circles on the challenges presented by widespread availability of strong encryption. In particular, the impact on the ability to recover information from telecommunications interception (TI).

Balanced against this is the more widespread requirement for security and user confidence in the emerging technologies for electronic commerce, which depend, to a lesser or greater extent, on strong encryption. The conflict between satisfying these objectives is clear. Products using strong cryptography must be widespread before companies and individuals will adopt them for their transactions, yet if they are easily obtainable, criminals can be expected to use them.

To address this conflict Orłowski puts forward the a concept he calls "differential key escrow" in which encryption products would be divided into two grades; those which can withstand a "general but not a sophisticated attack", and those which are more secure. Products using higher grade encryption would be required to submit to key escrow, where keys could be retrieved (presumably requiring a warrant) from some escrow agency, dubbed a "data recovery centre".

Orłowski contends that the lower grade products would be sufficient to meet the needs of the "majority of users", with the few remaining users (he envisages large financial institutions and government agencies) accepting the requirement for key escrow. He notes "criminals who purchased the higher level encryption products would immediately attract attention to themselves".

Leaving aside the obvious civil liberties issues (summarised as "make encryption illegal and only criminals will have encryption"), there are major technical obstacles to this approach. First, it assumes that the distribution of strong encryption can somehow be controlled using Australian legislation. As well, it presumes that law enforcement agencies can distinguish between different encryption methods by routine examination of the encrypted data. Both of these assumptions would seem to be questionable.

The widespread availability of the PGP (Pretty Good Privacy) shareware package demonstrates that it would be very difficult to restrict the distribution of such technology within Australia given the existence on the Internet of hosts beyond the jurisdiction of Australian law. The only available legislative approach is to make ownership or use of such technology illegal, assuming that a suitable definition could be made, and that such legislation did not conflict with the Constitution.

The second assumption is perhaps more difficult. The only way to determine if any given message is using strong or weak encryption is to attempt a decryption of it. To catch the criminals using strong encryption would require an mammoth continuous decryption effort by law enforcement agencies. Even this approach presumes that the message is encrypted using some straightforward (if strong) cypher. A message which has been disguised to look, for example, like a program binary would not be detected. In any case, this would represent a massive invasion of privacy, which would only increase the demand for stronger forms of encryption.

Finally, it is worth questioning Orłowski's contention on the acceptance of weak encryption by the majority of users. This assumes that only the law enforcement and intelligence agencies would possess the capability to perform routine decryption. Even if this could be assured, it could safely be presumed that most governments would have the capability. This may not be acceptable either to corporations conducting business outside Australia or to individuals. This would represent a major disadvantage to Australia's economic and human rights interests.

Readers are urged to consider and respond to Orłowski and also to the July 7th Consultation Paper on the Regulation of On-Line Information Services ([http://www.dca.gov.au/paper\\_2.html](http://www.dca.gov.au/paper_2.html)) issued by the Department of Communications and Attorney General's Computer Bulletin Board Task Force. ❖

Background:

# Netscape Break-in

David Purdue <David.Purdue@aus.sun.com>

On the 15th of August 1995, Damien Doligez decrypted a World Wide Web session that was encrypted using Netscape's Secure Socket Layer (SSL) protocol.

Doligez, a PhD student at the French National Institute for Research in Computer Science in Le Chesnay, France, used 120 assorted workstations to decrypt the message, and the task took 8 days to run.

SSL is the protocol used by Netscape's WWW browser to encrypt information sent over the Web to WWW servers. This, in theory, makes it safe to, for example, send your credit card number to a record shop in the US so you can order some cheap CDs.

If there is to be electronic commerce on the Internet, then it will be based on the existence of a safe and secure way to send money over the net. Netscape proposed SSL as the mechanism to allow this, and published the specification of SSL so that anyone could implement a Web browser or Web server that used SSL.

So how was it possible for Doligez to break this encryption?

In the case of SSL, it was not just possible, but relatively trivial. And the reason for that boils down to US export regulations.

You see, it is illegal to export "strong" cryptographic software from the US, and so the developers of Netscape were restricted in how strong they could make the algorithm in SSL. So SSL uses a "session key" in its encryption that is only 40 bits long. Thus there are only 2 to the power of 40 different session keys. That is 1,099,511,627,776 keys.

One way to break the encryption of a SSL encrypted session is to try each of these 1,099,511,627,776 keys. This "brute force" method is essentially how Doligez broke the code.

Now 1,099,511,627,776 may look like a big number, but it is really not all that many when considered against the speed of modern computers. In the past Netscape has said that to crack a message encrypted with a 40 bit key would take an attacker about 64 MIPS-years, which is roughly the equivalent of a Pentium PC running solidly for 6 months.

However, the problem is very easy to split up. If you had two Pentium PC's, one could work on the odd keys and one could work on the even keys and the job would take only 3 months. The problem can be split to use whatever computing resources are available, and make maximum use of the parallelism obtained.

The idea of choosing a 40 bit key was that it should be possible for law enforcement agencies to decrypt messages if needed, but too difficult for an amateur to do so. Doligez has shown that the choice of 40 bit keys falls down in the second point.

You may think that there is nothing to worry about. After all, there is such a large volume of traffic on the Internet that the chances of someone capturing the session where you transmit your credit card details is very small. And you probably reveal your credit card details to shop assistants every day without a second thought.

Still, it would be nice to know that when you do send your credit card details, they are sent in a secure manner. It is a similar precaution to destroying the carbons from the credit card slip.

Currently the 40 bit key in SSL does not provide that kind of protection. It is not that the SSL protocol is broken. The problem is political rather than technical. A longer key could easily be used, and the protocol allows that, but the US government would prohibit the export of a program that used the longer key.

Each extra bit of key-length doubles the number of keys available, and so doubles the length of time it would take to do a brute force search of the keys.

The Data Encryption Standard (DES) uses a 56-bit key, which gives it 65536 times the number of keys of 40-bit SSL. DES is still considered sufficient security for commercial applications.

The way for software companies to provide a reasonable level of security to customers outside the US seems pretty obvious: do not develop cryptographic software within the United States!

Australia has a number of experts in cryptography. But the algorithms used in SSL are not rocket science - they are well known and widely published. SSL could be implemented by any competent programmer.

US based software vendors should realise that if they want to include cryptographic security features in their products, and sell them to an international market, then they must either develop the products outside the US; or at the very least provide hooks in their software so that third party system integrators can incorporate strong cryptographic software.❖

## Opinion: Software Minimalism

Jamie Honan <jhonan@mpx.com.au>

How much longer can this software bloat go on? I'm talking about the ability of applications to take up more and more megabytes of ram and disk space.

Driven by the perceived need to add feature after feature to each version of software, and never to drop a feature, applications have become monsters, dwarfing the operating systems and basic system utilities.

Applications tend towards 'complete environments', whereby users do not have to switch out of the application to perform a function.

The antithesis of this software bloat is software minimalism. The best proponent of minimalism I've seen is 'Chuck' Moore, the inventor of the Forth language.

Several years ago at a Sydney seminar, Moore demonstrated a computer built around a early version of a 'Forth' chip. Moore was appalled at the tendency to waste computer cycles. For example, he found it hard to believe that IBM compatible PC's came with three or more microprocessors. As for using more than 16 bit addressing, Moore described his complete Printed Circuit Board design system which ran in 16 KiloBytes. Mind you, it wasn't a CAD system as most people would understand it, the graphics being done in text mode.

Moore's system had other uniquely minimalist features. He eschewed the traditional keyboard, opting for a system with only three buttons. For text entry, the buttons could be pressed together (chording), or one button produced a menu of alphabetical characters, the other buttons could be used to scroll to the required character.

Moore was so minimalist in his design principles that he rejected the use of 'pull-up' resistors in his computer. His idea was to use software to drive the input lines as output lines momentarily, then quickly detect if the input state had changed. If the state changed, it was probably floating. Another feature of Moore's computer was its lack of serial device. Software 'bit-banging' was used to generate and detect asynchronous serial io.

Lest I have given you the idea that Charles Moore was a crank who could be safely ignored, I should point out that the principles in the 'Forth' language have found very widespread usage. The most notable would probably be the 'PostScript' language used in high end printers and typesetting machines. Many of the ideas in PostScript are directly taken from Forth.

Probably the worst examples of software bloat are found in programs which are classed as GUI - having a graphical user interface.

It seems programs in this environment require large resources to run. The Microsoft Windows SDK includes a sample 'Hello World' program, which runs to several hundred lines of source code!

The X Window System also requires large resources. Linux is described as running very well in 4 Megabytes, but to run the X Window system in Linux effectively requires 8 megabytes. I've often asked myself whether there is something intrinsic in these display systems that makes them so huge, or are they badly designed systems?

An example of a very elegant system for writing GUI programs is the scripting language Tcl/Tk. Tcl/Tk has just recently been made available for MS-Windows and the Mac systems as well as its traditional XWindow platforms.

With Tcl/Tk it is possible to write applications in small amounts of code. There is a penalty for this elegance, the supporting system requirements are still quite large.

Tcl/Tk has its detractors. The language is fully interpreted, and runs computationally intensive code very slowly (something the core language Tcl was never designed for). Tcl is sufficiently different to look at that a number of people feel it's a 'side-line' language.

Language jihad is the term used on Usenet for the 'my language is better than your language' argument. It's one of those pointless arguments that arouses so much passion.

As much as they'd like to, Tcl's critics cannot ignore the obvious merit of writing GUI scripts in Tk. So in order to harness Tk's features, they leverage on one of Tcl/Tk's other strengths, its ability to be embedded in other programs.

Beta versions of the GNU extension language, a LISP derivative called Scheme, included the full Tk system. (Amusingly enough, Scheme came about because some people wanted the language design principles of

LISP in an implementation that didn't include the kitchen sink).

If you didn't like Tk as part of your Scheme, you could always include Scheme as part of your Tk! One of the extensions to Tk is a Scheme interpreter.

Of course, if following the gourd is not to your taste you can follow the sandal. The Perl language is a great hit because of it's ability to perform Unix system administration tasks, and its syntax is very easy to come to terms with for a new user. Tk is also available as a Perl-loadable module!

Are these software monsters really what users want, or will we collectively demand software parsimony? Will users ever turn their backs on multi-megabyte applications?❖

*Continued from page 16*

included a 500 metre run of fibre-optics cable from the exhibition centre to the conference centre and other connections around parts of the conference centre rarely seen. As well there were network connections available to each of the 75 stands. Nearly as impressive was that this network was torn down in less than a day at the end of the conference.

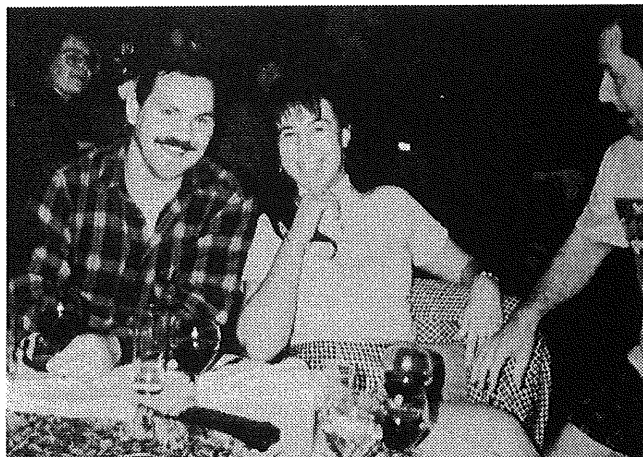
Like the conference, the exhibition was the biggest ever staged by AUUG and probably the biggest exhibition of Open Systems and Internet Technology ever staged in Australia. There were 75 exhibitors ranging from software to hardware vendors, ISPs and computer book publishers. This exhibition attracted nearly 5000 visitors, including nearly 4000 who only attended the exhibition.

