AUUGN

The Journal of AUUG Inc.

Volume 16, Number 4
August 1995

Background, Issues & Opinion

UNIX Tricks & Traps

Michael Paddon elected AUUG President

... plus reviews, Chapter news, and more

AUUG Inc.

UNIX & OPEN SYSTEMS USERS

ISSN 1035-7521

Print post approved by Australia Post - PP2391500002

AUUGN
Volume 16, Number 4
August, 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*

AUUG Executive

President: Michael Paddon
mwp@munnari.oz.au
Kodak
173 Elizabeth St.
Coburg VIC 3058

Vice President: Glenn Huxtable
glenn@fs.com.au
Functional Software
PO Box 192
Leederville WA 6903

Secretary: Peter Wishart
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
MTIA
509 St. Kilda Road
Melbourne VIC 3004

Committee Members:

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

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

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

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

David Purdue
David.Purdue@aus.sun.com
SunSoft
119 Willoughby Rd.
Crows Nest NSW 2065

AUUGN Business Manager

Elizabeth Egan
auug@auug.org.au
PO Box 366
Kensington NSW 2033

Table of Contents

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

IBM digital Sun

and proudly welcomes the newest of this group:

TELLURIAN PTY. LTD.

Contribution deadlines for AUUGN in 1995

Vol 16, #5 (October): September 22

Vol 16, #6 (December): November 17

AUUGN Correspondence

Please send all correspondence regarding AUUGN to:

AUUGN Editor
PO Box 366
Kensington NSW 2033

E-mail: *auugn@auug.org.au*

Submission Guidelines

Submission guidelines for AUUGN contributions are regularly posted on the **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

Trade marks

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

Editorial

Phil Anderson <phila@osa.com.au>

D'you ever get the feeling that everything's going so darned fast that you just can't keep up with it all? It's not enough that most of us don't get the opportunity to contemplate the infinite; we can't even settle down to nut through the latest software tool or information transfer paradigm, without the whole lot just leaping off and waving to us all to catch up. As the good Reverend Dodgson put it, the current state of play is a veritable Red Queen's Race — you have to keep running faster just to catch up!

Well, in the world of information technology, conference/trade shows such as APWWW/AUUG '95 will go a long way to helping you along the path to enlightenment. This year's conference promises to be an exciting and illuminating event. Even if you can only make it along to a single day, it'll be well worth the effort.

❧

This issue includes a liberal selection of some of the articles published in recent months under the auspices of AUUG in the computer section of the Tuesday *Australian*. Even if you may not agree with the opinions espoused by the respective authors, they touch on a broad range of issues that affect and influence more and more folk each day.

We're interested in your contributions to this most public face of AUUG; drop a line to Frank Crawford (also our erstwhile review editor) if you think you'd like to write something for the *Australian* column.

❧

I'm rushed off my feet this month, so I'll get out of the way, and recommend that you take a look at the first words from our new President, Michael Paddon. All the best for your presidential term!❖

President's Report

Michael Paddon <mwp@sector.kodak.com>

There's been a lot of activity and change in AUUG over the past couple of months (as always). You'll find a lot more detail on some of these in the following pages, but I'll summarise some of the more interesting developments.

As many of you will already be aware, Catrina Dwyer resigned from her position as AUUG business manager in early March. Our business manager faces unique challenges, given the size and scope of AUUG, and Catrina did a superlative job throughout the time she spent with us. I'd like to take this opportunity, on behalf of AUUG, to thank her for her dedication and hard work, and to wish her all the best in her new career.

The executive committee has spent a good deal of time looking for a new business manager. We are very happy to announce that after several rounds of advertising and interviewing we have appointed Elizabeth Egan to the position. By the time you read this, she should be settled into the AUUG office in North Sydney. Welcome aboard, Liz!

There has been a lot going on in the chapters, particularly with the recent tour of Brent Chapman and his firewalls tutorial. This was an enormously successful endeavour, and everyone I spoke to had the highest praise for Brent and his material. Of course, much of this success is due to the efforts of the local chapter organisers; they should be very proud of what they achieved. If you missed Brent's tutorial, rest assured that AUUG will continue to organise similar roadshows on a regular basis.

At this stage of the year, just about all of AUUG's resources are consumed by the upcoming Winter Conference and Exhibition. This year, I had the pleasure of serving on the programme committee, and I don't think I've ever seen a stronger line up of speakers (both Australian and international).

The Exhibition also looks like being more successful (and larger!) than ever before. Most open systems vendors went through a bit of a slump in the past few years with everyone selling pretty much the same software technology in the same sort of boxes. Some even got to the stage where they didn't bother to put a computer on their stand! Now, with the explosive advent of the World Wide Web, the vendors are once

... and now, a word from our new corporate sponsor:

President's report (continued)

again participating in a technology race, and the open systems industry is growing exponentially again.

This year's conference, of course, is special in another way. For the first time, we are holding it in conjunction with another major event: the *Asia Pacific World Wide Web International Conference*. The synergy between the two conferences was immediately obvious to both organising bodies, and I think that attendees will agree that the cooperation has been very fruitful. One of the most immediate advantages is that we have twice as many streams, which means more choice, but perhaps also more angst in selecting sessions to attend. Either way, we are sure you will enjoy and benefit from the experience. I hope to see you there.

It's become clear to many that the Web is not just another fad. Finally, the Internet genie has been let out of the bottle and (much to the disgust of the proprietary networks) changed the networking landscape forever. AUUG has an important role to play in this evolution; it is critical that the lessons of open systems are not forgotten as we enter the brave new world of wide access global networking. It's still not too late for the Internet to be stillborn, and for the market to be dominated by proprietary interests. What a disaster that would be.

Finally, your committee has changed slightly after the elections. Rick Stevenson is departing and we are joined by David Purdue. As always, we are here to make AUUG work better for you, so please don't hesitate to get in touch with anyone on the committee if you have any queries or suggestions.❖

Conferences & Announcements

Announcement and Call for Papers

2nd Symposium on operating systems design and implementation (OSDI '96)

October 29-November 1, 1996 Seattle, Washington, USA Sponsored by the USENIX Association

Co-sponsored by ACM SIGOPS and IEEE TCOS (pending) After a successful first OSDI symposium, the OSDI '96 will again bring together professionals from academic and industrial backgrounds. The focus will be on practical issues related to operating systems for modern computing platforms such as workstations, parallel architectures, mobile computers, and high speed networks. OSDI '96 will emphasize both innovative research and quantified experience in operating systems.

Topics of interest include, but are not limited to:

- OS structure and organization
- OS kernel internals, servers and applications
- Distributed and mobile computing
- Multiprocessor and parallel systems
- Communications Storage Management and I/O systems
- Security in distributed systems
- Scalability and availability
- Heterogeneous systems
- Performance and optimization
- Language support for OS
- OS interaction with HW architecture
- OS support for embedded systems
- OS support for real time
- Multimedia Interaction of OS and applications

The symposium will consist of one day of tutorials, followed by 2.5 days of single-track technical sessions with presentations of the refereed papers, and a half day workshop on a topic yet to be determined. One of the technical sessions will be dedicated to work-in-progress presentations and will be described in later announcements.

The refereed papers will be published in the Proceedings, provided free to technical session attendees and available for purchase from USENIX. The Proceedings may also be distributed to ACM SIGOPS members. Papers of particular merit will be selected to receive an award and will be published in the IEEE TCOS Bulletin.

PROGRAM COMMITTEE

Karin Petersen, Xerox PARC (co-chair)
Willy Zwaenepoel, Rice Univ. (co-chair) Peter Chen, Univ. of Michigan Richard Draves, Microsoft Research Carla Ellis, Duke Univ Ed Felten, Princeton Univ. Jim Gray, Microsoft Bay Area Lab. Kevin Jeffay, Univ. of North Carolina David Johnson, CMU Jay Lepreau, Univ. of Utah Jeff Mogul, DEC WRL Marc Shapiro, INRIA John Wilkes, HP Labs John Zahorjan, Univ. of Washington

IMPORTANT DATES

Full papers due: May 7, 1996 Notification to authors: July 30, 1996 Revised papers due for shepherding: August 19, 1996 Camera-ready full papers due: September 16, 1996

SUBMISSION PROCESS

We seek papers describing original work concerning the design, implementation and use of modern operating systems. Besides mature work, we encourage submissions describing exceptionally promising, well-grounded speculative work, or enlightening negative results. Papers will be judged on significance, originality, clarity, relevance, and correctness. The committee will favour papers with reproducible results, especially those supplying detailed data and explanations, or offering to make data sets or source code available.

Authors are required to submit full papers, no longer than 14 pages, single- or double-column format. Very similar papers must not have been published or submitted for publication elsewhere.

All submissions will be held in the highest confidentiality prior to publication. Accepted papers will be shepherded through an editorial review process by a member of the program committee.

Authors of accepted papers will be expected to provide an HTML page containing the abstract and links to their paper, slides, and software, if available. This will be collected after the event for inclusion in an electronic version of the symposium.

WHERE TO SUBMIT

Submission of all papers must be made in both paper and electronic form. Fifteen (15) paper copies (double sided if possible) of the paper must be sent to: Willy

Zwaenepoel Department of Computer Science Rice University 6100 S. Main St. Houston, TX 77005, USA and one electronic copy in Postscript (not ASCII) must be submitted by electronic mail to: osdi-**papers@cs.rice.edu**

For administrative reasons (not blind reviewing), every submission (in both its paper and electronic form) should include an additional cover sheet containing:

- paper title and authors, indicating any who are full time students, and
- for the author who will act as the contact to the program committee, his or her name, paper mail address, daytime and evening phone numbers, e-mail address and fax number, if available.

The cover sheet mailed with the electronic paper submission should be in ASCII to facilitate accurate on-line bookkeeping and should be included in the same electronic mail message as the PostScript file containing the paper. All submissions will be acknowledged. If your submission is not acknowledged by May 21, 1996, please contact the program chairs promptly at **osdi@cs.rice.edu**.

REGISTRATION MATERIALS

Materials containing the OSDI technical and tutorial programs, registration fees and forms, and hotel information will be available August 1996. If you wish to obtain this information, please contact the USENIX Conference Office; details can be found at the end of these announcements.

Announcement and Call for Participation:

USENIX 1996 Technical conference

January 22-26, 1996, San Diego, CA

The 1996 USENIX Technical Conference Program Committee seeks original and innovative papers about the applications, architecture, implementation, and performance of modern computing systems. As at all USENIX conferences, papers that analyse problem areas and draw important conclusions from practical experience are especially welcome. Some particularly interesting hot application topics are:

- Privacy and cryptography - Compression applications - User interface toolkits - Nomadic and wireless computing - Networks and distributed computing - Security

A major focus of this conference is operating systems practice and experience. We seek papers describing original work and results in the design, implementation, and use of modern operating systems. Submissions describing extensions or modifications of complete and widely used operating systems are particularly encouraged in addition to those describing research systems or prototypes. Topics of interest in this area include but are not limited to:

- OS structure and organization
- Performance and optimization
- OS support for real time & multimedia
- OS support for embedded systems
- OS interaction with HW architecture
- Microkernel internals, servers, and applications

Note that the USENIX conference, like most conferences and journals, requires that papers not be submitted simultaneously to more than one conference or publication, and that submitted papers not be previously or subsequently published elsewhere. Papers accompanied by non-disclosure agreement forms are not acceptable and will be returned to the author(s) unread. All submissions are held in the highest confidentiality prior to publication in the Proceedings.