Finally, AUUG95/APWWW95 was also the centre of both major demonstrations and announcements, with an impressive AV presentation put together during the conference by Sun Microsystems and the other vendors within the exhibition, and finally, the announcement by Novell, SCO and HP about their new alliance for the development of Unix.

In summary, this years AUUG95/APWWW95 Conference and Exhibition was bigger and better than any previous one, while work on next years conference has already begun and should be better still.❖

Just when you thought it was safe...

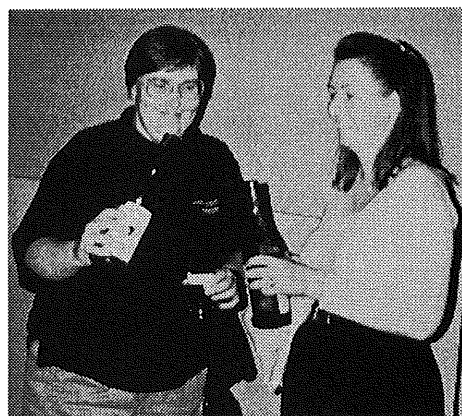
## More photos



Our new business manager (Liz Egan) with two exhibitionist- ah ... exhibitors. Note Liz's bandaged left forefinger, obtained in the course of duty while setting up the AUUG stand!



Jagoda "I used to edit AUUGN but I'm much better now" Crawford, with tireless AUUGN booster, Frank Crawford. The way they're carrying on you'd think they were married or somethin'!



Two peas in a pod ... Belinda Parsons and Angela Booth



# UNIX Tricks & Traps

Edited by Janet  
 Jackson<janet@dialix.oz.au>  
 Phone/Fax (09) 295 4753

In this issue we have some letters to the editor, including information about the very handy utility **xargs**, and a couple of articles about tricks with the shell: one for parsing strings and one for timing out a process.

Thanks to everyone who has sent in things for this column. I'm nearly at the bottom of the barrel now, so more please!

How about a brief note about the little feature that most reduces your stress levels? For me, it's probably combining C-shell history substitution with backticks, as in:

```
% grep func *.c
# find all C files that use
# some function
      (reams of output)
% grep -l !*
# show only the filenames
      (still lots of filenames)
% vi '!!'
# edit them all
```

One could, of course, skip straight to the third command with

```
% vi `grep -l func *.c`
```

... but I tend to make it up as I go along.

We all use Unix in different ways, so why not share your favourite trick? It doesn't have to be original, just useful. ❖

## Correspondence

Tricks & Traps is actually getting letters to the editor! I have had one letter suggesting a topic, and two in response to Graham Jenkins' article about **grepdir** (AUUGN Vol 16 No 3).

From Dinesh Pullat <dinesh@cs.uow.edu.au>

I'm a Master's student currently working on a project involving TCP/IP networking. It would be very helpful if the column on UNIX Tricks & Traps could also have some tips on the subject of networking. As you know, tips on UNI are found in some good books too but similar stuff on TCP/IP and UNI is almost impossible to find.

From David Purdue <David.Purdue@aus.sun.com>

I read Graham Jenkins' **grepdir** program with interest, and I thought it was a good and a clear example of shell programming.

However, the script is somewhat redundant, as the same effect can easily be achieved on the command line using the **xargs(1)** command, for example:

```
find /usr/spool/news -type f -print | xargs egrep -i ultra-widget
xargs(1), in the tradition of the best UNIX commands, is quite simple, yet very powerful when used in pipelines. It constructs commands from its arguments combined with arguments read from standard input. In the example above it would construct commands such as:
```

```
egrep -i ultra-widget /usr/spool/news/aus/archives/1 \
  /usr/spool/news/aus/archives/2 ...
egrep -i ultra-widget /usr/spool/news/aus/general/112 ...
...
```

and so on.

**xargs** is clever enough not to put too many arguments on the one command line, so you do not get "arg list too long" messages. It is also more efficient than using a "-exec" flag in find, as "-exec" will start a process for each file it finds, whereas **xargs** will use as few processes as possible without the lines getting too long.

Another example of using **xargs**—change the ownership of all files in a particular tree:

```
find . -print | xargs chown fred
```

There are too many UNIX utilities for everyone to remember all of them, but **xargs** is worth having in the mental toolkit.

From Tim Cook <wwcoot@lpdwhy.bhp.com.au>

Just thought I would be a smarty-pants and point out that Graham Jenkins' **grepdir** script isn't necessary. To perform the type of search he uses as illustration, simply use the following:

```
find /usr/spool/news -type f -print | xargs grep -i ultra-widget
```

The **xargs(1)** manual page should show how the magic in this is performed.



I think his `cmpdir` script might also be obviated by the `-r` flag on most modern versions of `diff(1)`.

There are plenty of ways to skin a cat with UNIX...

## Timing out a process from the shell

Glenn Huxtable <glenn@fs.com.au>

*I simply want to write a script that runs a student's program under some rigid time limit, and kills it if it exceeds that time limit. I have tried about a million variations of a shell script that looks like:*

```
limit cputime 60
student_program > output
```

*but the limit is simply ignored. Any suggestions?*

Yes, there are a few things that occur to me immediately (i.e., I haven't thought very deeply about this problem). Firstly, 60 seconds CPU time can be a `_long_` time in real time, depending on what the program is doing. Are you sure you want CPU time, or do you really want real time?

If you do really want CPU time, then setting `cputime` should achieve that (as long as the student project doesn't ignore `SIGALRM(?)`); however, you are also setting the `cpu` time limit on your shell, which may not be quite what you want. Something like the following would be better:

```
(limit cputime 60 ; exec student_program > output)
```

This would create a subshell that sets its own CPU limit to 60 seconds then execs the student program (overlays the subshell with the student program). The student program should inherit the time limit. (The `exec` isn't strictly necessary, but it will reduce the number of processes.)

If you want a real time limit, then I think you will have to resort to something less trivial:

```
#!/bin/sh

# background the student process, save its PID in CPID
student_program > output &
CPID=$!

# background a sleeper process, save its PID in SPID
(sleep 10; kill -9 $CPID) &
SPID=$!

# wait for the student process to terminate
wait $CPID
STATUS=$?
echo STATUS = $STATUS

# kill the sleeper process (if it's not already dead)
kill -9 $SPID 2>&1 > /dev/null
```

This backgrounds the student program, saves the process ID in `CPID`, then backgrounds a sleeper process (whose responsibility it is to kill the student process after 10 seconds real time -- adjust this as you want) and records the sleeper process ID in `SPID`. It then waits for the child process `CPID` to return (either naturally, or having been killed) and then kills the sleeper process (in case the child process terminated normally).

You `_might_` be able to use the exit status of the child process, `$STATUS`, to determine whether the process terminated normally (exit 0) or was killed, but that relies on your students using exit statuses correctly (and not many people do).

You could probably cast this into a simple shell procedure to make calling it as trivial as your original example.

There may be simpler ways to do this, but I can't think of any off the top of my head.

## Using IFS to make parsing easier

Peter Chubb <peterc@bookworm.sw.oz.au>

Sometimes when you want to parse something (we'll use the value of \$PATH as an example) you don't want to have to fire up `awk` or `sed` to split it up. In these cases IFS is very useful.

Here's an example -- a `which` program. The system-supplied `which` program prints out only the first instance of an executable program in the path. This one prints out all of them.

Setting IFS to ':' allows `dir` to take the value of each path component in turn. Then checking to see if `$dir/$1` is an executable file is trivial. (You can't just test for executability, 'coz searchable directories are marked executable.)

```
#!/bin/sh

[ $# -ne 1 ] && { echo >&2 "Usage: `basename $0` program"; exit 1 }

IFS=':'
for dir in $PATH
do
  [ -f $dir/$1 -a -x $dir/$1 ] && echo $dir/$1
done
```

❖

*John Lions (continued from page 15)*

At the beginning of 1978, when I was starting to wonder what to do for my first sabbatical leave, I had another late night call—this time from Berkeley Tague enquiring whether I might be willing to visit BTL—another easy decision.

*Berkeley Tague takes up the story:*

**BT:** Research asked me to invite [John] to work with the USG. He had written his wonderful book on the UNIX(r) system early in the game and we had found it most useful. We agreed to publish and distribute it and we wanted to help John continue his work as one of the UNIX apostles in Europe and Australia. He wanted to come to Murray Hill for his sabbatical so it was a win/win situation. He spent two or three summers at Bell Labs over the years and supplied us with many of his graduate students for sabbaticals and permanent employment. For me personally, he became not only a professional contributor, but a good friend. A truly delightful gentleman!

We offered [the Commentary] as a part of the documentation package for those who wanted to understand or modify the Unix(r) source that the USG shipped. It was very useful as an introduction even though the code no longer matched the book. It outlined the conceptual architecture very clearly in the early short form of the system before it accreted all the minor changes and feature additions that disguised

the original clarity of its structure. All new people were given a copy when they joined the USG and I suspect most development groups did the same.

**RH:** Do you have any idea why UNIX was so widely adopted at other Australian Universities?

**JL:** We spread the news evangelically... We were very anxious to share our accumulated knowledge and to experiment... and we wanted to share it with others. We were having fun!

**AB:** What have been the professional highlights of your career?

**JL:** For myself, three sabbaticals at Bell Laboratories have been highlights. For my students, opportunities arose for employment at the Laboratories. Andrew Hume is still there today.

**AB:** What do you think of AUUG today compared to its origins?

**JL:** AUUG has grown quite beyond my expectations in the late '70's. I guess I am proud to have been involved with such a successful organisation.

**AB:** Any closing remarks?

**JL:** I wish AUUG well for the future. I am sure it will continue to be successful.❖

# AUUG BOOK CLUB & PRENTICE HALL AUSTRALIA

## 20% DISCOUNT TO AUUG MEMBERS

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

**The Magic Garden Explained  
Solutions Manual**  
ISBN: 0132075563 Paper \$24.95\*

**OZ Internet:  
A Guide to Connecting to the Internet in Australia**  
ISBN: 0724809058 Paper \$34.95\*

\*Deduct 20% from listed retail price

Name: \_\_\_\_\_ Position: \_\_\_\_\_

Organisation: \_\_\_\_\_

Address: \_\_\_\_\_

Telephone: \_\_\_\_\_

**PLEASE SEND MY  
BOOK/S ON  
30-DAY APPROVAL**

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

Charge to me OR  Company purchase Order No. \_\_\_\_\_

Please charge my:  Bankcard  Visa  MasterCard

Expiry Date: \_\_\_\_\_ Credit Card No:

Signature: \_\_\_\_\_

Please add my name to your mailing list in the following areas: (please list subject areas)

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

OR  Use our PHONE SERVICE by calling  
Liz Guthrie SYDNEY (02) 9907 5648



**PRENTICE HALL AUSTRALIA**

7 Grosvenor Place, Brookvale NSW 2100

Tel: (02) 9939 1333 Fax: (02) 9905 7934 A.C.N. 000 383 406



**Prentice Hall Pty. Ltd.**

7 Grosvenor Place, Brookvale NSW 2100.

Tel: (02) 9939 1333 Fax: (02) 9905 7934

A V I A C O M C O M P A N Y

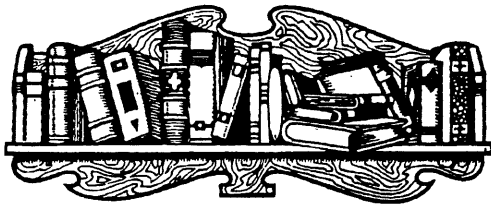
# Book Reviews

Frank Crawford <frank@ansto.gov.au>

Welcome to another important and interesting book review section, this time containing a number of important releases and one that is important to me personally.