DATES FOR REFEREED PAPER SUBMISSIONS

Extended Abstracts or Manuscripts Due: July 18, 1995 Notification to Authors: August 31, 1995 Camera-ready Papers Due: November 14, 1995

CONFERENCE SCHEDULE OVERVIEW

Tutorials: January 22-23, 1996 Technical Sessions: January 24-26, 1996 Birds-of-a-Feather Sessions: January 23-25, 1996 USENIX Reception: January 24, 1996 Vendor Display: January 24-25, 1996

PROGRAM COMMITTEE

Robert Gray, Program Chair US WEST Miche Baker-Harvey Digital Equipment Corporation David Black Open Software Foundation Matt Blaze AT&T Bell Laboratories Darrell Long University of California at Santa Cruz John Ousterhout Sun Microsystems Christopher Small Harvard University Jonathan Smith University of Pennsylvania Carl Staelin Hewlett-Packard Brent Welch Xerox PARC

HOW TO SUBMIT A REFEREED PAPER

It is important that you contact the USENIX Association office to receive detailed guidelines for submitting a paper to the refereed track of the technical sessions. Please telephone 510-528-8649, or send email to: **usenix96authors@usenix.org**.

In addition, specific questions about submissions to the USENIX 1996 Technical Conference may be made to the program chair via email at: gray@usenix.org.

The program committee will review full papers or extended abstracts. An extended abstract should be five manuscript pages (single-sided) or fewer in length. It should represent the paper in short form. If the full paper has been completed, it may be submitted instead of an extended abstract. Full papers should be limited to 12 single-spaced pages.

Include references to establish that you are familiar with related work, and, where possible, provide detailed performance data to establish that you have a working implementation and measurement tools.

WHERE TO SEND SUBMISSIONS

Please send one copy of an extended abstract to the program chair via one of the following methods. All submissions will be acknowledged.

Preferred Method: email (PostScript or ASCII) to: **usenix96papers@usenix.org**

Alternate Method - Postal Delivery to: Robert Gray U S WEST Technologies 4001 Discovery Drive, Suite 280 Boulder, CO 80303 Phone: 303 541 6014

The authors must also submit via email to **usenix96papers@usenix.org** the following information:

- The title of the manuscript and the names of the authors.
- The name of one author who will serve as a contact, their email address, day and evening phone numbers, postal mail address, and a fax number, if available.
- An indication of which, if any, of the authors are full-time students.
- A short abstract of the paper (75-150 words).

CASH PRIZES

Cash prizes will be awarded for the best paper at the conference and the best paper by a full-time student.

TUTORIALS

On Monday and Tuesday, you may attend intensive, immediately practical tutorials on topics essential to the use, development, and administration of UNIX and UNIX-like operating systems, windowing systems, networks, advanced programming languages, and related technologies. The USENIX Association's well-respected program offers you both introductory and advanced courses, presented by skilled teachers who are hands-on experts in their topic areas.

USENIX will offer two days of tutorials covering topics such as:

- System administration
- Systems and network security
- Distributed computing: DCE, DFS, RPC, CORBA
- Kernel internals: SVR4, Chorus, Windows NT
- Systems programming tools and program development
- Portability and interoperability
- Client-server application design and development
- Sendmail, DNS, and other networking issues
- GUI technologies and builders

To provide the best possible tutorial slate, USENIX constantly solicits proposals for new tutorials. If you are interested in presenting a tutorial, contact the Tutorial Coordinator:

Daniel V. Klein Phone: 412 421 2332
Email: **dvk@usenix.org**

INVITED TALKS

Coordinators: Mary Baker, Stanford University Ed Gould, Digital Equipment Corp

An Invited Talks track complements the Refereed Paper track. These talks by invited experts provide introductory and advanced information about a variety of interesting topics, such as using standard UNIX tools, tackling system administration difficulties, or employing specialized applications. Submitted Notes from the Invited Talks are published and distributed free to conference technical sessions attendees. This track also includes panel presentations and selections from the best presentations offered at 1995 USENIX conferences and symposia.

The Invited Talks coordinators welcome suggestions for topics and request proposals for particular Talks. In your proposal, state the main focus, include a brief outline, and be sure to emphasize why your topic is of general interest to our community. Please submit via email to: **ITusenix@usenix.org**.

WORK-IN-PROGRESS REPORTS

Do you have interesting work you would like to share, or a cool idea that is not ready to be published? Work-in-Progress Reports are for you! Work-in-Progress Reports, scheduled during the technical sessions, introduce interesting new or ongoing work. The USENIX audience provides valuable discussion and feedback. We are particularly interested in presentation of student work. To schedule your report, contact Peg Schafer via email at **wips@usenix.org**.

BIRDS-OF-A-FEATHER SESSIONS

The always popular Birds-of-a-Feather sessions (BOFs) are very informal gatherings of persons interested in a particular topic. BOFs often feature presentations or demonstrations followed by discussion, announcements, and the sharing of strategies. BOFs are offered Tuesday, Wednesday, and Thursday evenings of the conference. BOFs may be scheduled at the conference or in advance by telephoning the USENIX Conference Office at 714 588 8649 or via email to: **conference@usenix.org**.

VENDOR DISPLAY

In the USENIX Vendor Display, emphasis is on serious answers from technically savvy vendor representatives. Vendors will demonstrate the technical innovations which distinguish their products. Here, in a relaxed environment, conference attendees can ask questions and discuss how something will work with what they already have. Plus, you can review the newest releases from technical publishers.

Vendors: This is an exceptional opportunity for receiving feedback from USENIX's technically astute conference attendees. If your company would like to display its products and services, please contact: Zanna Knight. Telephone: 510 528 8649 FAX: 510 548 5738 Email: **display@usenix.org**

CONFERENCE PROGRAM & REGISTRATION INFORMATION

Materials containing all details of the technical sessions and tutorial program, conference registration, hotel and airfare discounts, and reservation information will be available at the end of September 1995. If you wish to receive the registration materials, please contact: USENIX Conference Office 22672 Lambert St., Suite 613 Lake Forest, CA USA 92630 Phone: 714 588 8649 Fax: 714 588 9706 Email: **conference@usenix.org**

The USENIX 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 Technical Conferences and symposia have become the essential meeting grounds for the presentation and discussion of the most advanced information on new developments in all aspects of computing systems.

The USENIX Association and its members are dedicated to: - problem-solving with a practical bias, - fostering innovation and research that works, - communicating rapidly the results of both research and innovation, and - providing a neutral forum for the exercise of critical thought and the airing of technical issues.

SAGE, the System Administrators Guild, a Special Technical Group within the USENIX Association, is dedicated to the recognition and advancement of system administration as a profession.

THE USENIX ASSOCIATION 1995/1996 CALENDAR OF SYMPOSIA AND CONFERENCES

January 22-26, 1996 USENIX 1996 TECHNICAL CONFERENCE

San Diego Marriott Hotel, San Diego, California
Program Chair: Robert Gray, U S WEST Technologies

The 1996 USENIX Technical Conference Program Committee seeks original and innovative papers about the applications, architecture, implementation, and performance of modern computing systems for presentation during the Technical Sessions. Papers that analyse problem areas and draw important conclusions from practical experience are especially welcome.

A major focus of this conference is operating systems practice and experience. We seek papers describing original work and results in the design, implementation, and use of modern operating systems. Submissions describing extensions or modifications of complete and widely used operating systems are particularly encouraged in addition to those describing research systems or prototypes.

Contact USENIX to receive guidelines for submitting a paper to the technical sessions; please telephone +1 510 528 8649 or email to: usenix96authors@usenix.org. The Invited Talks track of the Technical Sessions complements the Refereed Paper track. The Invited Talks and panel presentations provide introductory and advanced information about a variety of interesting topics.

USENIX' well-respected tutorial program on January 22-23 will offer both introductory and advanced courses, presented by skilled teachers who are hands-on experts. The evening Conference Reception is set for January 24 and a Vendor Display will be held January 24-25.

May 13-17, 1996 5th UNIX SYSTEM ADMINISTRATION, NETWORKING, AND SECURITY SYMPOSIUM (SANS V)

Sponsored by FedUNIX, Open Systems Conference Board, and SAGE, the System Administrators Guild Washington DC Program Chairs: Rob Kolstad and Michelle Crabb

September 30-October 4, 1996 10th USENIX SYSTEMS ADMINISTRATION CONFERENCE (LISA '95)

Co-sponsored by SAGE, the System Administrators Guild Chicago Marriott, Chicago, Illinois Program Chairs: Helen Harrison, SAS Institute and Amy Kreiling, University of North Carolina

October 29-November 1, 1996 2nd SYMPOSIUM ON OPERATING SYSTEMS DESIGN AND IMPLEMENTATION (OSDI '96)

Seattle, Washington Program Chairs: Karin Petersen, Xerox PARC and Willy Zwaenepoel, Rice University Co-sponsored by ACM SIGOPS and IEEE TCOS (pending)

OSDI '96 will again bring together professionals from academic and industrial backgrounds. The focus will be on practical issues related to operating systems for modern computing platforms such as workstations, parallel architectures, mobile computers, and high speed networks.

OSDI '96 will emphasize both innovative research and quantified experience in operating systems. The symposium will consist of one day of tutorials, followed by two and a half days of technical sessions with presentations of the refereed papers, and a half-day workshop on a topic yet to be determined. One of the technical sessions will be dedicated to work-in-progress presentations.

Refereed papers will be published in the Proceedings, provided free to technical session attendees and available from USENIX. Proceedings may also be distributed to ACM SIGOPS members.

Papers of particular merit will be published in the IEEE TCOS Bulletin. The Program Committee is seeking papers describing original work on the design, implementation and use of modern operating systems.

Besides mature work, we encourage submissions describing exceptionally promising, well-grounded speculative work, or enlightening negative results. For submission guidelines, please contact the program chairs at osdi@cs.rice.edu.

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

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. 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

IMPORTANT DATES

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

Background:

Internet-based Electronic Commerce will give us 'eye-level' access to world markets

Phil McCrea <pmc@syd.dit.csiro.au>
Industry Liaison Manager
CSIRO Division of Information Technology

For centuries, the European and North American maps have shown Australia to be at the bottom right hand corner of their world maps. Australian maps these days, of course, have Australia and the Asia–Pacific zone in the centre of world maps. This not only gives us a better perspective on the world at large, it does identify us fairly and squarely as part of an Asia–Pacific trading block.

The Internet will dramatically alter this view of where Australia fits in the world of commerce, because it essentially removes any form of geographical isolation whatsoever. There is an interesting analogy with supermarket shelves here: vendors of grocery products always want supermarket managers to put their products at eye level, because products displayed at eye level sell better than those elsewhere on the shelves. The Internet will effectively place Australian companies at 'eye level' in a global market.

How will it all work? There are three main components:

Use of the World Wide Web for advertising.

The Web has become the new publishing tool of the 90s. A recent Infosys report shows that 90% of the US Fortune 500 companies plan to use the Web as a strategic part of their advertising programmes. Web 'browsers' provide full multimedia support, and can be used as an advertising medium as an alternative to magazines, mail order catalogues, and junk mail brochures. Furthermore, because browsers support audio, sound clip samples from CDs can be broadcast over the Internet.

The Web is a broadcast medium that cannot be ignored by anyone. It will be used more and more over the next few years to advertise goods and services.

Use of Email for ordering

Electronic Commerce requires that there be an electronic link from purchaser back to vendor be in place. And not just any old link - it must be a secure one, so that eavesdroppers cannot read credit card numbers. Furthermore it must be able to indicate that the purchaser is who they claims they are (authentication).

The necessary technology for all this exists, but has yet to be "commoditised" so that it can be used by everyone, regardless of what software environments they are using. Fortunately the two main credit card companies, Visa and Mastercard, have agreed to a common security protocol, which implies that the rest will follow their lead.

Credit Card Purchasing

Most organisations carry out business with established partners, and settle on a monthly basis. While this model certainly can be used over the Internet, it is likely that more use will be made of credit card purchasing. Many organisations now actually prefer the use of credit cards for purchasing, and regard the credit card fee as an acceptable cost for the convenience and for the itemized records supplied by the credit card provider.

Likewise vendors generally prefer credit card purchasing, because it effectively means payment up front - and guaranteed payment at that. The Australian Government is moving towards greater use of credit cards for purchasing for their own use.

The use of credit card purchasing over the Internet will become commonplace as soon as Web browsers are supplied with the appropriate security mechanisms.

Delivery of Goods

The final stage is delivery of the products that have been ordered, and this role is carried out by courier companies. There are a number of worldwide courier companies that are trusted by most organisations - several of these being Australian owned. In other

continued page 15

Reflection:

Thoughts of an Outgoing President

Phil McCrea <pmc@syd.dit.csiro.au>
Industry Liaison Manager
CSIRO Division of Information Technology

Having just completed a three year term as AUUG President, I beg your indulgence to reflect on some of the events that have occurred in that period. It's been a great three years - although sometimes my employer(s) haven't been quite so convinced... Let me list some of the highlights, in no particular order.

My first annual conference as President was AUUG 92 in Melbourne. The conference was well under way when I came on board as President, but for some reason the public relations campaign went terribly wrong. Like most organisations or companies, AUUG uses a PR organisation to get us on side with journalists and the like. It just fell apart in 1992, with the only press report being a few disparaging lines by Susan Coleman in Pacific Computer Weekly along the lines of 'The annual meeting of UNIX boffins took place last week in Melbourne...'. This made me determined to work on our PR for the following year. I'm pleased to report that we have mastered the art of public relations, and AUUG is now well known, and (dare I say it) almost part of the establishment!

In terms of an IT event, the annual AUUG conferences and exhibitions are a major feature of the IT year. They are now keenly waited for by journalists, and esteemed broadsheets have begun to have an Open Systems supplement to coincide with the AUUG events - thanks Jeremy! Keep tuned for another such supplement in The Australian to coincide with AUUG '95 in September.

Speaking of the Australian, we are indeed grateful to Jeremy Horey, the editor of Tuesday's Computing section, for allowing us to run this column. I know it gets widely read, for I am frequently bowled up at a function or in the street, by people I know - and often ones I don't know - to say that they had read such-and-such in The Oz.

Another feature has been the establishment of Chapters in all most capital cities. There had been regional AUUG activities in Perth (under the name 'WAUG'), Melbourne (under the cute name 'SESSPOOLE', standing for South Eastern Suburbs Specialising in Programming and....), Canberra, and Brisbane. These were formalised into Chapters of AUUG, and new Chapters were subsequently formed in Sydney, Adelaide and Darwin. We have recently been having discussions in relation to a Chapter in Hobart.

We have held several successful tours by well known luminaries from the UNIX world: Kirk McKuisick, Gene Spafford, Linus Torvalds, and most recently Brent Chapman. Unfortunately we are constrained, from the perspective of the Exhibition, to hold our annual get-togethers in either Sydney or Melbourne, and as a result it is more difficult for our members in other states to attend. Organising roadshows by overseas experts is one way to provide value to AUUG members in other states.

The role of AUUG has changed over the past few years. Ten years ago it used to be mandatory for AUUG members to be bearded (assuming the appropriate gender!), and wear sandals, jeans, and T-shorts with cryptic messages that only UNIX neophytes would understand. It appeared to be necessary for each AUUG conference to contain a gaggle of papers on how to write device drivers! I think AUUG 94 may have been a milestone by virtue of the fact that there were no papers on device drivers!

It is gratifying to observe that many of these seasoned AUUG dignitaries from years past are still members. True, the hair may be more grey, the middle aged spread may be more noticeable, and jeans may have been replaced by suits in some cases (!).

Probably the biggest change that has occurred to AUUG over the past few years has been the broadening of its focus. When AUUG was formed over 20 years ago (yes, it's been that long!) the main focus was very much on the internals of a new operating system that had suddenly become popular in universities. Why was this operating system popular? For the simple reason that it was the first operating system to be written in a readable language - C. People could 'read' the operating system, and hence it made a valuable teaching vehicle in Universities. Furthermore, the ready availability of C compilers for different computers meant that UNIX could run on lots of different computers. But that is all history.

In due course UNIX became the standard for 'openness' in computing - and it still is. Everything became 'open' - even AUUG changed its full name to 'The Australian UNIX and Open Systems User Group', although the AUUG initials still remain. As a result, AUUG attracted many new members who were not interested in kernel hacking, but were more interested in marketing, sales support, systems development, and so forth.

In just about every respect the Open Systems war has been won. All operating systems (except those from Microsoft...) are now looking like UNIX - at least from the outside. We have survived the heady days of the UNIX wars, where the UNIX International alliance fought the warriors from the Open Software Foundation (quaintly termed the 'cabal of the devil' by Gareth Powell!).

So, are Open Systems finished? Not on your life! The battle has merely moved to a different front. The 'enemy' used to be embodied by companies like IBM with their proprietary operating systems. "Don't get locked into Open Systems" was the oft heard cry of the IBM salesman! But how things have changed! MVS and OS400 are virtually Spec11/70 compliant, which means they may actually in due course be allowed to be called UNIX!!

The new battle is the networks front. UNIX's networking software, TCP/IP, is now the standard for openness in the networking space. The Internet is entirely based on TCP/IP, and is also embodied with the same spirit of openness which UNIX itself brought to the computing platform. This has been embraced by AUUG, with the result that the theme for AUUG 95 is very much on the Internet. The last thing we need is for proprietary interests to commandeer the networking 'space'.

In conclusion, thank you one and all for your assistance over the past three years. Thanks to the 700 or so individual members of AUUG and the 350 odd institutional members, who have enabled AUG to survive and grow. And a special thanks to our four corporate sponsors - Sun, IBM, DEC, and Tellurian. I wish the new President, Michael Paddon, all the best.❖

Feedback

Andrew Tune <Andrew.Tune@technix.com.au>

An article in the June 1995 *AUUGN*, "A Better Way to Access Files - Mapping" presented a number of points in regard to the use of the **mmap()** system call. I would like to take something of an opposing view, and to explain my reasons why.

I take issue with the article at four levels: it is one-sided, it is technically inaccurate or at least unclear in several places, it makes no mention of standards-conformance, and it appears to pay no attention to portability. Firstly, the technical inaccuracies or technically unclear areas. Under the heading "File Mapping for I/O" the article describes the process of reading a 16MB file, and how this involves "all 16MB" being brought into memory, copied, and how it "will require being written back to the paging device *as it has been modified*. All before you have touched the first byte" (my italics). This is incorrect. Firstly, the use of the paging device is system-dependent, even amongst UNIX systems. Secondly, blocks will only get written back if they've been changed. Thirdly, only those blocks whose data is actually accessed get read into memory.

In the next paragraph is the statement that the use of **mmap()** means that "you don't use twice the memory". The use of **read()** and **write()** does not double the memory required at all: the disk buffer cache will use a configurable amount of memory only. The article goes on to state that "if you don't modify the page, the system doesn't need to write it back". The reality is that what gets written back is largely under application control in any case, and that no reasonable VM system will write back clean pages in any case.

My greatest concern, however, lies in the impression which a less experienced reader may have gained from reading the article, that **mmap()** is a better alternative to more conventional methods of file access. This it is, but only in particular circumstances. For example, it is not possible to use **mmap()** with non-regular files in general (devices, pipes, etc.). This means that you can't use **mmap()** in programs intended to operate on arbitrary files, or for that matter in filters. Filters (programs which read stdin and write stdout) are amongst the oldest and most distinctive features of UNIX. If we all used **map()**, we wouldn't have filters! The view presented of the very real benefits of **mmap()** gives no coverage to the disadvantages. For example, different systems provide different levels of support. Sun's support is quite comprehensive. HP supports it on some hardware platforms and not others. SCO doesn't contain **mmap()** at all! For people writing real-world code, this sort of thing is important. Furthermore the use of **mmap()** to sequentially read large files will upset VM algorithms on some systems.

Finally, for the great majority of applications programming, we should all be paying more attention to standards. The **mmap()** call may be a "staple part of SVR4", but it's not in P1003.1 (the POSIX standard) and it's not in XPG4! The use of **fread()** and **fwrite()**, despite their inefficiencies in some circumstances, makes good sense for most situations, and certainly from the point of view of portability.

While **mmap()** has its application, a generalised "better way to access files" it is not!❖

Background:

Professionals On-Line: Australian IT consultants on the Web

Ian Taylor <ian@wordsimages.com>
<http://www.wordsimages.com/>

Why bother using placement agencies if you can scan an on-line database to find the people you need? Or, if you're a consultant and need additional expertise for a project, why limit yourself to your local area? Why not use the same on-line database to locate expertise and work virtually across the Internet?

Because such on-line databases didn't exist. Well now one does. **Professionals On-Line** is an on-line database of Australian IT consultants. The first of its kind on the Web, still very much in the prototyping stages, it is free to its users.

What do you need to get onto Professionals On-Line? To be in the database you need an Internet or Compuserve email address. To use the database to find fellow consultants or people you might wish to hire, you'll need a web browser such as Lynx, Mosaic or Netscape, which you point at:

http://www.webcom.com/~wordimg/pro/

It works as follows: Each consultant on Professionals On-Line has his or her own Web page. The page is set up as a fixed-format fact sheet and is created when the consultant returns an e-mailed questionnaire or fills out an on-line form. The fact sheet contains summary information about skills, experience, education, preferred work locality, availability. It does *not* contain information on age, gender, marital status or other equally peripheral issues.

Professionals On-Line uses GLIMPSE keyword search. For example, entering "project management; Brisbane" returns a results page listing all consultants whose fact sheets contain the words "project management" and who have "Brisbane" as their preferred work locality (the semicolon is an AND operator). The results page also shows the lines from the fact sheets that contained the search words. So with the GLIMPSE results page, you not only get matches, you get some context as well. This is important and I'll come back to it in a moment. From within the results page, the user simply selects/clicks on a consultant's name to jump to that consultant's fact sheet. From within the fact sheet, the user can then email the consultant, download CVs or description of services, or jump to the consultant's own web pages, if available.

Coming back to the idea of context, in Professionals On-Line information is supplied in a series of layers: search results with summary information, fact sheets with more detail, detailed descriptions of services or other web pages. In my opinion, this is how WWW search information should be presented. It's very hypertext; you descend to increasing levels of detail only if you wish.

If contacted by a company as a result of an on-line query, the consultant and company negotiate directly (although, of course, the consultant could pull in an agency to negotiate for him if he or she prefers). For companies and consultants this has two clear advantages: Financial: no agent commissions or ongoing payments; Business: consultants can build up their own direct clientele.

As important, perhaps more important, is the ability to network with fellow consultants. You can use the Professionals On-Line search tool to find people with complementary skills. You can pass work on to fellow professionals. To assist in this, a mailing list is being set up by two POL users, Nathan Bailey and Bruce Lyon. You can set up (virtual) teams for projects. Professionals On-Line will give you the ability to search through a vast range of skills, locate the expertise you need for a project, wire those experts in for that project, and then set up a new team (rewire) for the next.

Professionals On-Line has links to specimen CVs and to Australian IT job servers, advice on virtual associations and corporations, links to international resume services. Still on the vapourware side are specimen contracts, tax advice, pointers to good training software, CVs and brochures in document exchange format.

The state of play is that there are already over 100 consultants on the site with almost as many different skills--it really is amazing how diverse IT is, and how many different kinds of specialist knowledge there are. The idea has been received enthusiastically and some good suggestions for improvements have been made and implemented. It's received some coverage in the Australian computer press with mentions in *PCWeek*, the *Sydney Morning Herald*, and July's *LAN*

Magazine, to name a few. The site won't start to make waves until there are at least 250 consultants on it; 500 is better. It's a long way from reaching critical mass.

Where does it go from here? Well Australia would be a good start! The site's still based in the States for the simple reason that Australian ISPs don't provide the tools I need. When they do, at an affordable rate, I'll move it here; by that time, setting up my own server may be an option and hell will probably have frozen over. This has its frustrations and restricts me from developing the site and the idea as fast as I'd like. For that I'd need multiple search tools and the ability to add some CGI extensions. As Professionals On-Line gets bigger, I'm hoping some form of sponsorship might let me set up a dedicated Australian server.