We have a number of releases related to the Internet, from the history to connecting and onto how to secure it, as well, there are also releases related to such things as bash, perl, UNIX system programming and UNIX internals; in fact, one of the widest ranges we have had for a while.

As many of you have noticed, we currently have lots of books coming for review. The current practice is to post a note to the mailing list [auug-books@ansto.gov.au](mailto:auug-books@ansto.gov.au) and the newsgroup [aus.org.auug](mailto: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 9273, with your contact details and preferences. ❖



## Internet Security Firewalls

by Brent Chapman with Elizabeth Zwicky  
O'Reilly & Associates, Inc. 1995  
ISBN 1-56592-124-0  
Reviewed by Warren Tomey ADFA  
<wkt@cs.adfa.oz.au>

I asked to review this new Nutshell book as I already had Cheswick & Bellovin's book 'Firewalls and Internet Security' from Addison-Wesley, and wanted to compare the two.

I have always found the Nutshell books by O'Reilly to be excellent and informative, and 'Firewalls and Internet Security' is no exception. The book is divided into several parts, and I will cover each part in turn.

Part 1 (chapters 1 to 4) gives an excellent coverage of the risks any organisation takes in connecting to the Internet. It outlines the functions of the basic Internet protocols and the services that use these protocols to perform their work.

An organisation must create a suitable and useful network security policy. An Internet firewall provides a set of mechanisms that can help improve security, but it cannot cure all security problems. A firewall does not set security policy – it implements security policy. The sorts of security mechanisms that a firewall can provide are described and, just as importantly, so are the mechanisms a firewall cannot provide.

The general architectures of firewall systems are enumerated. This clearly explains how the components of a firewall system are integrated to form a complete architecture. The book indicates possible alternatives to the main architectures given, and in particular it indicates which architectures provide little or no increase in network security, and why.

Part 2 (chapters 5 to 10) show how to configure the components of a firewall system: the bastion host, network routers and packet filtering, securing the services provided by the organisation, and providing secure proxy services.

A bastion host is intended to be one of (if not the only) machines in an organisation which is visible to the Internet. Where possible, the number of services it provides should be minimal and secure. The book explains what services aren't required, how to disable them, and how to replace the remaining services with secure ones. A lot of time is spent describing how to set the bastion up securely, and how to maintain its security.

One thing I have yet to see in any Internet security book is a good discussion on what to do with the system logs. We are told to read them, but who wants to read hundreds of kilobytes of logs a day. I suspect that, live me, other sysadmins have written their own scripts to find potential and actual security breaches (and attempted breaches). It's time to start sharing these around!

The bastion host usually acts as a proxy for clients inside the organisation. For some services, such as mail and the Web, existing servers already provide proxy service. Others can be modified using packages such as SOCKS and the TIS Firewalls Toolkit to act as proxies. Unfortunately, some services such as X Windows are not easily proxied.

One area that Brent's book covers better than Cheswick & Bellovin's is packet filtering for the routers in a firewall system. This is tricky to get right and, if not done correctly, the filtering may still allow the wrong packets through. The reader is walked through the packet types of IP, TCP, UDP, ICMP, RCP, and their use by Internet services. The book describes what packets to filter, why, what filtering pitfalls can occur and how to avoid them.

Part 3 (chapters 11 and 12) describes the sorts of actions a sysadmin needs to take to maintain the

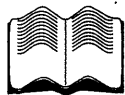
firewall system. This falls into three categories: housekeeping chores (backups, account management and disk management), monitoring the system's operation, and keeping the system up to date.

If a security breach (or an attempted breach) is monitored, the sort of steps that can be taken are outlined. Again, the actions to be taken form part of the organisation's security policy, but some actions (documenting the incident) must be done in any case.

The appendices describe the available programs, packages, organisations (CERT, AUSCERT etc.), mailing lists, web pages, papers and books that are useful in the construction and maintenance of an Internet firewall. Each resource has a short description of its location and its worth.

If you run an Internet-connected network, or are contemplating connecting your network to the Internet, buy this book. Not only does this cover Internet firewalls, their design and the tools used to create them, but it also outlines the sorts of policy decisions you need to make about your Internet connection. And if your machines are successfully attacked, the book tells you what to do and what you should already have done in preparation for such an attack.

Any Internet-connected network needs a book like this. Congratulations to Brent and Elizabeth for writing it.❖



## Casting the Net: From ARPANET to INTERNET and beyond...

by Peter H. Salus  
Addison-Wesley 1995  
297 pages  
ISBN 0-201-87674-4

*Reviewed by Michael Usher*  
University of Sydney  
<miku@extro.ucc.su.oz.au>

When I asked to review "Casting the Net", I had previously only skimmed through Peter Salus' earlier book "A Quarter Century of UNIX". The back-cover blurb tells me Salus has written over 100 other books, articles and reviews, but I haven't seen them and won't comment. The back-cover also tells me that this book is 'the first full-length history of ... the Matrix...'. Well, it may be the first, but I'm not so sure about 'full-length'.

One notices quickly that there is very little original material. Only 160 of the 310 pages have been written by Salus himself. The book is divided into five parts covering different periods of network development. They correspond roughly to conception (8pp), initial research (53pp), the ARPANET NCP period (57pp), the ARPANET TCP/IP period (22pp) and the post-NSFNET explosion (20pp). This would seem to be a logical progression, but the more recent history actually gets a lighter treatment, and the later chapters are more organized by the development of applications and services rather than a purely historical order.

The remaining pages of the book are made up with a selection of maps, RFCs and poems. In particular, we receive the full text of RFCs 1121, 968, 1097, 527, 1149, 1216, 1217, 1313, V. Cerf's "Requiem for the ARPANET", 1438, 1605, 1606, Leonard Kleinrock's "THE BIG BANG!", and RFC 1607! Essentially, he reprints all the April 1 material. Many other RFCs are quoted in part throughout the book. The originals were often very funny, but the reprinting is as bad as reading out FAQs as part of a conference paper!

There are subtle warnings in the first few pages. Vinton G. Cerf's foreword reads as a short essay on past and future directions for the Internet, but seems to have been written independently of Salus' book. There are only two references to the book, and both seem to be added "post scriptum". The most telling portent appears in the preface where Salus acknowledges his sources of information: those interviewed, including Cerf, Kahn, McKenzie, Padlipsky, Postel, Quarterman and Walden; Marlyn Johnson of SRI and staff of BBN for supplying early documents; and a number of net personalities who provided corrections and additional information.

The material in the book seems to be closely focused on these sources. Leonard Kleinrock, Vint Cerf and Bob Kahn are prominent, but I combed the entire book for the name 'Van Jacobsen' or a reference to the MBONE and was sadly disappointed. Unfortunately, the net is growing so quickly that detailed history is simply out of the question, if the book is to claim to be up-to-date.

Salus presents the very early history with an unusual angle. He briefly covers early attempts at remote access to computers from the 1940s, then describes the early history of the Advanced Research Project Agency from the cold war onwards. This sets the scene for the ARPANET project.

Part two I found the most satisfying. It starts with the IPTO interested in computer networking for resource sharing among the various ARPA projects. Larry Roberts moved from the Lincoln Laboratory to the IPTO, set up a group for discussion, and very soon the idea of a packet-switched network relying on

"Interface Message Processors" (IMPs) attached to host computers was born. Salus then describes the project tender awarded to Bolt Beranek and Newman and the development of the RFC process. Most interesting of all, he describes the hardware and software elements which made up the IMP-based ARPANET. Much of the material in this section is culled from BBN reports, but it is forgivable here when few of the documents are on-line.

He closes the historical section of part two with the International Computer Communications Conference of 1972 in Washington D.C. which was the major public 'unveiling' of the ARPANET and the beginning of its use as a functional entity. Part two closes with a brief aside describing the ALOHA packet radio, Ethernet and SATNET technologies then under development.

Part three chronicles the growth of the network and its services through to the traumatic change from NCP to TCP/IP on 1 January 1983. Salus starts with a brief discussion of networking outside the USA. In particular he focuses on the precursors to OSI: the CYCLADES project in France, EPSS in the United Kingdom and mentions work in Germany and Japan.

Alas, no mention of Australia's ACSnet or CSIRONet. There follows a discussion of the early attempts at network mail up to the RFC 733 specification. Then back five years to examine Bob Kahn and Vint Cerf's proposal of the "Transmission Control Program" as a replacement for NCP. Three pages unexpectedly intervene on early litigation related to computer networks in the USA, followed by extensive criticism of the OSI protocols in comparison to TCP/IP (presumably derived from Mike Padlipsky) with many confusing diagrams of the OSI models and standards lists. At this point Salus introduces UNIX, covering the various TCP/IP implementations which culminated in the September 1983 release of 4.2BSD. UUCP and Usenet are discussed, including an intriguing examination of the early alt hierarchy (but not net.suicide). Backwards again, he returns to the NCP/TCP transition, which leads to a discussion of the infamous sendmail program and the revised mail standards (RFC 822) necessary due to the transition. Salus implies that sendmail's design was driven by the traumatic period of change on the net. He concludes with a brief discussion of some store-and-forward networks including BITNET, Fidonet and the emergence of UUNET.

Part four examines the various independent computer networks which developed in the early 1980s, including IBM's VNET and the research networks in the UK, Europe and Japan. Then the ARPANET fragmentation started. MILNET was created and the NSFNET and many other regional networks appeared. Salus briefly discusses the network re-organization

that followed. Finally, part five provides a snapshot at the time of writing. He covers the 'standard' recent applications (archie, gopher and the WWW), discusses security and the provision of commercial services, briefly touches on IPng and concludes with some projected network statistics.

Overall, the book does not provide a broad historical view of the Internet, but it does provide some interesting information on the early ARPANET technology. My personal interest in net history is directed at the pre-TCP/IP period, so Salus was actually satisfying, but only for a quarter of the book. As the net developed, it became quite diverse, resulting in enough material for many more volumes of this size.

Salus wavers between a technical treatise and a popularisation of net history, but there isn't much detail of the former (in fact, reading this review you already have most of the information), and I suspect people without networking background will become quite lost beneath the assumed knowledge. Every so often, a grammatical or typographical error indicates that the book was put together quickly, but, as with most other books on the Internet, it is already showing its age, with a coverage that extends only to 1994. It is an adequate introductory work for those without enough time to search for papers and RFCs. Perhaps Salus should simply remove the "and beyond..." from his subtitle. ♦



## The Magic Garden Explained Solutions Manual

by Berny Goodheart and James Cox  
Prentice Hall 1995

271pp., \$22.95

ISBN 0-13-207556-3

Reviewed by Adrian Booth

Tandem Computers

<booth\_adrian@tandem.com>

My first impression upon picking up a copy of TMGESM was that it was surprisingly large: almost half the size of TMGE. This was soon explained by a quick flick through the book: more than half of it consists of source code!

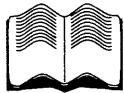
As you would expect, TMGESM closely follows the structure of TMGE, even going as far as omitting Chapter One, since there were no exercises in that chapter of TMGE!

A very full solution is given for each of the exercises in TMGE: Berny and James must have regretted putting in a few of the more jagged exercises! In many cases source code is provided that fully fleshes out the detail related to a particular exercise, even when the exercise itself did not explicitly call for a program to be developed.

Full source code is given for such things as `malloc()`, a RAM disk driver, and a simple filesystem. Fortunately people who purchase the book can FTP the source code for non-commercial use.

I found it a little annoying that some answers are presented in a slightly loose fashion—I prefer conciseness and rigor in a technical reference. This was only apparent in a few of the answers however.

TMGESM would be a worthwhile addition to the bookshelf of anyone who found TMGE interesting, whether they struggled with TMGE and want to see some answers to the exercises, whether they want to delve further into the details, or whether they believe that source code is the only way to explain concepts. Recommended!♦



## Learning the Bash Shell

by Cameron Newham and Bill Rosenblatt  
O'Reilly and Associates, Inc. 1995  
240pp.

*Reviewed by Kate Lance*  
University of Newcastle  
<clance@cs.newcastle.edu.au>

A few years ago I took over the system administration of a site where the local default shell was bash. Until then I'd been a C shell user, but apart from some initial puzzlement about the different format for aliases it was very easy to adapt to the new shell. A constant annoyance, however, was that only the man pages were available for advice and insight into its workings. Now the ever-dependent O'Reilly and Associates have come to the rescue with another of their beautifully produced and organised handbooks, "Learning the Bash Shell".

bash is a backwardly-compatible descendant of the original Bourne shell, which includes many of the most useful features of other shells as well as its own innovations. It's part of the GNU (Free Software Foundation) project, and has regular updates, fixes and enhancements. It's also the shell distributed with Linux, the free UNIX operating system for PCs.

The major advantages of bash are that it offers the powerful programming syntax of Bourne shell, the job control and alias features of C shell, and the command-line editing of Korn shell. It also provides the text-completion features of emacs—at the command line, a TAB will attempt to uniquely complete a partially-entered string for a command, function or filename. bash functions allow users to define and execute a chunk of shell code by name, which is faster and more efficient than running an equivalent script.

Command-line history scrolling and editing is provided by the Readline library, which may be used in vi or emacs editor mode. If you haven't previously experienced command-line recall and editing, I can only say that it is possibly the single most useful tool available to UNIX users -- it would be hard to imagine working without it now.

"Learning the Bash Shell" introduces bash and shell concepts, then gives a very nice summary of editing features for people new to emacs or vi. It goes on to describe how to customise your environment with the startup files, `.bash_profile` and `.bashrc`. It works through simple and advanced shell programming principles, flow control, command-line options, input/output and command-line processing, process handling, and program debugging.

Two very useful chapters cover installation, configuration and security issues for system administrators, while the Appendices discuss bash in relation to Bourne shell, POSIX standards, the Korn and other shells, future features to be implemented, a series of reference lists and a syntax summary, and finally, how to obtain the program.

A minor criticism: I was hoping for more information on the `typeset` and `declare` commands: with the `-x` flag, for instance, they define and simultaneously export a variable without an explicit `export` statement. (However, the copy of the book I have is still at the draft stage, so perhaps this will be remedied.)

"Learning the Bash Shell" answered a number of niggling questions I'd had about bash over the years, and provided a welcome structure to understanding its features and power. There will probably never be an "ultimate" shell, but bash comes close to being one of the most generally useful and flexible -- it's backed up by the ideals, resources and responsiveness of the GNU project, and best of all, unlike the Korn shell, it's totally free. This book can only help bash become

better implemented, appreciated and understood, and I recommend it to all UNIX shell users. ❖



## Teach Yourself PERL in 21 Days

by Dave Till

Sams Publishing 1995

841pp., \$49.95 (paperback)

Reviewed by J. Wright

Guru Software Services

<guru@acslink.net.au>

Once upon a time, the only way to learn Perl was to acquire the "camel" book *Programming Perl* (Wall and Schwartz). This was a book written for fairly experienced programmers. Later a second volume (the llama book) called *Learning Perl* (Schwartz) was released by the same publishers. This was a much gentler introduction but in a very similar style. Novice programmers could survive but it was still a fairly spartan introduction. Worthy as these two volumes are, there was still a real need for a book written for complete non-programmers. This volume purports to be exactly the solution required.

The book is large (thicker than both the O'Reilly books together). It features, in pseudo-O'Reilly style, a bunch of pearls on the front. The style is easy to follow and full of examples (both working and non-working). Each example is followed by an analysis section which explains every line in the example. In almost all cases the analysis is considerably longer than the program and often variations or alternatives are discussed.

At the end of every chapter is a series of useful revision questions that ensure the reader has understood the material. (Answers are supplied in an appendix). As well as examples, the text is interspersed with grey boxes to emphasise or illustrate various points. Each of these boxes is accompanied by a symbol to denote the type of box - this is eye catching and certainly complements the surrounding material. Each lesson (there are 21 in total) has a set of goals at the start and a summary at the end. There is also a weekly set of goals and a weekly summary.

As an experienced Perl programmer, I found the book to be repetitious but for a complete stranger to the language, this will probably re-enforce the concepts being presented. It would also seem that the author was probably a C programmer before learning Perl. This is reflected in the way he presents C concepts first and leaves many of the "different" features until later in the book. In some ways, this is a pity, as most expert Perl programmers tend to use constructs that fit the

task at hand rather than adopting those with which they are most familiar.

The book also defies certain "traditions" that have been widely adopted (mainly due to discussions on comp.lang.perl.\*) and to reviews of Larry Wall's own code. For example, perl programmers normally use the construct:

```
print "Text and $arguments mixed together." ;
```

whereas the author prefers to use:

```
print('Text and ', $arguments, 'mixed together.');
```

which looks very C-like. There is a brief explanation for this around Day 15 but it is not convincing and there is always the risk that a novice using this style could confuse the two and end up with:

```
print 'Text and ', $arguments, 'mixed together.' ;
```

which is ambiguous if the first string looks like a file pointer.

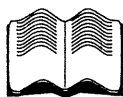
Apart from the question of ordering the contents and the overall novice approach to the language there are a significant number of errors in the book. These run from trivial spelling mistakes and the poor colouring of the comment boxes (only in the later chapters) through to significant blunders in the sample code. The errors include both poor coding and incorrect algorithms. One of the significant advantages of Perl is that it supports most constructs from most languages which means that there is almost always a construct to achieve exactly the result required. For example, in perl there is the "foreach" loop. This is not mentioned until after considerable discussion of how to loop through arrays without such a construct. For a novice (non-programmer), it is irrelevant that you can achieve the same results without a particular construct. If it is available, they need to know how to use it.

A final disappointment is the coverage of Perl 5. This is done, briefly and inaccurately, in an appendix. Given the publishing date (1995) and allowing six to nine months lead time, it still would have been possible to include a better discussion of this major new revision. After all discussion on the features have been widespread through the newsgroups and also the perl-porters mailing list. In fact, AUUG last year (1994) included a tutorial on Perl 5 which was only possible because it had already become available.

In conclusion, I would not recommend this book. At least, not yet. It is possible that the next edition will fix the glaring errors and cover Perl 5. In the meantime, if you would like to learn about the "Swiss Army



Chainsaw" of computing languages, try the camel and llama books from O'Reilly. ❖



## System Programming for UNIX

by David Curry

O'Reilly & Associates, Inc. 1995

519pp, paperback

ISBN 1-56592-163-1

Reviewed by Michael Haldey

<mhaldey@ozemail.com.au>

*Disclaimer: what I reviewed was not an actual book, but a final draft, so I am not aware of what was changed in the final print.*

The System Programming for UNIX by David Curry is based on his previous book, Using C on the UNIX System. The only UNIX implementation discussed is Solaris 2.x and the book tries to cover virtually all aspects of everyday programming in the UNIX environment. Special attention is paid to the aspects of porting software from BSD based environments.

The book starts with the Introduction. After that the author considers various functions like manipulating strings and characters, dynamic memory allocation, etc. Then low level i/o routines are described, standard I/O library, files and directories, special-purpose file operations (including multiplexing with select and poll), memory mapped files, time of the day operations, users and groups, system configuration and resource limits, processes, terminals, interprocess communication, networking with sockets and TLI and some other routines such as sorting, pattern matching, internationalisation.

The Appendices describe the changes in ANSI C, accessing file system data structures, using /proc file system and working with DLPI.

The book is filled with a large number of small examples illustrating the system calls and functions described. The text itself is easy to read but the book doesn't have any schemas or drawings. You won't find a lot of tables either. Such style, while acceptable for more basic books, reduces the use of the book as an everyday reference and sometimes makes the life of the reader quite hard. Say, in Chapter 10 (Signals) the author lists 31 signals (paragraph per signal) and then says that 'All UNIX systems provide the first 15 signals in the list above'. It is up to the reader to count where the fifteenth signal is.

Obviously, because the scope of the book is very wide, it cannot be advanced enough in everything it covers.

There are a number of chapters which are planned to be just introductions - fortunately the references for further readings are provided straight in the text (networking, internationalisation). But sometimes the coverage is not deep enough and no reference is given for further details. For instance after reading that '...in Solaris and SVR4, it is possible to enable the automatic restart of system calls on per signal basis' I couldn't find any idea of how to do this, which calls can or cannot be restarted or where to look for further information.

Sometimes the references are not provided even within the book. For example, the author uses the concept of 'locale' in the Chapter 7 (Time of Day operation) without saying what it is though the brief discussion is in Chapter 16.

Generally, this is a book that leaves very mixed feelings and is very hard to judge. Some parts are very good and author has really managed to say a lot in a very small space. There are a lot of small and useful tips scattered around the book. But sometimes there are obvious omissions and some important issues are not discussed properly. Just a few examples. When the soft links are discussed, it is not said which functions follow links, which are not. In the discussion about multiplexing (Special file operations, Chapter 6) the functions select and poll are discussed without mentioning that select is a function based on poll system call. These omissions are numerous.

So, for whom is the book? The author himself has defined three categories of readers:

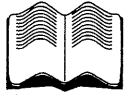
- UNIX System Programmers who are not familiar with Solaris 2.x system and need to port something to the new system
- People who aren't system programmers but still need to port some software from some other version of UNIX to Solaris 2.x
- C programmers who wish to move into UNIX systems programming.

For a person from group 1 the book maybe quite useful especially if they had some experience with a BSD derived version of UNIX. Such person would probably skip half of the text but the rest would do the job. As said above, there is a lot of useful information and advice about porting from BSD to Solaris 2.x.

For group 2 the book is just right providing that a person knows a bit about UNIX.

For a person who wants to program in the UNIX environment professionally, I wouldn't suggest buying this book at all. Firstly, it is limited to Solaris 2.x only. I can't disagree with the author that it is the most popular implementation of SVR4 today, but it would take quite a while before a majority of running UNIXes would become SVR4 compliant. On the other

hand, the POSIX standards coverage is not very thorough (for instance, POSIX signal handling is discussed but which signals are defined by POSIX is not shown). And, finally, there is a better book on the market for this category of people, namely *Advanced Programming in the UNIX Environment* by R. Stevens. ❖



## OzInternet: A Guide to connecting to the Internet in Australia

by Berny Goodheart and Frank Crawford

Prentice Hall Australia 1995

289pp., \$35.99 (soft back)

ISBN 0-7248-0905-8

*Reviewed by Jon Wright*  
*Guru Software Services*  
*<guru@acslink.net.au>*

At first glance, this book is yet another book on the Internet. At second glance one of the authors is a technical expert who writes in-depth technical articles and is co-author of "The Magic Garden Explained". Neither fact is likely to encourage the casual reader but this would be a major mistake. Firstly the book is about the Internet but it covers a gap in the available literature and secondly the book is easy to read and yet technically accurate.

In the introduction, the authors casually mention that there are more than 2000 ISBNs assigned to Internet books. That seems almost impossible to imagine. The problem is that almost all of these books assume that you have an internet connection and then work from that point. In particular the comment is often given that connections are getting cheaper so ask an ISP (Internet Server Provider). That only works if you already know what you want and are in a position to understand the available choices. The other problem is that many of the available Internet books have a very USA-centric view of the universe which is not particularly helpful here in the fringes of American civilisation.