What's in it for me? Well, one, it's a good idea and I like pursuing good ideas. I also get publicity for my technical writing and electronic publishing company, Words & Images. I get web-building work from people who realise that it's ideas and content, not this week's bells and whistles, that make web sites. I get contacts in all areas of IT, which will help me grow my business. So it's not altruism: I get a lot out of it. If Professionals On-Line gets big enough, we all will.❖

from page 11

words there are trusted delivery mechanisms for worldwide distribution of goods ordered over the Internet.

What does all this mean for Australia?

Putting all the above together, it is quite possible for an small company in Tasmania (say) that produces a niche product, to advertise over the Web at very little expense. Furthermore the cost of Web based advertising to cover Hobart, is the same as the cost to cover the entire world. Once this company has captured the attention of a purchaser in North America (say), that purchaser can then order and request delivery using payment and delivery mechanisms that are known and trusted.

The impact of this new method of carrying out business will have an enormous impact on Australian business, and on our exports in particular. As a country, we are well equipped to take advantage of Internet - based commerce, and we have a window of opportunity that we must not pass up.❖

Background:
Sun, Tk + Java

Jamie Honan

There used to be a whole industry of consultants, journalists and general conference-going hangers-on who could be classed as IBM watchers. The latest rumours concerning the directions of 'Big Blue' were dissected somewhat in the manner of ancient soothsayers discerning meaning from the entrails of chickens.

These days the gloss has slipped from IBM. Today, the pronouncements of some underling at Microsoft are jumped on and studied as if pearls of wisdom.

Sun Microsystems is a little easier to divine, even if it's pronouncements are not regarded as earth shattering. Sun's senior technology officers have made statements to the effect that Sun regards the Internet and Internet technology as a major field in which it wishes to participate.

The direction this will take is easily seen from several very public projects that Sun is undertaking.

Tk

Sun has hired John Ousterhout, the author of the Tcl / Tk toolkit, to continue work on his freely available package. Tcl stands for Tool Command Language - a scripting language that is very easy to embed in a C program. A program that incorporates Tcl provides a very powerful and flexible means of scripting.

Tk is an extension to Tcl that provides windowing capabilities. Most windowing systems, X Windows or MS-Windows for example, have a very complicated programming interface. This complication means that windowing programs are very difficult to write. Generally, programmers require a long time to train up to write for these systems, and the resulting programs are large and cumbersome.

Tk is a very clever mechanism for building and using the interface components of X-Windows (the technical term is widgets, I kid you not) in a simple and powerful manner. As an example of how simple the system is, the games of Minesweeper and Solitaire (which are instantly recognisable to any MS-Windows user) can each be built in Tk in less than 100 lines of code.

Since joining Sun, Dr. Ousterhout has extended his Tk system considerably. The system is still freely available, even to Sun's competitors! Check the newsgroup comp.lang.tcl for more information. The direction that Tk has taken has been very much oriented towards hypertext viewers and Web browsers. As a small example, the latest beta test versions of Tk have in-built support for pictures, especially the embedded sort seen in Web pages.

Sun has recognised that platform independence is one of the keys to success with Internet tools. Dr. Ousterhout has promised to work towards supporting Tk on Macintoshes and MS-Windows systems. There are primitive versions already available for these two platforms.

Java

Another major Sun project, again very visible, is Java and the HotJava browser. Java is a language rather like C++. Unlike Tcl, which is a pure scripting language and is not compiled, Java compiles to 'byte-codes'. These byte-codes are then interpreted by a small 'engine'. The advantage of such a system is that the compiled code can be run on many dissimilar systems, just by providing a different engine.

The byte-codes are more efficient to execute than Tcl scripts. The proponents of Java demonstrate how a Web browser based on Java could download code to interpret data that the browser didn't know how to handle, for example pictures in a non-standard format. Of course, this immediately raises security issues, but these issues also seem to have been addressed by Java's designers.

Java has very strong run-time checking. It should be possible to 'corral' foreign code and set it strict boundaries. For example, Java implements run-time array bounds checking.

There are many aspects of Java that make it worth studying. It seems to overcome many of the weaknesses of C++, while adding valuable features. Java implements garbage collection, for example, an invaluable method of automatically handling the allocation and release of memory. 'Memory leaks', the result of getting this wrong, are one of the biggest bugbears of large scale programming in C and C++.

More information can easily be obtained from the Sun's repository at **http://java.sun.com**.

With Tk and Java, one can envisage 'active' Web pages and email messages. One example is an extension to Tcl, called Safe-Tcl, that was developed by the founders of the MIME standard (the format for passing multimedia messages via email). The example that comes with this extension has a number of buttons you can press to hear the author's daughters extolling the virtues of their girl scouts 'cookies'. This cute example then leads on to take your order and email it back to the author.

This playful example is the essence of a very business oriented system that Sun obviously hopes to build in the future.❖

Issues:
Who is doing your System Administration?

Frank Crawford <frank@ansto.gov.au>

Talk to any computer salesman, whether they are selling a PC or a supercomputer, and they will tell you that their system needs no administration, or that their systems are so simple that any one can do it. If a car salesman told the same sort of story you would not believe it, yet many people seem to believe it from a computer salesman.

As is soon obvious to anyone buying a computer, some sort of system management is needed, be it changing the PATH definition, configuring a new disk or optimising the system. It is obviously not a full time job to administrator a single PC, or even a simple multi-user system, but as the number of connected systems grow or the complexity of the multi-user system increases, the point is quickly reached where it pays to employ a person to do the job.

Unfortunately, it is often too easy for management to ignore the need for an administrator, especially in an organisation where the number of computers has grown from small numbers. After all, there is no administrator and yet everything is working fine. However, in reality it is not true. There may be no official position for an administrator, but there is someone doing the job. Often this person just happens to be someone who knows more about computers than his colleagues, and is performing these administration duties on top of his other role, and often to the detriment of both.

Before considering how to properly staff such a role, it is worth considering what it entails. Initially, it starts with performing backups and giving advice on occasional problems, but it soon moves to performing software installations and upgrades, configuring and installing new systems and informally planning system growth within the organisation. As the site becomes more complicated the knowledge required quickly outstrips the capabilities of an amateur. Even worse, often this support exceeds the desire to give it, as there is no recognition for the work, and even less time for their official role.

For a large organisation, there is no problem employing a professional administrator, but for a smaller organisation, it isn't quite as clear cut. There is an "obvious" need for everyone to be "productive", and it is difficult to justify the extra overhead. If, however, it is considered from the perspective of what is best for the organisation, then it becomes much easier to justify. Firstly, the time already lost by whoever is currently performing the role is significant. Secondly, the time saved by a professional approach to much of the work gives many productivity benefits, and finally, the saving in time no longer lost due to proactive planning all add to considerable benefits to an organisation.

The issue that comes up is what is meant by a "professional" administrator. The first requirement is technical knowledge; they must be both sufficient and current. This is often the easiest item to test as many vendors are now offering certificate courses, e.g., Certified Netware Engineer from Novell, SCO Certified Administrator, etc. However, this is not always available and should be tempered by an evaluation appropriate to your company. It is also necessary to ensure that this has been kept current.

Secondly, it is necessary that they have experience, although the level of experience often affects the expected salary. Certainly, there needs to be room for new administrators to gain experience, however, in critical situations, there is no substitute for such experience, no matter how much training. This is something that needs to be judged by the organisation.

Finally, it is also necessary to ensure that such an administrator has a professional attitude. Often they have access to corporate information due to the nature of their role. For example, the ability to circumvent database security or access any users documents, as they are the "computer security manager" for site. Within a large organisation, it is possible to put procedures in place to ensure that administrators do not exceed their responsibilities, but in a small organisation, often these roles are concentrated in the one person. Because of this it is necessary to be able to ensure the integrity of any person in this role.

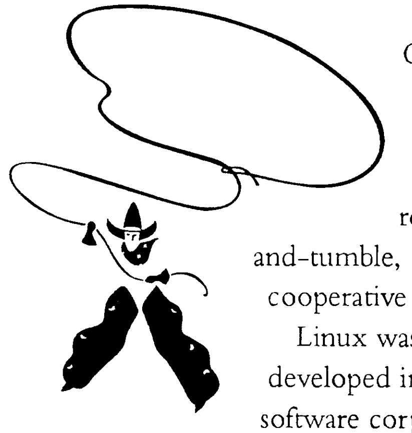
While it is possible to sight a certificate to prove knowledge, and even measure experience, it is very difficult to prove integrity. In an attempt to address this The Systems Administrators Guild of Australia (SAGE-AU) has produced a Code of Conduct, a voluntary code which outlines an ethical basis for how a system administrator should act. As well, SAGE-AU also has a Job Description Booklet (produced by USENIX/SAGE) which outline qualifications and job descriptions for various levels of UNIX System Administrators.

The combination of these documents provide an excellent start in defining what is required of an administrator for any organisation, from only a few PC's up to a multinational with sites around the world.

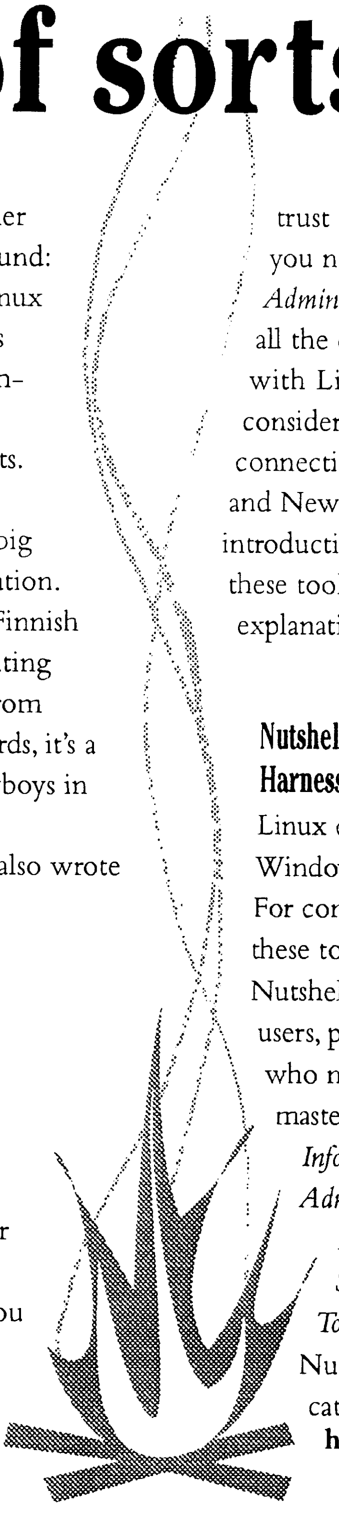
For more information on SAGE-AU you can contact the Secretary, Frank Crawford on (02) 717 3015 or email <secretary@sage-au.org.au>.❖

THERE'S A NEW OPERATING SYSTEM IN TOWN

A campfire story, of sorts.



Gather 'round: Linux has rough-and-tumble, cooperative roots. Linux wasn't developed in a big software corporation. Centered on a kernel maintained by Finnish student Linus Tovalds, the entire operating system was developed by volunteers from many different countries. In other words, it's a fine example of hackers at work. Cowboys in their own right, you might call them. Adventurous, free-spirited volunteers also wrote a lot of documentation (and O'Reilly pitched in, too).



Get the lay of the land with O'Reilly's field guides to Linux.

Running Linux by Matt Welsh and Lar Kaufman can make you a Linux power user. It discusses topics not addressed anywhere else, including everything you need to install and use Linux—even programming, Web servers, and X configuration are covered. Don't trust those fancy automatic X installations; you need our guide. *Linux Network Administrator's Guide* by Olaf Kirch describes all the essential networking software included with Linux, plus some hardware considerations. You'll learn about serial connections, UUCP, routing and DNS, mail and News, SLIP and PPP, NFS and NIS. Basic introductions to configuring and managing these tools are offered in easy-to-understand explanations.