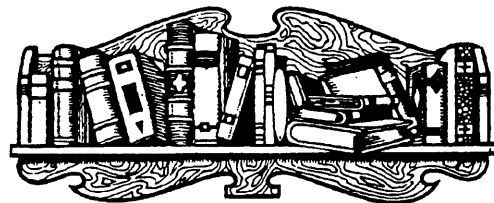
OzInternet addresses all of this but explaining how to become connected to the internet. There are specific chapters for the Macintosh, OS/2, and a PC running that other operating system. In each case the instructions are step by step to ensure that nothing is missed. The only assumptions are that you are fairly familiar with a computer but the actual sequence of actions is very clearly described. This is particularly valuable because setting up the connection is not as trivial as it may seem.

As well as the chapters which specifically cover the various systems, there is a chapter on Modems which should be a prerequisite for anyone attempting to purchase one. There are also chapters on security, internet societies, networking basics, history, connections and some background material. There are a number of very important appendices including: Public access sites; Archie.au; Anonymous FTP archive sites; FTP mirror sites; Australian newsgroups; WWW sites in Australia and also a description of the Parliament internet trial. (This is not a court case about the Internet; it is a trial whereby information from the Parliament is available on the Internet).

The one concern with this type of book is the status of the information it contains. For example, how long will the list of ISPs remain up-to-date? This issue has been considered and the authors document where the updates are available. For example, some of the material is taken from a FAQ, the location of which is documented in the book. There is also a new edition coming at the end of the year but even without that, the book contains enough information to make it invaluable. Any updated information can be obtained once the user is "on the net"!

The only note of caution I would make is that all references to the ACS service (ACSlInk) should be ignored. Since printing, the ACS have decided to disband the service.

I highly recommend this book to everyone. For those who are thinking about a connection for the first time, this book will guide your choice. For those who have a connection through work, this book will help you set up a link at home. For those who don't know what the internet is, read the introduction—then you will also need it to help you set up a connection. Finally for those of us who have "always" been on the internet; this is the book for anyone that asks you how to do it. It may also help if you decide to set up links at home or to change ISPs for staff working from home. ❖



New edition of  
one of our  
bestselling  
UNIX books  
now  
completely  
revised &  
updated.

## UNIX FOR THE IMPATIENT

Second Edition

Paul Abrahams and  
Bruce Larson

0-201-82376-4  
650 pages/ Paperback



20% off for  
AUUG  
members

Clear, concise, and readable. Thoroughly cross-referenced and indexed, UNIX for the Impatient is the power-user's guide to mastering UNIX quickly.

**Organized** functionally and useful for any UNIX implementation on any platform, including System V, BSD, LINUX, Solaris, POSIX.2, ULTRIX, UNIXWARE (Novell), and others.

Covers **all essential information** including:

- File operations
- Data manipulation using filters
- The awk programming language
- Shells and shell scripts, now using the KornShell
- Editors — vi, ex, ed, and GNU Emacs
- Mailers, remote addressing, telnet, ftp, uucp, WWW, other Internet issues
- GUIs, including X Windows

Available late August — get your advance order in today!

I would like to order \_\_\_\_\_ copy/copies of **UNIX for the Impatient** by Paul Abrahams and Bruce Larson (82376-4) at the special AUUG price of \$35.95 (RRP \$44.95)

I enclose a cheque for \$ \_\_\_\_\_ **OR**  Please charge my credit card No: \_\_\_\_\_

BANKCARD     VISA     MASTERCARD     AMERICAN EXPRESS ID No: \_\_\_\_\_

EXPIRY DATE:    /    /    AUUG MEMBERSHIP NO: \_\_\_\_\_    PHONE NO: \_\_\_\_\_ (Wk Hrs)

NAME: \_\_\_\_\_    SIGNATURE: \_\_\_\_\_

STREET ADDRESS: \_\_\_\_\_

STATE: \_\_\_\_\_    POST CODE: \_\_\_\_\_

Send your  
order to ►

## Addison-Wesley Publishing Company

Unit A1, 6 Byfield Street, North Ryde, NSW 2113, Australia.  
Telephone (02) 878 5411    Customer Service Fax (02) 888 9404



---

WAAUG news:

## From the Western Front

*Edited by Janet Jackson <janet@dialix.oz.au>*

Well AUUG95, or whatever it was, is over, and Glenn and I are starting to settle in to our new house in the hills (we moved about three weeks before the conference), and I'm starting to feel like I might actually be catching up with my to-do list (it's getting a little shorter). I'm late with this column though—sorry Phil.

Organisation is under way for the 1996 Perth summer conference. In his report below, Adrian makes it sound like I'm organising the whole thing (no thanks!), but he is still in charge as chair of the conference committee. I am *only* doing the program and (hopefully) MCing the conference; other volunteers are taking care of the catering and logistics. If you would like to help, please contact Adrian. If you would like to present a paper or tutorial, please contact me. The deadline for abstracts is 15 November 1995.

James Patton has started on his job of Meeting Reporter, as you will see below. However, he needs volunteers to help him out by writing up the meetings. So if you are able to review/summarise a meeting or two, please contact James.

(See the Committee Contact Details below for how to contact Adrian, James or myself.)

---

### From the Chair

*Adrian Booth <booth\_adrian@tandem.com>*

I'm writing this a few days after returning from AUUG95. Briefly: the conference was alright, the exhibition was bland, the conference dinner cool and the socialising great! Angela (my wife) really enjoyed Sydney, the conference dinner, and the socialising too: and I am told that a partner program may be organised for next year!

It was also great to have an opportunity to meet AUUG's new business manager, Liz Egan; AUUG has achieved a hat-trick of attracting very experienced and capable (not to mention charming!) people into the position. On behalf of WAUG, welcome aboard, Liz!

I thought it was disappointing that none of the international speakers at AUUG95 were flown around to any chapters, especially after the success of the Linus Torvalds visit to WAUG after AUUG94. I guess, though, that none of this is going to happen by magic!

Due to a last minute kerfuffle with my US trip, I had to miss the AUUG chapter council meeting with almost no notice. However, now that AUUG95 is over I hope to see the chapters again starting to focus on the benefits we can bring our members.

Speaking specifically about WAUG: as you know by now, Janet Jackson is the programme chair for the 1996 Perth Summer Conference, and has been doing a great job! Not only is it a relief to give it up after three years, but I hope now to get firmly stuck in to organising new and better events over the next few months: I'd be keen to hear any suggestions that *you* have!

I'd like to see WAUG fly over a good interstate speaker before the end of the year, and hopefully be involved with another international speaker's Down Under tour in the near future. Again, suggestions are very welcome!

Our change of venue to the Freeway Hotel, South Perth, seems to have coincided with a fall-off in the number of people attending our meetings, which is strange since it is a vastly superior meeting place! If you haven't been along to one of our meetings since the change, give it a try; you are sure to be pleasantly surprised!

I have not heard from anyone who would be interested in taking on the role of publicity officer for WAUG, and so in the interim I will be adopting that role. However, I'd love to hear from anyone who would be interested in filling it in the longer term.

Having missed the last few meetings after travel to the US and then AUUG95, I'm looking forward to a quiet few months in Perth and catching up with everyone again at the next few WAUG meetings...look forward to seeing you there!

## WAAUG meeting reports

James Patton <caf@inet.net.au>

### July

#### MPI and PVM Parallel Programming Environments

Presented by Dr. Chris Macdonald  
The University of Western Australia  
<chris@cs.uwa.edu.au>

Chris discussed the two most successful message passing environments that offer interprocess communication between heterogeneous systems. These were the rigorously defined Message Passing Interface (MPI) and the current *de facto* environment, the Parallel Virtual Machine (PVM).

Chris said that PVM and MPI are both free environments that have evolved over time, and they both achieve pretty much the same end. PVM and MPI are used by the scientific and engineering communities on networks of heterogeneous machines to perform user-level message passing from C and Fortran. Chris commented that along with Linux, PVM and MPI are two of the most interesting developments of the Open Systems/free software movement.

Some of the users of parallel programming include physicists, industrial chemists, mathematicians and, of course, computer scientists. MPI and PVM have been used by Boeing to calculate air flow over aircraft, by the Numerical Analysis Group to perform statistical analysis, and to estimate the depth and density of oil wells. These tasks are ideally suited not only to networks but also to multi-user, multiprocessing machines such as those made by Fujitsu and Tandem.

PVM and MPI are software systems that enable a heterogeneous collection of computers (sometimes called a cluster) to be programmed as a single machine. They provide the programmer with process-control and message-passing calls for communication between tasks running on different hosts. PVM achieves this by establishing a master/slave relationship between machines on a network. The master farms off the work to the slaves, and the master interprets the results.

PVM currently supports Fortran and C programs on a variety of machines (from workstations to supercomputers) and a limited number of operating system/network environments. The operating systems include various ports of UNIX as well as Windows and Windows NT. Chris suggested that PVM is quite language-independent, whereas MPI is more suited to Fortran.

In both PVM and MPI you can bypass the message-passing by writing straight to shared memory. Both MPI and PVM add about a 13% overhead to IP packets for network message-passing traffic. So if you can get your code to determine the architecture of the machine you can speed up your application. However, this foils portability.

Chris said that PVM is controlled by Oakridge Labs. PVM has not been rigorously standardized, but there is only a single version, which is released at six-monthly intervals. Also, there is only one set of PVM source. Problems are often reported on a small set of newsgroups.

MPI, in contrast, has three or four implementations available, and has stronger standardization.

PVM is OS-independent, and supports threads at the kernel level.

MPI has no process management. If MPI starts one process, that process automatically starts up on all available processors. During execution processors can die, but MPI can't add new processors on the fly.

Tcl and Tk are useful tools to visualize interprocess communication over a network. Traditional debugging tools such as dbx are not much good for debugging MPI/PVM applications. The PVM project has produced a couple of useful tools for PVM programming:

HeNCE: Heterogeneous Network Computing Environment  
AB window Analysis and Debugger

HeNCE is an Window System parallel development tool for PVM. Using HeNCE, you can graphically describe the parallelism in your program, and HeNCE will automatically add the PVM code for parallelization. HeNCE will make a trace file so you can have a post-mortem look at execution.

AB is an tool which lets you trace individual messages. It has VCR-type displays which allow you to control the flow of your program while it's running. Your code is instrumented by linking to a special library.

PVM is available via anonymous ftp from netlib2.cs.utk.edu. MPI is available from Argonne National Labs and a couple of other places.

22

## August

### ServerNet

Presented by Adrian Booth  
Tandem Computers  
<booth\_adrian@tandem.com>

Adrian's topic was Tandem's new ServerNet.

Tandem bills ServerNet as a "Breakthrough Technology for the Coming Generation of Applications". Adrian described it as being designed using a "Meccano set" philosophy. That is, the system has unlimited scalability and can be customised. For example, this week you may set up the system for fault tolerance, and next week you could redesign it for maximum processing speed. ServerNet can be configured in various ways with hundreds or thousands of processors.

Adrian stressed that one of the main strengths of ServerNet is its ability to satisfy massive data throughput requirements. For example, today a high-speed storage device can deliver data at about 12 megabytes per second. A high-speed processor I/O bus connection can handle about 20 megabytes per second. Internal processor buses can be as fast as 1 gigabyte per second. In contrast, Tandem claims that the first generation of ServerNet will offer aggregate throughput of up to 150 terabytes (150,000 gigabytes) per second! Ouch!

Such extraordinary levels of throughput will become increasingly useful to companies which collect, analyze and access huge quantities of data.

Adrian gave several examples of business applications that would require these massive throughput capabilities. One was decision-support systems such as data mining, executive information systems, and data warehousing. Adrian used the example of the Fly Buys scheme. He claimed that the chief purpose of Fly Buys was to collect huge amounts of data about consumers' buying habits. This data could then be processed to give companies useful marketing information, such as when a person might be most likely to buy a new car. Marketing campaigns could then be targeted at individuals to provide the best service to customers and the maximum profits to retailers.