Nutshell Handbooks:™ Harness the power of Linux

Linux comes with everything from the X Window System to TCP/IP to Perl. For concise and highly readable books on these topics and more, check out our popular Nutshell Handbooks. They're ideal for power users, programmers, and system administrators who need in-depth coverage of hard-to-master topics. Look for *Managing Internet Information Services*, *TCP/IP Network Administration*, and *Learning Perl.*

And don't miss *Volume 3: X Window System User's Guide* and *UNIX Power Tools*. For the low-down on all of our Nutshell Handbooks, browse our online catalogue at **http://www.ora.com/**

O'Reilly & Associates

Available From All Good Bookstores

Distributed in Australia by Woodslane Pty Ltd. ACN 003 677 549
Unit 7/5 Vuko Place, Warriewood NSW 2102 Telephone (02) 9970 5111 Fax (02) 9970 5002

Background:

Internet 3.0 Envisioned

Gordon Bell

Reprinted with permission from the 4/26/95 issue of UniNews, published by UniForum.

Symmetry and bandwidth will be needed in the future

"It's symmetry and bandwidth, stupid," said Gordon Bell, Internet pioneer and former Digital Equipment Corp. engineer, in explaining what he feels are the two greatest needs of the Internet in the coming years. Bell, a 1991 recipient of the National Medal of Technology, led the DEC design team that produced the VAX minicomputer and later headed the government team that developed the National Research and Education Network (NREN), an information highway precursor.

By symmetry, Bell means that the information flow on the Internet or any successor to it should run in all directions, not predominantly from a group of information providers to information consumers. By bandwidth, he means that the infrastructure has to be provided for transmitting hundreds or thousands of times more data volume than is now possible for most users.

Symmetry is needed, he said, because "Anyone should be able to be a producer of a consumer, and be able to send and receive equally." The more that becomes true for everyone, the better it will be for commerce, since participation will bring in more users, he said.

The need for bandwidth is obvious for transmission of high-quality audio and video-to a level of five million bits per second or higher to achieve good video transmission, Bell noted. Most home users now are limited to 14.4 kilobits per second. To achieve the goal, cable and telephone companies need to be pressured to provide high bandwidth, he said. "Get the phone guys to do it for us-they're getting paid for it anyway."

Bell said he's adamantly opposed to the cable TV model of one-way information for the National Information Infrastructure. "Cable TV is to the Internet as mainframes and SNA were to distributed processing. It turns users into couch potatoes and it's totally hierarchical."

Bell estimated that the Internet is now in 154 countries, consisting of 40,000 networks and 20 million to 40 million users. But the huge growth of Internet use has created a lot of hype that needs to be deflated, he added. With the world's population currently growing at 1.6 percent a year and Internet use doubling each year, Internet users would surpass the world population by 2003, he estimated. In some cases, Internet use hasn't even been productive. "The Internet is really Game Boy for adults, without the attendant improvement in hand-eye coordination," he said.

He defined the Internet's development as 1.0, 2.0, and 3.0 versions. He called his NREN project Internet 1.0. NREN was an extension of the National Science Foundation Network (NSFnet), the Internet backbone, to provide real-time high-speed connectivity between government, commercial, and research institutions. In addition, it provided a fiber-optic network used as a test bed for high-speed communication and computer technology.

Internet 2.0 is what we have now, Bell said. What he hopes for is Internet 3.0, which would feature a universal digital dial tone for speedy transmission and 25 megabits per second of bandwidth at the home level. He said ISDN will be inadequate. "There are cable guys who can give us the bandwidth that we need and they'll get there a lot faster than the phone companies."❖

Background:

The Desktop of the Future

Frank Crawford

During the 1980s it was common for serious computer users to have two or even three terminals on their desk. When the PC became standard equipment this dropped back to one device, however, in recent times more devices are again appearing on the desk. While in the typical office environment a single PC will satisfy most requirement, in many other environments, such as an R&D group, there are likely to be PCs, X-terminals and even Macintoshes, all found side by side.

The reasons for this are obvious, each platform has its own strengths and weaknesses, and a knowledgeable user will take advantages of these strengths. Unfortunately there is only so much room on a desk, and so a rationalisation of the available equipment is needed. To most people this would mean being constrained to an "inferior" system. In reality, like the combining of multiple terminals onto a single PC, it offers many advantages to the user.

Probably the most common systems to be merged are MS-DOS/MS-Windows and UNIX, for which there are a number of possible schemes. By far the simplest and least satisfactory is a dual boot PC, i.e. selecting at boot time between MS-DOS and UNIX. This provides access to MS-DOS and MS-Windows at one time, and UNIX and X-windows at others, but no sharing between the two.

A better setup is to run a package such as DOS-Merge or SoftPC, which gives a virtual 8086 machine on a UNIX system. In its simplest form it is run on a PC-based UNIX system meaning that there is no difference in the code, only in the handling of MS-DOS calls. Most such packages have been modified to run MS-Windows 3.1, general in enhanced mode. The performance is comparable with a PC running only MS-DOS, but generally a little slower, and with the advantage that you get access to UNIX's virtual memory to provide as much memory as your MS-DOS program requires. For people who mainly want to run UNIX applications under a PC architecture, this is a very good solution, as it allows full access to all UNIX functionality and to the most common MS-Windows programs through a single device.

This has even been taken one step further in UnixWare, Novell's version of UNIX SVR4.2, which has had the Novell networking protocols built into the UNIX kernel. This allows access to Novell file and printing services, directly from the UNIX command line, and from most applications, without any additional knowledge by the user.

Unfortunately, this approach is not perfect, there are limitations as to what MS-DOS and MS-Windows programs you can run, for example no win-32 applications are supported. Further, whenever Microsoft make changes to MS-Windows or write an application that uses an undocumented feature, there is a good chance that it will not run under UNIX until the package has been modified.

If you are concerned about the cost of such a package, it is probably cheaper than a second PC for everyone, and if that isn't convincing, then there are the publicly available packages "dosemu" and "wine" being developed for Linux. At present they are still unstable, but as with most Linux products, they are being improved almost daily.

One perceived limitation is that these products are limited only to PC-based UNIX systems is not true. SoftPC has been ported to a number of other architectures, including Sun's Sparc and Silicon Graphics systems. This is done by running a 8086 emulator on these other architectures, allowing any 8086 program to run. This obviously requires a higher performance system, but such workstations generally do have this higher performance.

Emulation of MS-DOS and 8086 code is one approach, however, SunSoft have developed a different approach. Instead of worrying about MS-DOS, they have implemented an interface between MS-Windows and X-windows, call WABI (for Windows Application Binary Interface). This takes the MS-Windows' calls and generates equivalent X-windows' calls, to display directly to an X-terminal. There is no support for running MS-DOS programs, and in fact, running MS-DOS is seen as an entirely separate issue.

With these techniques, a heavy UNIX user, who occasionally needs to use MS-DOS or MS-Windows, can be satisfied, with only one device on his desk. In fact, the techniques given here have also been applied to the Macintosh, with an emulator available for Sun Sparc systems, which can run MacOS applications, and conversely SoftPC has been ported to Macintoshes to allow MS-DOS/MS-Windows applications to be run on them. Both of these work through appropriate emulation.

So what of the poor MS-DOS user? Generally a different approach is taken for them. Rather than trying to run UNIX under MS-DOS (an impossible task), products are available to allow them to either run TCP/IP network services (e.g. Telnet and FTP) or to display X-windows applications on their PCs. Most of these run under MS-Windows and so are normal applications, their only special requirement is that TCP/IP be supported by the PC. For users who primarily access PC applications, and only need occasional access to UNIX and X-windows based applications, this works well. For users who make extensive use of UNIX, it is not as good, as they often run into either memory or graphics resources problems on their PC.

Finally, for a different concept altogether, operating systems such as OS/2 offer the ability to run all these systems in their own separate partitions, and thus to make use of the best features of each. This is technically much more complicated, and likely to suffer from more restrictions, but may also suit the needs of a special group much better than either of the earlier proposals.

So, what of the future? It is obvious that the current emulation techniques will not work indefinitely, as Microsoft make more use of the power of the Intel processor, it is difficult for the developers of DOS-Merge and SoftPC to keep up. It is likely that these packages can be extended to handle Windows95, but they will not work with the likes of Windows NT. On the other hand, new approaches are being adopted, such as on the PowerPC, which has hardware support for Intel emulation, or other systems which provide an additional Intel processor, specifically to run MS-DOS product. All this supports the idea that even in the future, one device will be all you need, it will just be more difficult selecting the right one for you.❖

Background:
Information services

Jamie Honan

Lose weight quickly, no dieting, no change of lifestyle. You'd know I was lying if I told you that.

What about a corporate information service, across Mac's, PC's and UNIX workstations, with no change in work practices, expensive software or steep learning curve?

For companies running the TCP/IP network protocol, this 'holy grail' is very achievable.

Two researchers, Ken Yap and Bill Simpson-Young from CSIRO's division of Information Technology, gave me a demonstration of just such a system. Bill was somewhat embarrassed at first, almost apologetic that something so simple and 'un-hitech' should prove so successful in a research organisation like the CSIRO.

The information service provides such facilities as internal library searches, bulletins, repositories of internal memos and documents. It has a simple ordering and approval system, inter library loan facilities, all integrated with the email system.

The approach used is borrowed from Internet World Wide Web tools. Although links to the Internet are used, the real success story of this system is how it is used so effectively in a local way by staff who have received no special training.

The approach used is so simple, the tools used freely available from Internet sources, that any organisation with a UNIX server and administrator / programmer could set up their own similar system.

The front end used for browsing the information service is Mosaic or Netscape for all platforms - Mac's, PC's and UNIX workstations. Much has been written about these freely available packages in other columns here, mostly in reference to the Internet. The important thing to note about these tools is that they have a completely different feel when used over high speed networks such as Ethernet. Graphic images, which normally load very slowly over congested links or across modem lines, flash up very quickly.

To place information into the system, staff write their memos using Word or similar word processing tools on their PC's or Mac's. They then save the file as RTF - rich text format - the *lingua franca* amongst word processors, and mail the file as an attachment to the information server.

The server, not a person but a machine, converts the file into the HTML format suitable for use by WWW viewers such as Mosaic. All this is done automatically, the RTF conversion done by freely available tools, the automatic mail processing by standard UNIX mail handling tools. The new file is then entered into the information service, with a new hypertext link automatically generated.

Bill Simpson-Young wanted to brush past the opening hypertext page, with it's seemingly mundane references to 'How to use the projector in the conference room', 'Phone codes' and 'Cake days'. The layout was attractive, the administrative staff responsible for the memos had their photos above the hypertext reference items. I saw the link to the page which I wish was on every site I have ever visited, 'How to use the new photocopier'! This is the sort of memo that gets lost, forgotten about until the day you really need it.

The local library search, inter library loans and book ordering systems utilise the forms mode of the WWW system.

In this mode, users fill in fields in a form, click various buttons representing choices. When completed, the form is automatically returned for servicing.

Ken Yap devised some simple Perl scripts to do the processing. In the case of the library search simple search techniques are used, but the principles could be used to support sophisticated database systems. There are a number of Perl extensions that interface directly to database systems such as Oracle, Informix and Sybase.

The code to generate and process the book ordering system consists of 156 lines of Perl, half of which are comments. After a user has filled in the order form, the Perl script e-mails the item to an approval clerk.

Every few weeks the in house magazine, 'ondit', is e-mailed to the information service in RTF format. It is automatically processed and placed in the repository complete with attractive graphics and banners. Everyone receives email notifying them of its availability.

It is easy to imagine extensions to such a system to cover a multitude of organisational activities. While a traditional database system, using proprietary tools such as PowerBuilder or Access might seem the first choice, there are several advantages of using the WWW hypertext system.

The viewer is quite economical in using disk and memory, so it doesn't 'take over' the user's system. The layout and tools for generating memos is simply whatever format the user has used in the past. To enter memos into the system, the user simply e-mails the memo, an action most people are now familiar with. The hypertext nature of the WWW technology and of viewers such as Mosaic lead untrained users in a natural 'point and click' way to the information they are seeking; the learning curve is gradual.