Other prime candidates for ServerNet would be applications that require fast throughput of voice, video and image data. Often these large-scale, high bandwidth objects need to be moved from place to place with minimal processing in between. For example, a company with an Internet shopfront may need to be globally accessible 24 hours a day, and

provide thousands of customers with large amounts of data such as video presentations or complex satellite images. The I/O architecture of ServerNet would overcome traditional bottlenecks and allow petabytes (thousands of terabytes) of data to be moved around with minimal processor intervention or delays.

Adrian said that the ServerNet architecture achieves these goals through any-to-any direct connections, allowing data to flow directly from processor to processor, from processor to device, or from device to device. This avoids hardware and software bottlenecks, and conserves memory bandwidth and CPU cycles that can be wasted in more conventional architectures that use shared memory and devices.

Adrian said that ServerNet would be an "open system". For example, it would support PCI, SCSI, ethernet and ATM standards. It could run various operating systems, including UNIX (of course) and NT. One interesting comment was that perhaps AT&T's Plan 9 would be ideally suited to exploiting the capabilities of ServerNet. ServerNet was also billed as being "affordable". However, no prices were given.

Adrian concluded his talk by admitting that he had not yet seen one of these machines. However, Tandem were flying him to Texas the following week to see one in action, and at the October meeting he would present a brief to bring WAUG up to date.

---

## Meeting information

WAUG meets at the Freeway Hotel, 55 Mill Point Road, South Perth. We meet at 6:15pm on the third Wednesday of each month.

Our meetings are advertised in the Diary column of the Computers section of Tuesday's *West Australian*.

If you need further information about the next meeting, please contact Mark or one of the committee.

### **SPEAKERS ARE NEEDED!**

...especially ones who can actually commit to giving a talk on a certain date! So if you can give a talk, or know someone who can, please let us know. Mark (our meeting organiser) cannot produce them out of thin air.

## WAUG Email Aliases, Newsgroups and Web Page

WAUG has the following mail aliases on [uniwa.uwa.edu.au](mailto:uniwa.uwa.edu.au):

**waug-membership** for membership enquiries  
**waug-chair** our Chairperson  
**waug-meetings** our meeting organiser  
**waug-secretary** our Secretary  
**waug-newsletter** for contributions/enquiries  
**waug** for general correspondence (will be read by the Secretary, as a paper letter would be).

So, for example, you may send general correspondence to [waug@uniwa.uwa.edu.au](mailto:waug@uniwa.uwa.edu.au).

Check us out on the World Wide Web at <http://www.auug.org.au/auug/waug/waug.html> (thanks Canberra AUUG).

Also see the newsgroups [wa.waug](mailto:wa.waug) and [aus.org.auug](mailto:aus.org.auug) for announcements and discussion.

## Committee Contact Details

Office-bearers:

Chair:  
 Adrian Booth 321 9111 [booth\\_adrian@tandem.com](mailto:booth_adrian@tandem.com)

Treasurer:  
 Patrick Ko 483 8111 [pko@DIALix.oz.au](mailto:pko@DIALix.oz.au)

Secretary:  
 Major 357 5076 [major@yarrow.wt.com.au](mailto:major@yarrow.wt.com.au)

Ordinary committee members:

Mark Baker 491 6081 [baker@telecomwa.oz.au](mailto:baker@telecomwa.oz.au)  
 (Meeting Organiser)  
 David Buck [dbuck@ncc.telecomwa.oz.au](mailto:dbuck@ncc.telecomwa.oz.au)  
 Luigi Cantoni 474 3700 [lui@DIALix.oz.au](mailto:lui@DIALix.oz.au)  
 Don Griffiths 351 7691 [griffith@cs.curtin.edu.au](mailto:griffith@cs.curtin.edu.au)  
 Tom Hallam 380 2665 [thallam@geol.uwa.edu.au](mailto:thallam@geol.uwa.edu.au)  
 Glenn Huxtable 328 8288 [glenn@fs.com.au](mailto:glenn@fs.com.au)  
 Janet Jackson 272 5061 [janet@DIALix.oz.au](mailto:janet@DIALix.oz.au)  
 (AUUGN Sub-editor)  
 James Patton [jrp@mrwa.wa.gov.au](mailto:jrp@mrwa.wa.gov.au) (Meeting Reporter)

A publicity officer has not yet been appointed. Please volunteer. (Adrian is acting in the interim, but don't you think he has enough to do already?)

## For Systems Administrators: Local SAGE-AU Meetings

The WA Regional Group of the Systems Administrators Guild of Australia (SAGE-AU) meets on the fourth Tuesday of each month at 6pm, in room G3 at the Alexander Library.

If you manage computer systems for a living, we'd like to have you along. SAGE-AU is not another UNIX group. All systems and network administrators are welcome. We would particularly like to see more PC network administrators attending, so if you know any, send them along.

For more information, please contact the regional group chair, Janet Jackson <[janet@DIALix.oz.au](mailto:janet@DIALix.oz.au)>, (09) 272 5061, or the meeting organiser, Mike Horton <[mikandfi@DIALix.oz.au](mailto:mikandfi@DIALix.oz.au)>, (09) 479 8424. For information about SAGE-AU in general, you may also look at <ftp://ftp.sage-au.org.au/pub/SAGE-AU> and <http://www.sage-au.org.au:8080/> ♦

# AUUGN

## IS YOUR JOURNAL!

Without you, there is no AUUGN: if you've knowledge to share, share it through AUUG's bimonthly journal.

You'll be reaching over 700 individuals, and more than 300 organisations involved in the UNIX/Open Systems world.

### WE'RE LOOKING FOR:

Talk to your local Chapter contact for ideas, and see elsewhere in this issue for submission guidelines.

- Papers
- Reviews
- Articles
- News
- Comment

---

QAUUG news:

## Qld. Chapter report

### September meeting

A healthy crowd turned up to Brisbane's historic Regatta Hotel on Tuesday the 26th for the September QAUUG meeting. While there were certainly most of the "regular" attendees we also welcomed some newcomers—always a healthy sight, and we hope they enjoyed the meeting and will attend future events.

Michi Henning from DSTC chaired the meeting, and began with a brief summary of the AUUG 95/WWW conference. There were varying opinions as to the overall quality of the event, but generally it was felt that some of the WWW speaker content left a lot to be desired, while there was unanimous praise for Keynote Speakers Geoff Huston and Rob Pike.

Michi also introduced Sequent's Colin Webb, who proceeded to deliver an entertaining and educational presentation on RAID technology. Colin is an expert in the area, and his knowledge and attention to detail shone through.

The Queensland Chapter continues to grow from strength to strength, and the monthly meetings (always held on the last Tuesday of the month) continue to provide an excellent forum for users, developers, and managers of UNIX and Open Systems. For further information on QAUUG you can contact Rick Stevenson ([rick@stallion.oz.au](mailto:rick@stallion.oz.au)) or Mark White ([m.white@brisbane.tandem.com](mailto:m.white@brisbane.tandem.com)).

---

AUUG Canberra news:

## Canberra chapter

### The INTERNET Project

If you are an AUUG member in the Canberra Chapter you can get email and news access to the Internet, and there is no cost (so long as you remain a financial member). This is provided by a dialup service which has been recently updated and has a direct IP connection to the Internet. Full IP access to the Internet can be obtained at extra cost (\$120 for 400 hours, approximately). If you are interested in this service,

please contact John Barlow (mobile: 019 935477) to have a chat about it.

We have a Linux box and a FreeBSD box available via the same dialup service, so if you want to examine these great (free, full source) UNIX implementations, here is a prime opportunity! Existing users who wish to use these boxes need to email [linux@auug.org.au](mailto:linux@auug.org.au) or [freebsd@auug.org.au](mailto:freebsd@auug.org.au) to have an account set up.

If you want a specific topic discussed at a future meeting, or want a specific UNIX presentation made, please contact John Barlow.

Secretary, Canberra Chapter of AUUG Inc.  
John Barlow Tel.: 019 935477  
[cauug.secretary@auug.org.au](mailto:cauug.secretary@auug.org.au)



# From ;login:

Reprinted with permission

The following reports are published in this column:

- Report on POSIX.2: Shell & Utilities
- Report on POSIX.1b: Realtime
- Information Infrastructure

Our Standards Report Editor, **Nick Stoughton**, welcomes dialogue between this column and you, the readers. Please send your comments to [nick@usenix.org](mailto:nick@usenix.org).

## STANDARDS

# An Update on Standards Relevant to USENIX Members

by *Nicholas M. Stoughton*  
 USENIX Standards Report Editor  
[nick@usenix.org](mailto:nick@usenix.org)

I have on my bookshelf at home a copy of the first POSIX standard, the 1986, trial use version of POSIX.1, also known as "IEEEIX." Two years later, the first full use standard was published. Two more years, and this was revised: the now famous POSIX.1-1990, the standard that has been adopted by ISO as International standard 9945-1:1990.

Five years have passed since then, and POSIX has not been sleeping; a whole raft of standards have been published. But after five years, the IEEE insists that every published standard is checked, in case it has become obsolete in any part. PASC, the Portable Applications Standards Committee of the IEEE Computer Society, who are the sponsors of POSIX standards, have decided that they do not have the resource at present to revise POSIX.1. There is to be a ballot, which I am responsible for coordinating, on either reaffirming POSIX.1 as it stands, or withdrawing it completely.

If, like me, your copy of POSIX.1 is the book you read most; the bible you follow for all your applications, then you should be voting to reaffirm it. If you want it revised, the best way is to join the working group, who are sure to revise it before the next five years is up.

Of course, if you think it's obsolete, and a complete waste of everyone's time and effort, now is your opportunity to drive a nail through the coffin and vote to withdraw the standard (incidentally killing all the standards that have been based on it, such as POSIX.1b-1993).

We all know that maxim well, and to a great extent, it helps us more than harms us if there are two overlapping standards. Users can choose which is the one most relevant to what they want to do. Of course, vendors hate having to supply support for both, but their whole reason for existence is to satisfy their customers (oh, and make some money while they are at it).

What you *don't* do when you have competing standards is to allow governments to legislate as to which is better. The failure of the OSI protocols to dominate the Internet is a good example; governments tried steering commerce into their preferred direction, but the power of democracy won, and we have IP and its related protocols.

In Europe, we have the mad bureaucrats of Brussels who have now made it a criminal offence to use the wrong standards if you sell fruit and vegetables (yes, pounds and ounces can now get you at least a fine, if not a prison term). Anybody from any government anywhere in the world reading this, *don't go making the same mistake with our standards!*

I am going through one of those periods everyone has from time to time when it seems as if everything is changing. As you are undoubtedly aware, back in April we voted to change the structure of the Sponsor Executive Committee (SEC) of the Portable Applications Standards Committee (PASC), effectively halving it. This will affect *every* POSIX working group still in existence. The change is designed to cause only a minimal impact in *what* it is we are doing, being more than anything an administrative device to simplify and streamline the procedural aspects of the work.

That's a minor change in comparison with some of the personal ones going on around me at present as I start my own company, specializing in standards consultancy, and my wife, as I write this is working on increasing quantity of our children by 50% (I did my bit nine months ago, now its her turn).

### Report on POSIX.2: Shell & Utilities

*Tom Shem <tshem@hposl62.cup.hp.com> reports on the July 10-14, 1995 meeting in Nashua, NH*

POSIX.2b (Shell and Utilities) draft 11 closed its "changes-only" recirculation ballot on July 6th. The results from the ballot were not yet available at the time of the meeting. Also, a copy of POSIX.2b draft 11 was sent to ISO. Comments are not expected back from ISO till around the end of the year.