All this is possible because of two important technological factors - the TCP/IP network protocol provides the network glue to bind disparate systems such as Mac's, PC's and UNIX workstations and servers, and the UNIX information service host has such flexible processing options such as Perl, the WWW server and automatic mail processing.

Sometimes the simplest things are the best, and sometimes the best things are free. Sometimes, rarely as in this instance, the best things are simple and free!

A 'fact sheet' listing the software components used in dit's information service system, and from where they can be retrieved using ftp is available from Jamie Honan. Just send email to **jhonan@jolt.mpx.com.au** asking for the information service references.❖

Opinion:
Long Live Text

Jamie Honan

Kelly Stan Bootle said it best in the magazine Computer Language: "The frayed dictum that a picture is worth a thousand words has caused much confusion in our industry. Note first that the ideas expressed in the 18 words of the previous sentence cannot themselves be crisply expounded using any purely pictorial, nonverbal drawing I can think of. Send me a sketch if you disagree."

One measure of the progress of our civilisation is the fact that we have council libraries filled with books. The act of reading is a discipline, a demand we make of ourselves. The rewards from reading are the fantastic insights, the charged emotions, the transfer of incredible ideas.

Sadly, the height of the computer revolution has produced popular software that reduces our ability to interact with computers to below that of 14 year olds. I'm not referring to games, the best of which are truly demanding and absorbing, but to the plethora of 'Windows based' software, most of which an advanced user actually has to battle against to get much done.

As an example, consider the word-processor. In its modern incantation, this ubiquitous tool allows you to select fonts, layouts, special effects and any one of a million special features. Of course, the beast does nothing to help you accomplish the primary task: actually write sensible, meaningful, sentences. This task could be accomplished via a simple text editor, or, dare I say it, a pencil and the back of a nearby envelope. I have often wondered how many spreadsheet calculations could be reduced to the same technique.

Could it be that our obsession with the tool, the computer, has blinded us to the question as to whether our productivity has actually been enhanced?

Although it may seem old fashioned to be proud of a character based terminal heritage, many UNIX applications can run on a variety of character oriented displays. Anyone who has used a computer in any serious way will always appreciate a command line interface, or the fact that much of what we deal with in day to day work can only be done with words.

I have seen many attempts to fit data entry and database applications into Windows or Windows-like environments. This has always struck me as absurd, there is not a jot of productivity to be gained in changing from a character mode system to a GUI system for such applications. Instead the applications run slower, or require more expensive hardware. The GUI interface demands more memory and disk space, costlier displays and faster CPUs.

The real shining light of UNIX is its ability to manipulate text, text processing as it's called. There are a host of tools for searching, replacing, counting and otherwise munging text. The humble 'grep' searches for that elusive word, phrase or acronym, across hundreds of files in minutes. I have many hundreds of text files on my system, all crammed with titbits of information that I am continually going back to reference. When confronted with a Microsoft developers' CD-ROM, with it's Windows' based search and retrieval engine, I pine for my humble tools. I have no way of knowing whether the information I require is present, the tools aren't there to help me find it.

No diatribe against the modern trend away from words and text would be complete without mentioning the 'information superhighway'. In the mass media, this concept is presented not as a way to access the fantastic textual resources of the world's information repositories, but as a chance for various spiv entrepreneurs to distribute interactive TV, with home shopping the most creative service they can think of.

Yet it will be some time before the information superhighway can deliver the information bandwidth and convenience of your local library, and some time before the spivs have gone broke and cleared off and the gates of the new global age have been opened to Joe and Joanna Ordinary.

The only existing, working (as opposed to gleam in the eye of a marketing type) global information superhighway is the Internet.

Absolutely ignored by the mass media, but critical to the success of the Internet are the fundamental protocols used - TCP/IP. TCP/IP has so many advantages over other protocols touted for this purpose, its chief one being that it actually works for the scale required by such a massive network.

As an indication of our indulgent use of computers and information technology, the truly productive services provided by TCP/IP are ignored. Glossy Web pages have won out over the truly 'user-enabling' text based tools such as email and news. To many now, the Internet is the Web page, often a thinly disguised advertisement, on their Windows' PC.

Contrast this with email. Email represents the renaissance of the humble written letter. The composer of a well written letter is educated and civilised. The writer has every opportunity to show erudition, wit, charm and flirtatiousness. The writer acts, is not passive, transmits something of him or her self.

What can listening to rock music on the Internet, or the viewing of Web pages containing pictures of 'Randy's lunch', or interactive shopping on TV offer us but more mindless indulgence, a deadeningly passive third hand set of experiences?❖

Background:
Firewalls

Frank Crawford

Despite its cost effectiveness and many benefits, there are risks in connecting to the Internet. Security threats range from curious prowlers to well-organised, technically knowledgeable intruders who could gain access to your site's private information, or interfere with your ability to use your own systems.

While it is a good idea to make your workstations, servers, and other systems as individually secure as possible, this is not sufficient to defend your site against attack. Without the ability to protect an entire network at its connection point, your defence is only as strong as its weakest link, and securing each and every system is a complex and cumbersome job with no guarantee of success.

However, by analysing and defending against threats at your site's point of connection to the Internet (or a parent organisation's WAN) you can take advantage of most Internet services, such as electronic mail, anonymous FTP, and Gopher, while at the same time limiting your risk of intrusions.

What is a firewall?

An Internet firewall is a security mechanism that allows limited access to your site from the Internet. It is designed to limit access through such mechanisms as packet filtering, proxy services and dual-homed gateways.

However, as every site is individual, there is the need for the configuration of their firewall to reflect this. While it is possible to buy a firewall off the shelf, or hire a consultant to configure one for your site, invariably there is still the need for local knowledge. Alternatively, there are freely available toolkits for constructing your own firewall, but this obviously requires a greater degree of knowledge. Further, knowledge of the concepts behind a firewall is necessary not just during the initial configuration, but on ongoing basis as your firewall needs to be updated to track changes at site.

For example, consider the dramatic growth in the World Wide Web, any site that hasn't modified their firewall in the last two years wouldn't be able to participate. Alternatively, consider the effect on an organisation that has changed from a small number of UNIX hosts to everyone having a PC on their desk. This would require someone to understand both how data passes through the firewall, and how to configure the organisations PC to use the services available.

Even worse, despite their best efforts, new technologies, new approaches and new techniques may well develop methods to bypass the safeguards in current firewalls. As an example, recently there have been reports of break-ins through firewalls, using a technique that was described nearly ten years ago, but until now not practical. Using these techniques crackers would have been able to penetrate most firewalls available at the time. Fixes have how been developed, but unless you have an understanding of the principles behind your firewall, it would be impossible to implement.

Unfortunately, while the knowledge about firewalls is not lacking in Australia, the rapid growth of the Internet has meant that this expertise has been stretched thin.❖

Opinion: Economics

Jamie Honan

"It's the awful international mediocrity that they are forcing onto us", Red Kev, my father-in-law, was rambling on in his usual manner.

Our government, in league with McDonald's and the Coca Cola company, is somehow forcing us to live in international uniformity. Mostly this seems to have to do with international finance, the sale of Telecom and the demise of tariff barriers, but I usually glaze over and sip his excellent wine before we get too deep.

"The great God, economic rationalism", Kev is usually mumbling at this point.

It's undeniable that we are giving up national sovereignty, little by little. Economists would probably call it "surrendering to the discipline of the market", and yet I wonder.

Kev would be shocked by the comparison, but the militia groups in the US have a similar complaint. The US government has ceded decision making to international bodies. The extent, of course, is arguable.

And yet what is the alternative? If we do not go down the route of international decision making, how are we to resolve problems such as pollution, terrorism, controls on armaments (nuclear and conventional). If we are not to isolate ourselves completely, it seems to me that international co-operation is inevitable to some extent.

As an example of a country attempting isolation, take Burma. During the seventies, medical treatment included black market payments for medicine. To properly implement the "Burmese road to socialism", western medicines could not be imported. Somehow Kev's polemics would evaporate if this was the future for his grandchildren.

The problem is not international law, but the manner in which is it imposed. Instead of international democracy we have international bureaucracy.

This is hardly surprising, as up until now it has been technically impossible to conduct international election processes, provide free flow of information to individuals so that they are informed, and all the other trappings that democracy requires.

At this juncture of our history, individuals now have the ability to communicate and interact in a trans-national way: not in Cyberspace, but in communities gathered in a way unaffected by distance.

As Rupert Murdoch said it "Right now, anyone, anywhere in the world, can walk up to a computer and communicate with anyone else in the world". Murdoch was exaggerating, of course. How will the Bengali villager walk up to a computer? But it might be an approximation, a start, as to how we might organise ourselves against international bullies, rip-off merchants and other undesirables. How else can you describe, for example, companies that export their high pollution factories to the third world, or various tin-pot army thugs terrorising civilians.

The current situation, where the vanguard is being spearheaded by international bureaucracies such as the UN, the International Atomic Energy Commission, the World Trade Organisation, is necessary but not sufficient. These groups will inevitably lack teeth, hampered by national governments fearful of losing control, fearful of a local backlash. In addition, being bureaucracies, they function best as advisors, implementors, under the direction of political leaders.

The political and social forces that the facilities that the Internet provide will take time to work their way

continued page 29

UNIX Tricks & Traps

Edited by Janet Jackson
<janet@dialix.oz.au>
Phone/Fax (09) 272 5061

Welcome to *UNIX Tricks & Traps,* aka the *Code Column*. Short programs are, of course, welcome here, but I'd also like to see some textual how-to articles. These don't have to present original ideas. Remember, there are probably many readers who have less UNIX experience than you. So how about sharing some of your knowledge?

I'd like to hear readers' suggestions for topics too, so I can solicit articles on them.

This time we have two different **iota** counting programs, written in **Perl**, to be used from the shell or wherever you need them. Greg Rose calls his program **count**. This is probably a more sensible name than **iota**, but it is his article that explains why these things seem to be traditionally called **iota**.

By the way, if you don't understand about Greg's ": ;" prompt, see this column in the April 1995 issue of *AUUGN*.❖

Greg's Perl Iota

Greg Rose <greg_rose@sydney.sterling.com>

Sometimes you need to generate a lot of lines, but a particular number of them. Other times you might need a list of unique somethings. I wrote a program one day when both of these situations happened to me within hours of each other. The thing that both of these have in common is that you don't particularly care what the list has in it.

Sometimes you actually want a list of numbers. The APL **iota** function, or equivalent. Once **perl** came along, this became particularly easy, but the original version of this program was much less functional, and predated **perl**.

The first use I had was to simply execute the same command ten times. I got used to typing:

```
: ; for i in x x x x x x x x x x; do <command>; done
```

... except of course I always either miscount or mistype the line. Now I say:

```
: ; for i in `count 10`; do <command>; done
```

... and I'm less likely to make a mistake.

The other thing I needed it for that day was creating a bunch of special files in the **/dev** directory. I know I'm showing my age here. But I could type:

```
# for i in `count 0 63`; do /etc/mknod /dev/special$i c 7 $i; done
```

and be reasonably sure it did it right. I once wiped out the contents of a disk because of a typo in the middle of a sequence of individually typed commands to make the devices one at a time.

Anyway, here is the program. It is much more complicated than it needs to be for the examples I've given, but it has nice defaults.

```
#!/usr/bin/perl # generate a list of numbers

if (@ARGV > 3 || grep(/[^-0-9]/, @ARGV)) {
   die "usage: $0 [[low] high [increment]]\n"
}

if (@ARGV == 0) {
   $low = 1;
   $high = 10;
   $incr = 1;
} elsif (@ARGV == 1) {
   $low = 1;
   $high = $ARGV[0];
   $incr = 1;
} elsif (@ARGV == 2) {
   $low = $ARGV[0];
   $high = $ARGV[1];
   $incr = 1;
} elsif (@ARGV == 3) {
   $low = $ARGV[0];
   $high = $ARGV[1];
   $incr = $ARGV[2];
} else {die;}

# generate
if ($incr > 0) {
   for ($i = $low; $i <= $high; $i += $incr) {
      print $i, "\n";
   }
} if ($incr < 0) {
   for ($i = $low; $i >= $high; $i += $incr) {
      print $i, "\n";
   }
}
```

❖

WAUG news:

From the Western Front

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

What's new in Perth? Well, thanks to Canberra AUUG, WAUG (the WA chapter of AUUG) now has a World Wide Web page at **http://www.auug.org.au/auug/waug/waug.html**.