David Korn reported on the Regular Expression (RE) Experts meeting which was held in Toronto, Canada on June 28th. The meetings objectives were to address a number of outstanding POSIX interpretations dealing with RE's. The results from the this meeting of experts were used to close a number of outstanding POSIX.2 interpretation requests. The RE Experts meeting should be considered a sub-meeting of the POSIX.2 working group for all intents and purposes.

A large portion of the meeting was spent addressing interpretations to the POSIX.2 (IEEE 1003.2-1992) standard. In addition, we revisited several interpretations examined at the RE Experts meeting. In addition to those, interpretations were addressed in the following areas: `awk`, `ed`, `expr`, `shell`, filename expansion, `make`, `mkfifo`, and `mail`. The results from the interpretations meetings will be sent to the Interpretations Functional-chair (Andrew Josey) as official interpretation responses.

An outstanding issue for the last few POSIX.2 meeting has been the lack of internationalization (I18N) experts to address I18N concerns in the POSIX.2 standard. It was agreed that this would be a good basis for a new POSIX.2

project if we could get commitment from several key I18N experts. Further discussion on a new Project Authorization Request (PAR) will be conducted at the next meeting in October.

Lastly, since the IEEE results from the POSIX.2b draft 11 ballot had not yet been received by the close of the meeting, we were unable to begin work on resolving ballot objections. In anticipation of their impending arrival, volunteers were assigned as Technical Reviewers (TR's) to specific chapters and sections of the document. The results will be tabulated by the ballot coordinator (Don Cragun) and distributed at a later date.

### Report on POSIX.1b: Realtime

*Joe Gwinn <GWINN@SUD2.ED.RAY.COM> reports on the July 10-14, 1995 meeting in Nashua, NH*

So what is the POSIX System Services Working Group, Realtime Extensions Subgroup, doing?

The big news is that as part of the general reorganization of PASC/POSIX, the POSIX.4 Working Group no longer exists, having been subsumed into the newly constituted "System Services Working Group" (SSWG), as of the end of the April POSIX meeting. SSWG is composed of the antebellum POSIX.1, POSIX.4, POSIX.8, and SRASS Working Groups. SSWG retains all the documents of the old POSIX.4 WG except POSIX.13, which goes to the new "Profiles Working Group" (which consists largely of the old POSIX.0 WG). The July meeting was the first under the new organization. There was surprisingly little chaos. In spite of the internal Brownian motion, the same people are working on the same documents as before.

The new SSWG has largely retained the old structure of the former Working Groups. The former POSIX.4 WG has become the "Realtime Extensions" subgroup of the SSWG, abbreviated SSWG-RT.

SSWG-RT has three approved standards:

- 1003.1b-1993 (realtime extensions, was called POSIX.4)
- 1003.1c-1995 (pthreads, was called POSIX.4a)
- 1003.1i-1995 (technical corrections to 1003.1b)

SSWG-RT has four active formal projects:

- POSIX.1d (more realtime extensions, was POSIX.4b)
- POSIX.1j (yet more realtime extensions, was unofficially called POSIX.4d)

STANDARDS

- POSIX.1n (technical corrections to 1003.1c)
- POSIX.13 (realtime profiles), discussed below. POSIX.1n is a new project as of this meeting.

Where there are two paragraphs covering a given draft standard, the first paragraph gives the current status, while the second describes the content of that draft standard. These descriptive paragraphs do not generally change greatly from report to report.

IEEE Std 1003.1b-1993, the merger of 1003.1-1990 and the realtime extensions POSIX.4, is now available from the IEEE (800 678 4333 in Canada and the USA, 1 908 981 1393 elsewhere, and 1 908 981 9667 by fax) as a published, bound 590-page book.

A useful overview of the realtime POSIX standards has been written by four stalwarts of the former POSIX.4 WG: "NGCR Operating Systems Standards Working Group (OSSWG) Advisory: A Programmer's Overview of the IEEE POSIX Realtime Standards," NGCR Document Number OSS-A 001 ver 1.0, issued 31 May 1994, 73 pages. Available to good homes for the asking. Send email request specifying your snailmail address to Kar Chan at *chank@smtp-gw.spawar.navy.mil*. Snailmail requests also accepted. Write to Kar Chan at the Space and Naval Warfare Systems Command, SPAWAR 331-2, 2451 Crystal Drive, Arlington, VA 22245 USA.

POSIX.1c (pthreads) passed International Standards Organization (ISO) Committee Document (CD) balloting in February 1995 with only one editorial comment, and Draft 10 was approved by the IEEE at the June 1995 Standards Board meeting. POSIX.1c will now be merged into IEEE Std 1003.1b-1993. This published book will likely be called "IEEE Std 1003.1c-1995." and will include all of POSIX.1, POSIX.4, and POSIX.4a, plus the two technical corrections standards POSIX.1i and POSIX.1n. It will take about a year to get to a published merged book, if history is any guide.

POSIX.1d (was POSIX.4b) is in ballot resolution, which is almost complete, except in the areas under discussion with the members of the Common Reference Ballot. Interrupt Control, a particularly contentious area, has been deferred and now resides in a non-normative appendix. The current plan is to recirculate in October 1995.

POSIX.1d, 130 pages, contains a number of realtime interfaces and options that arrived too late to be included in 1003.1b-1993 (which itself consists of POSIX.1 and POSIX.4 combined). The major new interfaces and options are: *spawn()*, a functional merger of *fork()* and *exec()*, needed both for efficiency and to allow use on platforms

lacking memory management hardware, a sporadic-server scheduling policy, used to prevent asynchronous high-priority processing from totally consuming the computer, cpu-time clocks and timers, used both to measure and to bind use of cpu by processes, *devctl()*, the successor to the *ioctl()* of classic UNIX, Interrupt Control, a set of interfaces intended to allow direct application-level control of devices such as array processors and radar signal processors, and Advisory Information, a set of interfaces that allow an application to declare to the kernel that for instance a specified file will be read sequentially, allowing the kernel to optimize performance. A number of existing interfaces are also being augmented by the addition of variants supporting timeouts.

POSIX.1i (also known as "POSIX.4 technical corrections") was approved as a standard by the IEEE Standards Board in June 1995.

POSIX.1i consists of correcting the POSIX.4 interfaces to remedy clashes detected when POSIX.4 was merged into 1003.1-1990 to yield 1003.1b-1993, as well as minor problems discovered by early implementors of 1003.1b-1993. POSIX.1i is a fast-track effort with a very limited and specific scope. As the changes are small, simple, and few, approval is expected to be rapid, at least as POSIX understands the term.

POSIX.1j (was "POSIX.4d") was voted out to ballot at the April 1995 POSIX meeting. Ballot is currently scheduled by the IEEE for February 1996; this 9-month delay is a matter of great concern to us. Production of standards is already far too slow. The issue is being worked with the IEEE. The issues turn on overloading of ballot groups, and of the IEEE's own handling of ballot logistics.

POSIX.1j, 73 pages, contains a number of realtime interfaces and options that arrived too late to be included in POSIX.1d. The major new interfaces and options are:

- Typed Memory, a set of interfaces supporting the *mmap()*-like mapping of diverse kinds of physical memory (e.g., SRAM, DRAM, ROM, EPROM, EEPROM) via multiple and/or diverse physical paths used for instance to access special hardware and memory via attached VME busses
- *nanosleep\_abs()*, a high-resolution *sleep()* allowing the user to specify when to awaken, rather than how long to sleep;
- Barrier Synchronization, a set of interfaces intended to support efficient implementation of parallel DO/FOR loops on massively parallel computers;

- Reader/Writer Locks, used to allow efficient parallel access to data in situations where reads vastly outnumber writes;
- Spinlocks, a very fast synchronization primitive for use on shared-memory multiprocessors;
- Persistent Notification for Message Queues, an option for 1003.1b-1993 Message Queues.

POSIX.1n, technical corrections to pthreads, is not yet a full approved project, as its Project Authorization Request (PAR) must be approved by the IEEE Standards Board. Approval is likely, and work continues in parallel. POSIX.1n, like POSIX.1i before it, is a fast-track effort with a very limited and specific scope, a similar timescale.

POSIX.13 ballot resolution is nearly complete, and POSIX.13 will be recirculated in September 1995. This may well be the final recirculation. The major changes were due to inclusion of profiles to support POSIX.5b, the Ada bindings to 1003.1b-1993. (POSIX.5b is expected to pass its own final recirculation in a matter of months.) Although POSIX.13 now falls under the new Profiles Working Group, all the POSIX.13 folk came from the former POSIX.4 WG, now SSWG-RT, so we will continue to report on the progress of POSIX.13 here.

ESSE, VSO's "Embedded Systems Software Environment," a new project to write binary, driver, and toolpath standards for embedded computers, is analyzing the current draft of POSIX.13 to see what additional things are suitable for standardization. As POSIX concerns itself only with source-level APIs, while ESSE will consider non-source-level standards, the POSIX and ESSE standards should complement each other. For more information on ESSE, contact the VITA Standards Organization at <http://www.vita.com>. VSO holds the VMEbus Standards, formerly held by the IEEE, and is a part of VITA, the VMEbus International Trade Association, which is largely composed of VMEbus hardware manufacturers.

POSIX.13, 122 pages, is a family of four related realtime profiles ranging in size from the very small through a full-featured platform conforming to essentially all of 1003.1b-1993 (POSIX.1 plus POSIX.4) and POSIX.1c (threads), and the parallel Ada binding POSIX.5b, with realtime options chosen. The smaller profiles specify just that subset of POSIX interfaces needed to "clothe" widely-used small kernels such as pSOS, WindRiver, and VRTX32, and the ORKID interface standard, which, although very similar in function, differ greatly in interface details. (As a matter of interest, there are more of these small kernels in UNIX systems than there are UNIX kernels because, for instance, many I/O controllers and peripherals themselves use one of

these small kernels.) Standardization of these interfaces will yield the same benefits for embedded and realtime as standardization of UNIX did for workstations. In addition, the POSIX.13 interfaces are chosen to allow multi-computer distributed systems to be built, such as those used in factory automation. Such systems are typically set up as a hierarchy, with a few large-profile machines at the top, and a large number of smaller profile machines at the bottom controlling this or that piece of machinery, perhaps with an intermediate layer of supervisory machines between top and base, all communicating with peers, superiors, and subordinates, as needed.

## New Items

A number of new items, as yet un-numbered and homeless, are under consideration: the Trace Interface, and various issues relating to time and clocks. These are described in the following paragraphs.

### Trace Interface

The proposed trace interfaces were the focus of considerable interest and much discussion at the July 1995 meeting. A joint meeting of SSWG-RT and SSWG-FT (the former SRASS WG) was held. A number of specific APIs were proposed, based on this or that existing system, but it became clear in the debates that no one proposal encompassed all major requirements; heretofore, these requirements had been unspoken, and were revealed only in the clash and bang of debate. So, we have backed up a bit, and are developing explicit requirements. We will return to development of specific APIs after the requirements are agreed to.

The purpose of these trace interfaces is to allow the collection and presentation of trace logs of application calls on the operating system, I/O activity, user-defined events, and the like, with an eye to debugging user code running at essentially full speed. This is certainly a requirement for realtime applications. A number of vendors have come forward with implementations, and there has been great interest in coming up with standard APIs and, to some extent, log file contents. The proposed trace interfaces do not in the least resemble standard inspect-and-change debuggers, and require no kernel knowledge to use. The feeling of the vendors appears to be that a trace API would allow them to sell something very much like their existing trace tools beyond their present user base. The details are changing daily; stay tuned.

### Time and Clocks

The addition of some new kinds of clock to the list in IEEE Std 1003.1b-1993, section 14.1.4, `CLOCK_REALTIME`

STANDARDS

and CLOCK\_ABSTIME, are under discussion. The new kinds are tentatively named CLOCK\_SYNCHRONIZED and CLOCK\_FUNDAMENTAL. (These new kinds of clock are defined in earlier reports.) Clock kinds are in effect qualitative descriptions of what an application writer can and cannot expect of a clock; few numerical limits are to be specified.