At our June meeting Stewart Snell, Engineering Manager at JTEC, gave an interesting talk on ATM (Asynchronous Transfer Mode), a new low–level (as far as I can see, it's at the data link layer in the OSI reference model) networking technology that looks like being most useful for internetworking a medium-sized area like a campus, plant or town.

ATM is not new in the sense of having just been invented, but it is new in the sense that it's only just beginning to reach the stage where end-user organisations can get some sort of hardware. JTEC, which bought out QPSX, looks like being one of the companies you might buy the hardware from. However, Stewart's talk was not a sales pitch, more a description of ATM and how it can fit in to an organisation's network.

I thought we could have used more technical detail in some parts, but on the whole, the talk was pitched at the right level.

The new committee met in June; Adrian describes most of the results of this in his column. We have appointed a meeting reporter (or, if you like, meeting report coordinator), so you can expect the standard of meeting summaries and reviews to improve drastically.

This is probably my last column before AUUG'95, so I'll see some of you there.

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.

UNIX Tricks & Traps (continued)

Janet's Perl Iota

Janet Jackson <janet@dialix.oz.au>

In accordance with the **Perl** motto "there's more than one way to do it", here's my version.

It is much less user-friendly than Greg's program; its increment is always 1 and it doesn't have any defaults. However, because I've used **Perl**'s range ("..") operator, it has the extra feature that I can say "iota a z" or "iota a0 b9" and get what I (at least) expect. Not that I ever seem to use that feature.

Be careful though — **Perl**'s alphanumeric string increments don't always do what you might intuitively think.

```
#!/usr/bin/perl

# iota # Janet Jackson <janet@dialix.oz.au> 1994-10-17

# Example of action:
# Given
# iota 5 8
# outputs
# 5
# 6
# 7
# 8

@ARGV == 2 || die( "Usage: $0 botnum topnum\n" );

foreach $n ( $ARGV[0]..$ARGV[1] ) {
   print( "$n\n" );
}
exit 0;
```

From the Chair

Adrian Booth <booth_adrian@tandem.com>

I've been somewhat lax in my chairing duties as I've been interstate during the last two WAUG meetings. The first of these meetings was the AGM, at which I was re-elected as chair. There have been several changes in the committee which I'm sure you will be able to read about elsewhere in this AUUGN *[WAUG election results were in the June issue -Ed.]*.

The Brent Chapman visit to WA attracted 28 attendees. The WA leg was organised by David Buck of Industrial Micro Products, who did a fine job. As a result of his great work the WAUG committee voted to offer David the newly-created position of Publicity Officer for WAUG. Unfortunately due to work commitments David was unsure whether he would be able to give the position the effort it would require.

So we are now looking for a WAUG member who would be interested in becoming our publicity officer. This would chiefly involve making sure that details of our meetings were publicised early to the media, as well as handling the publicity for the summer conference. If you might be interested and would like to discuss this further please contact me.

The appointment of a publicity officer goes hand in hand with the appointment of a committee to organise the 1996 summer conference. Janet Jackson has agreed to be the program chair. Other committee members and their duties will be announced once roles have been finalised. I will be mainly taking an old fart role, giving guidance to the committee. It's time to start thinking about what *you* are going to talk about — last year I had to take the approach of simply asking several people to give a talk, meaning that a few people missed out on the opportunity. Don't let that happen to you this year!

Another informal position that was created at the last WAUG committee meeting was that of Meeting Reporter. Getting regular meeting reports has been a little like pulling teeth, and so James Patton from Main Roads will be browbeating various people to write a report on a meeting. It doesn't take very long to write one, so please think about it if James approaches you.

You will have noticed that my email address has changed. By the time you read this I will be working for Tandem Computers in a (still technical — really!) pre-sales position. After beating my head against coalface-level UNIX workstation administration for several years, I'm looking forward to a higher-level and broader role. It'll be interesting to see what things are actually like on the other side of the fence.

❧

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**:

waug-membership for membership enquiries

waug-chair our Chairperson

waug-meetings our meeting organiser

waug-secretary our Secretary

waug-newsletter for newsletter contributions or 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**.

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** and **aus.org.auug** for announcements and discussion.

❧

Committee Contact Details

Office-bearers:

Chair: Adrian Booth 354 4936 booth_adrian@tandem.com Treasurer: Patrick Ko 483 8111 pko@DIALix.oz.au Secretary: Major 357 5076 major@yarrow.wt.com.au

Ordinary committee members:

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

A publicity officer has not yet been appointed. Please volunteer.

❧

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**>, (09) 272 5061, or the meeting organiser, Mike Horton <**mikandfi@DIALix.oz.au**>, (09) 479 8424.

For information about SAGE-AU in general, you may also look at **ftp://ftp.mel.dit.csiro.au/pub/SAGE-AU** and **http://sage-au.dstc.edu.au:8080/**. ❖

QAUUG news:

Queensland Chapter update

Mark White <m.white@brisbane.tandem.com>

While not reported in these pages of late, the Queensland Chapter of AUUG continues to prosper. We have held a successful "summer" conference (in late April), as well as well-attended Chapter meetings in May, June and July.

❧

Conferences

Firstly, the summer conference attracted a record number of attendees, and boosted the Chapter coffers significantly, but most importantly was considered to be of an exceptionally high standard while maintaining excellent value. Special thanks are extended to our interstate and local speakers, as well as to all those who attended. The success of this conference argues well for the continued growth and confidence of AUUG in Queensland throughout the next twelve months. This growth is reflected in our recent and forthcoming meeting schedule - we'll continue to have local speakers, but will also attract a growing number of interstate speakers to our sunny climate.

❧

Meeting reports

Andrew Tridgell, of SAMBA fame, fortune, and pizza vouchers, was the guest speaker for our May meeting. Andrew's talk of course featured a review of SAMBA's development and feature lists, and also some amusing anecdotes from the development cycle. What, some Australian guy finding holes in Microsoft's secure networks????

June's monthly meeting featured QAUUG stalwart and popular speaker David Hughes, Technical Manager of Web Publishing Australia. David's talk documented the history and phenomenal growth of the World Wide Web. Active discussions about security and potential uses of the 'Web ensued.

The June meeting also acted as the chapter's AGM. A new committee, consisting of the familiar faces of the old committee, was elected unopposed. (The committee's contact details are listed below). Tim Butterfield later stepped down as Chairman of the Queensland Chapter, a position he has held almost since the first meetings of the Queensland UNIX User's Group - an informal gathering that later became affiliated with the national body. Our thanks are due to Tim for his contributions to AUUG over the last four years.

Also held in June was Brent Chapman's tutorial. Over 50 attendees agreed it was a valuable experience, and again a successful venture for the Queensland chapter.

The July meeting featured Greg Rose of Sterling Software. Greg's talk about PGP and the Web of Trust was well received by all.

Our next meeting, on August 29, features Ken McDonell of Silicon Graphics. We endeavour to send out meeting notifications via e-mail a week or two before each meeting, and also a fax-out for those who may not be on e-mail. If you don't already receive one of these notifications and would like to, please contact one of the committee members. The monthly meetings of QAUUG provide a forum for engineers, managers, dealers and end users to exchange ideas and experiences. If you have an interest in open systems, come along and meet like-minded people. Our meetings are always held on the last Tuesday of every month, at The Regatta Hotel, Toowong

QAUUG committee

Michi Henning **michi@dstc.edu.au**
Rick Stevenson **rick@stallion.oz.au**
Mark White (Chair) **m.white@brisbane.tandem.com**
Tim Butterfield
Paul Reithmuller **par@aus.sun.com**
Stuart Remphrey **Stuart.Remphrey@aus.sun.com**
Duncan Unwin **duncan@acslink.net.au**
Steven Hocking **sysseh@devetir.qld.gov.au**

They may be collectively e-mailed at **qauug-exec@bond.edu.au**. A World Wide Web home page is under construction, and may be viewed at **http://AusWeb.com.au/computer/qauug/**

Finally, looking through the last issue of AUUGN has caused me to move a vote of thanks to Phil McCrea for his efforts as AUUG president for the last few years. Phil was instrumental in the establishment and affiliation of the various chapter groups, amongst his many other achievements. Congratulations Phil, on a job well done!❖

from page 24

through to fabric of the daily lives of non computer-literate citizens. Yet, simply for technical reasons, these facilities are perhaps the only way we can address global issues.

Similarly, it represents the best hope for people whose plight is being ignored or exacerbated on a local level, but who should be able to call on the consciences of the rest of the world, knowing, grateful, they can do something about it.

Now is the time for great experiments with this tool, the Internet. Now is not the time to hand over fledgling government policies in this area to a bunch of cowboys who want to extract ten cents for every message. We, the people, should control this tool, take it where we want. To be become true citizens of the world, we may have to overcome obstacles put up by our own government, hand in glove with an entrepreneur with no insight into the world beyond what his own ego presents.

The choice is ours.❖

Readers respond:

More tricks; more traps

David Purdue
SunSoft Senior Support Engineer
<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, where as **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 prticular 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 tool-kit.❖

AUUG BOOK CLUB
&
PRENTICE HALL AUSTRALIA

20% DISCOUNT TO AUUG MEMBERS

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

□ ***SCO Open Desktop***
Open Server User's Guide 3.0
ISBN: 0131068164 Paper $59.95

□ ***Keys to Successful UNIX System Management***
ISBN: 0139483810 Paper $54.95

*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 FAST 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) 939 1333 Fax: (02) 905 7934
A VIACOM COMPANY

Book Reviews

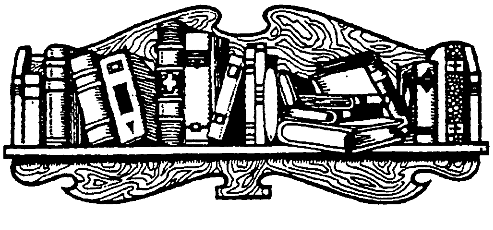
Frank Crawford <frank@ansto.gov.au>
Review editor

Well, we roll into another book review section and this issue, it is packed, with new books from O'Reilly and Associates, one by a whole new publisher (WoodsLane, ORA's Australian distributor) and a major new technical book by Richard Stevens.

As well as the books, there are a number of peripheral items going on over the next month. First, we have **Tim O'Reilly**, the founder of *O'Reilly and Associates* as an invited speaker at AUUG95 in September. I believe he is a very interesting speaker and expect you all along to hear him. : -)

The second item is creations of a new mailing list for announcements related to the AUUGN Book Reviews. You can subscribe to this by emailing to **<auug-books-request@ansto.gov.au>** containing in the body the single word **subscribe**. (There are obviously more commands possible than this, but I'll leave that to the interested reader to discover.)

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>** and the newsgroup flaus.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.❖



The Australian Internet Book

by Geoff Ebbs and Jeremy Horey
WoodsLane Press 1995
295pp. + diskette, $49.94
ISBN 1-875889-00-0
Reviewed by Phil McCrea
CSIRO Division of Information Technology
<philip.mccrea@syd.dit.csiro.au>

This is the first in what will inevitably be a procession of books on the Internet in Australia. It came out in March this year, and has been in the top 10 best selling list for quite some time now. It is most lucidly written, which is only to be expected from two authors who are journalists. Most of you would recognise Jeremy Horey as being the Editor of the Tuesday's *Australian* supplement. Geoff Ebbs is also well known as a freelance journalist, and has a column in the same newspaper entitled "Sermon on the Mouse"!