Under discussion is which clock the various time-related 1003.1b-1993 APIs will follow, where no argument to specify a clock is provided. Relative timers (e.g., wait 10 seconds) will most likely use or act like they use CLOCK\_FUNDAMENTAL, while absolute timers (e.g., wait until 10:00) will likely use CLOCK\_SYNCHRONIZED. The handling of *pthread\_cond\_timedwait()* is an issue, as it uses absolute time to achieve relative semantics.

## Information Infrastructure

by Jim Isaak  
<isaak@ljo.dec.com>

The ANSI (American National Standards Institute) IISP (Information Infrastructure Standards Panel) started in the summer of 1994, and has held four meetings since that time.

This committee was formed at the highest level in the U.S. (ANSI) to bring together formal SDO's (Standards Development Organizations), consortia, industry associations, industry and government organizations to collaborate on delivering the standards needed to advance the U.S. Information Infrastructure. You can find background documentation on this at URLs <http://stdsbbs.ieee.org:7010/pub/ieeestds.htm> (under Information Infrastructure), [gopher://stdsbbs.ieee.org:7010/pub/NII.GII/giiintro.htm](http://stdsbbs.ieee.org:7010/pub/NII.GII/giiintro.htm) and <http://www.ansi.org/home.html>.

While there is much hype on "information infrastructure," there is also substantial progress. This includes evolution and expansion of the Internet, expansion of cable services into education and healthcare, development of video on demand, and significantly increased government awareness and interest in the U.S. and globally. For info on U.S. government activities in this area, see the Information Infrastructure Task Force URL <http://iitf.doc.gov/> or in europe <http://www.earn.net/ECI>. Elements of this will be moved forward independent of any skepticism or hype, and significant players in many diverse industries are positioning their approaches to segments of this market, many of which reflect their various business models. Also, this area involves widely differing views of standardization, spanning collaboratory partnerships, formal standards, and even

legislative and regulatory mandates. The recent IITF (Administration II task force) AC (advisory committee, mainly consisting of industry leaders) has called for NII access by all citizens by the year 2000 (via schools, libraries, etc.) and direct individual access by 2005 (see IITF URL above for "Common Ground" paper).

The "threat" of government interest and leadership is one of the forces encouraging the creation of the IISP. The primary tasks of this group include:

- To bring together the diverse groups of SDOs and consortia, encouraging awareness and collaboration;
- To identify requirements for standardization needed to advance II;
- To identify organizations who have a response (or are willing to develop a response to these requirements);
- To make available to all interested parties (e.g., government, industry) the identified requirements & responses.

To date, a list of 16 requirements have been identified, mostly at a fairly high level (e.g., requirement for network to network interoperability). These are being distributed to the SDO's and consortia to identify groups that have a response or interest in responding.

One fairly well developed example of an area of discussion is the "NII appliance" including "set top device" requirements and interests. There is a pre-existing issue between the TV manufacturers and the cable industry about the set-top device in particular. The cable industry's tradition is to rent these devices, and have control over them (to protect their signal distribution, billing, etc.). Such devices have output on just one channel to the TV. The TV manufacturers seek to put additional functionality into the TV, such as the ability to view two channels at once - which does not work with the traditional set top. Collaboration in this area was mandated by legislation in 1992 through the FCC, and further legislation has been proposed in 1995 that is targeted here as well. (The Blily/Markey bill, related to consumer access to communications appliances) This is in addition to 1995 legislation on reform of the 1934 Telecommunications bill, targeted at allowing competition between various suppliers of telephone and/or other communications (e.g., TV, cable) services to the user's premises. This "convergence" expands the range of affected appliances from set tops to include telephones, TVs, personal computers and more. This example also indicates that standardization in this area is not just

a question of voluntary standards, but is a likely target for legislative and regulatory action.

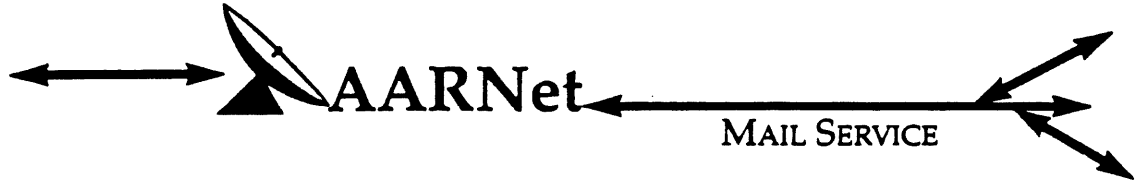
The concept of NII appliances is a useful way to view the direction that work is taking in this area. Traditional standards within industries (e.g., TV, cable, Telco, computer) are being displaced by specifications (and potentially regulation) that will apply to many or all of these industries.

Other key sources of information about Information Infrastructure Standardization include XIWT, who has developed an Architecture document: [http://www.cnri.reston.va.us:3000/XIWT/documents/arch\\_doc/title\\_page.html](http://www.cnri.reston.va.us:3000/XIWT/documents/arch_doc/title_page.html) and CSPP, whose vision and "interoperability" documents on NII are good overviews (but not, alas, online): accessible via phone: 202 662 8407 or email: [cspp@mcimail.com](mailto:cspp@mcimail.com).

Future IISP meetings:

- Oct 30 - Nov 1, Washington DC area
- Jan 17-18, Washington DC area

To join, contact: Chick Hayden, at ANSI; [chayden@ansi.org](mailto:chayden@ansi.org).



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).

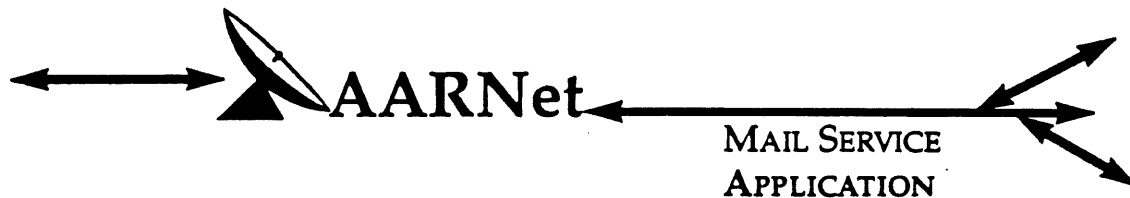
The fee is intended to cover use of AARNet bandwidth for your network traffic. Registration with AARNet, however, provides ONLY the registration of your address in worldwide address tables - your site will be unreachable without this registration. The fee does NOT cover the costs involved in obtaining 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 Catrina Dwyer at the AUUG office on (02) 959 3656 (catrina@swift.sw.oz.au) or myself (frank@atom.ansto.gov.au).

Regards,  
Frank Crawford  
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.

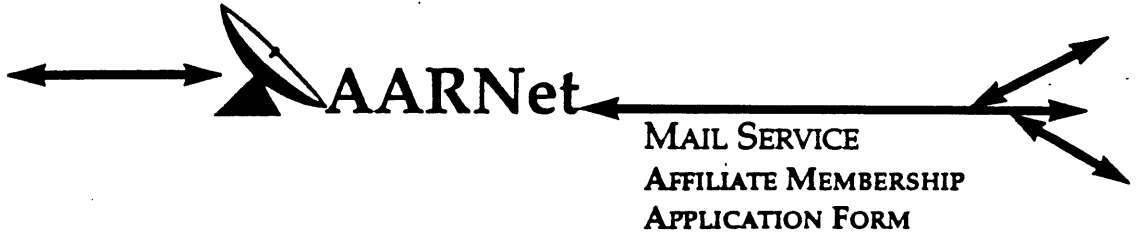
I understand that the AVCC and AUUG Inc maintain the right to vary the Mail Service Affiliate Membership charges from year to year, and maintains the right to cease offering this service to my organisation at the start of any year, at their discretion. I understand that in the event of any variation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





**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 AARNNet (see Note A next page)

Domain Names Requested: \_\_\_\_\_

Gateway Addresses: \_\_\_\_\_

\_\_\_\_\_

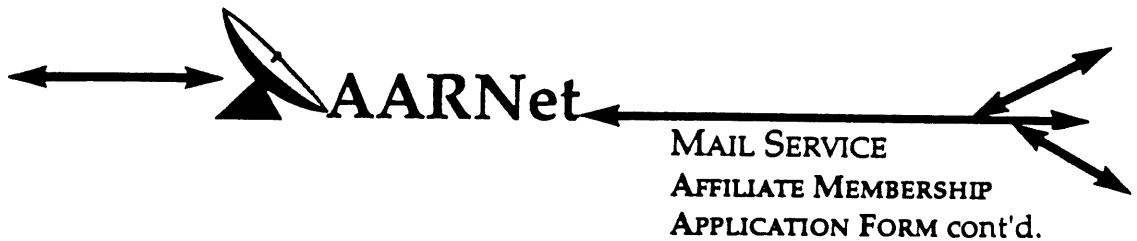
Expected Link Protocol: UUCP SL/IP MHSnet Other: \_\_\_\_\_

\* \* \* \*

Send this page only to:

AUUG Incorporated  
PO Box 366  
Kensington NSW 2033

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



### Note A. Mail Delivery Information

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

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

Example Mail Directory Information request:

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

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

### Note B. Getting Connected

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

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

# MEMBERSHIP APPLICATION

## INDIVIDUAL



UNIX® AND OPEN SYSTEMS USERS

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 or 1 800 625 655 • 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.

**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 air mail  \$ 60.00

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

\_\_\_\_\_  
Signature

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

\_\_\_\_\_  
Date

**LOCAL CHAPTER DESIGNATE:**

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

**TO BETTER SERVE YOU, PLEASE PRINT YOUR CONTACT INFORMATION:**

Name/Contact: \_\_\_\_\_  
Position/Title: \_\_\_\_\_  
Company: \_\_\_\_\_  
Address: \_\_\_\_\_  
Postcode: \_\_\_\_\_  
Tel: BH \_\_\_\_\_ AH \_\_\_\_\_  
Fax: BH \_\_\_\_\_ AH \_\_\_\_\_  
email address: \_\_\_\_\_

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

I, \_\_\_\_\_ certify that  
(administrator)  
\_\_\_\_\_ is a full time student at  
(name)  
\_\_\_\_\_ and is expected to  
(institution)  
graduate approximately \_\_\_\_\_  
(date)

\_\_\_\_\_  
Signature

\_\_\_\_\_  
Title

\_\_\_\_\_  
Date

Over for Institutional Membership

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 or 1 800 625 655 • 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.

We, \_\_\_\_\_  
(Company Name)

do hereby apply for:

Renewal/New\* Inst. membership of AUUG  \$350.00  
 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) 332-4066 or post it to:

AUUG Membership Secretary  
 P.O. Box 366  
 Kensington, NSW 2033  
 Australia



UNIX® AND OPEN SYSTEMS USERS

(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: \_\_\_\_\_

Postcode \_\_\_\_\_

Tel: BH \_\_\_\_\_ AH \_\_\_\_\_

Fax: BH \_\_\_\_\_ AH \_\_\_\_\_

email address: \_\_\_\_\_

**PLEASE PRINT YOUR NEW CONTACT INFORMATION:**

Name/Contact: \_\_\_\_\_

Position/Title: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

Postcode \_\_\_\_\_

Tel: BH \_\_\_\_\_ AH \_\_\_\_\_

Fax: BH \_\_\_\_\_ AH \_\_\_\_\_

email address: \_\_\_\_\_

**AUUG Secretariat Use**

Date: \_\_\_\_\_

Initial: \_\_\_\_\_

Date processed: \_\_\_\_\_

Membership # \_\_\_\_\_