I really enjoyed reading the book. Although let me be honest, I scanned it more than read it. The last Internet book I had reason to pour through with any degree of scrutiny was Ed Krol's seminal work, "The Whole Internet". This was published a year earlier than Ebbs and Horey and includes a chapter on the Web, which was very obviously an update to the 1992 edition. As testimony to how important the Web has become to the Internet, Ebbs and Horey start off with the Web, after the mandatory series of chapters on what the Internet is. While Krol's book is pitched more at the more technical Internet user, Ebbs and Horey is aimed at the average PC user who has just come across the Internet.

The first service the book discusses is the Web, with lots of screen dumps of interesting sites, complete with their URLs. Then follows chapters on gopher, FTP (described as "The Worker"), and telnet. Then follows a chapter on Searching, where archie, Jughead and Veronica get a guernsey. All of these topics form a meta topic area called "Exploring".

The next meta topic area is called 'Communications', and looks at email, news, Internet Relay Chat, audio services, and then finding people of the Internet.

The final meta topic area is the Reference section, which lists Australian Internet Service Providers - at least those who were around in December 94. A few months is a long time in this game, when you realise that AusNet is not one of those listed. Next follows a list of Australian WWW sites, which is probably only a tiny fraction of those on the net today. Gopher and FTP sites follow. There is a useful glossary of terms, and finally a disc providing access to Enternet.

All in all, a most useful book, and one that all casual Internet users (they insist on the dreadful term *Internauts* ...) should have on their reference shelves. It is, however, the kind of book that dates very quickly, but I hear on the grapevine that a second edition is in the wind.❖

TCP/IP Illustrated, Volume 2: The Implementation

By Gary R. Wright and W. Richard Stevens
Addison-Wesley Publishing Company 1995
1174pp., $54.95, Hardback
ISBN 0-201-63354-X

Reviewed by Michael Haldey Genasys II Pty Ltd
*<michaelh@genasys.com.au>

After I had read Volume 1 of the TCP/IP Illustrated (The Protocols; see review in the AUUGN Volume 15 Number 4, pp 43- 44), I was expecting the second Volume with great interest. The book has completely fulfilled my expectations and it has just set yet another quality benchmark in the technical literature devoted to the UNIX and networking.

The Volume is a detailed comment on the BSD implementation of the TCP/IP stack (4.4BSD-Lite version). Unfortunately this is not clear from the title. "The BSD implementation" or even "The UNIX Implementation" could sound a lot better. The authors present and comment on the whole kernel part of the TCP/IP stack (more than 15000 lines of C code). This is definitely a challenge itself and they have managed to do it in a very clear, precise and balanced manner.

The book is aimed at the advanced reader who is able to read and comprehend this large and complex code. Basic working knowledge of TCP/IP and networking in general is a must as well as C language proficiency and basic understanding of the operating systems design (the code in the book is a part of the BSD kernel).

The authors present the code using the bottom-up approach, so that when you read a description of the code you basically know everything the code is based on. They start from the introduction and the discussion about mbufs (the memory structures widely used by the BSD kernel). After that they present the Interface Layer, discussing the code of the ethernet, SLIP and loopback drivers. Then, after the discussion and presentation of the code dealing with IP addressing information and data structures to support concurrent operation of the multiple protocols, the IP layer is presented, including ICMP, multicasting, IGMP and multicast routing. After that, an implementation of the sockets system calls is discussed (frankly, I expected to see it at the end of the book, as the highest layer), then the routing tables and algorithms implementation, ARP, Protocol Control Blocks, UDP and TCP. The last chapters present the code for BSD Packet Filter and Raw Sockets.

The description of the code is very thorough - the files and their locations in the source code are always specified as well as the data structures and the global variables used. There are a lot of diagrams, tables and schemes that make reading much easier. The comments themselves are very informative, with the necessary references, bug notifications and the history notes.

Even if the original idea was to produce a companion book for Volume 1 (The Protocols), Volume 2 is largely independent of any external reference. The authors put information inserts at the beginning of each chapter before the code is introduced. The inserts are well balanced, clear and informative. They are basically sufficient to understand the code and the comments. Only in rare cases the reader is referred to Volume 1. But, again, this independence means only that Volume 1 doesn't have to be open. Without general understanding of how TCP/IP operates it is useless to read the book.

The book can be used as an advanced self-study tutorial - numerous exercises are provided at the end of each chapter (a lot of them with answers in Appendix A). I didn't have any problem to using it as a reference either - there is a large general index, an index of the functions and an index of the data structures so that the necessary information is quite easy to find. The text has numerous cross references so it can be read from almost any point (Though I would suggest reading the Introduction and mbuf chapters beforehand).

The book is a logical continuation of Volume 1 for those who want to know how the protocol parts are fitted together with each other and the operating system. It contains a lot of details which are omitted in Volume 1. For instance, the description of the IP multicasting and IGMP protocol takes only 10 pages in Volume 1, but it is about 100 pages in Volume 2. The logical way of the learning about TCP/IP seems to be to start from Volume 1 and jump to the Volume 2 when additional details are required.

It is a pity that the book doesn't present any code of the SNMP agent. The authors are hardly to be blamed because the code is not actually a part of the 4.4BSD kernel implementation. The kernel structures used by SNMP agents are described thoroughly throughout the book.

Overall I rate the book as excellent and would strongly recommend buying it to anyone who wants to know the details of the TCP/IP stack internals in the UNIX environment. For the those who are using non-UNIX implementations of the TCP/IP (say, PC programmers using Winsock interface to the numerous MS Windows based TCP/IP kernels) the book may still be of some value - these implementations, though rather different in the interaction with the operating system, are likely to have a lot of code directly inherited if not copied from the BSD implementation.

At the end let me compare briefly the book with Comer's TCP/IP Volume 2 (The Internals) - the only book about TCP/IP implementation that was published before. Comer's book presents Xinu implementation of TCP/IP; the implementation seems to be close to the BSD one. For a professional who works with or wants to know TCP/IP implementation details, Wright and Stevens~ book would be more useful and with all respect it supersedes Comer's. But for a student who is trying to understand the basics of protocol implementation using TCP/IP as an example (and doesn't need a lot of details), Comer's book can still be recommended as a 'lighter' overview of how things can be done.❖

Networking Personal Computers with TCP/IP

by Craig Hunt
O'Reilly and Associates 1995
450pp.
ISBN 1-56592-123-2
Reviewed by Craig Macbride
<craig@rmit.edu.au>

Here is a book which is trying to do the impossible. Its aim is to cover TCP/IP networking on IBM-compatible PCs, which is a big aim indeed, given the different PC operating systems as well as the vast number of different implementations of TCP/IP stacks and TCP/IP applications for PCs. The manuals which come with some individual PC TCP/IP packages are as big as this book.

The author starts by giving us a fairly extensive rundown on PC hardware, concentrating on those aspects, such as interrupt handling, which are most likely to prove useful to someone adding a network card. Operating systems are also covered, though the ones which this book deals with are limited to MSDOS, the various flavours of MS Windows, (including NT, but not Windows 95) and Novell NetWare. No mention is made of OS/2 at all. PC implementations of UNIX are deliberately left out, nominally because they are covered in other books. It would be just as fair to say that someone setting up TCP/IP on a UNIX system would not need a book like this anyway, as the vast majority of problems with running TCP/IP under MSDOS, MS Windows or NetWare don't exist under UNIX.

The next two chapters are devoted to what you do before setting the network up, such has planning your network, training users, installation, software tools, troubleshooting and so on.

Several chapters follow with the details on setting up TCP/IP in various ways for MSDOS, Windows, Windows for Workgroups and Windows NT. TCP/IP applications, including telnet, ftp, NFS, WWW browsers and name servers are also covered though, inevitably, the examples are generally limited to one particular implementation of each application.

Once past the problems of actually installing the basic working software, the book heads into configuration control, such as how to use a BOOTP or DHCP daemon on your UNIX machine, or possibly even an MS Windows machine, to do the work of configuring each PC on the network, instead of having the configuration physically present on each PC.

The next two chapters cover electronic mail and file and print servers. I would consider these purely as illustrative, considering the number of different UNIX machines and PC TCP/IP packages available. For example, a number of PC packages supply their own remote printing systems, so that NFS, with all its overheads, does not have to be running on the host machine.

Finally, we get to a chapter on getting NetWare to coexist with TCP/IP. Various ways are investigated for getting this job done. The best of them sounds to be the NetWare IP product, but it is not mentioned that this is costly, whereas even Microsoft aren't charging for TCP/IP networking these days! No specific mention is made of integrating NetWare with

Windows for Workgroups, but there should be. This is a job which requires detailed information to tell you about all the additional NetWare drivers you have to get and information on how to use them and integrate the whole mess, but none of this is mentioned in this book.

Networking Personal Computers (with TCP/IP) is one book in which you would want to have an extensive index, so as to be able to find particular details when you need to. The manuscript has no index. I can only hope that the final version has a really comprehensive one.

My overall impression of this work is that it is a worthwhile overview. If you are planning to add PCs to a TCP/IP network and have never got down to the nitty-gritty of PCs before, it's probably a good place to start. However, if you already know TCP/IP, already know PCs and are planning to use a commercial package to put TCP/IP onto your PC, it may be superfluous. If your PC TCP/IP package has well-written documentation, it should cover all that you need, and it will be more relevant, by being specific to the software that you are using.❖

The Keys To Successful UNIX System Management

by Robert King Ables
Prentice Hall 1995
316pp., Soft Cover
ISBN 0-13-1948381-0

Reviewed by Frank Crawford ANSTO
<frank@ansto.gov.au>

When I first saw this book, I wondered how it would fit into the library of your average System Administrator. The author admits, this book is not aimed at performing standard SysAdmin tasks, but rather at the "whys" and "hows" of effective System Administration. Further, while it is written by a UNIX Administrator, most areas are applicable to anyone involved in such a task, from the UNIX Administrator through to Novell CNEs, to Computer Operations Managers.

Once I started reading, it soon became apparent that this is a book that has long been needed by System Administrators, as it goes to lengths to describe why SysAdmins are needed, what they really do, and what are the effects if they don't do their job. It is the perfect book for someone who is starting out as an administrator, or someone who is managing an administrator.

It is the first book I have seen that tries to put it in terms a manager can understand, what a SysAdmin's tasks are, what value they have to an organisation and how they should be treated. For example, the description given of why a SysAdmin service is still needed (despite claims by salesmen):

"To perform effectively as a system administrator, one must understand the concepts of system management, and be at least somewhat technical. Most users still are not at that level, and indeed often do not need or wish to be there. Do you take your car to a mechanic or do you fix it yourself?"

The book is not just about making a case for the employment of SysAdmins, it has lots of useful advice for anyone in the job, especially someone starting new. It gives a very good background to backups, user registration (including paperwork and authorisation procedures), third-party software, hardware devices, heterogeneous systems and system tuning (to name a few). It also goes into detail about the physical environment required for safe computer operations. Finally on the technical side, it goes into detail about networking and security.

After this, it gets into some of the more interesting sections, on the skills required by a SysAdmin, including both administrative and personal skills. It finishes up with a chapter on the career possibilities for SysAdmins, covering such topics as the hiring requirements, job descriptions, career path and how to keep a SysAdmin (especially useful for your boss).

Finally the book includes an extensive glossary, which makes up for some of the lack of detailed technical information. In fact, I found the glossary a very valuable tool, it lists many articles and references that I'd previously heard about but had never really tracked down.

I found this a very valuable book, not for its technical content (which is generally a good introduction but not in depth) but for its attempt to list the reasons for why System Administration is necessary. It is a book that is necessary for both, those contemplating a career in System Administration, as a way to find out what they may be getting in to, for those starting such a career, as a road map for what they need to learn, and for experienced SysAdmins, as something to give to their managers to describe their role. It is well written, easy to read, and didn't suffer from too much

American chauvinism. I certainly plan to pass it onto new starters to given them an idea what they are getting themselves into.❖

Using csh & tcsh
Work Faster, Type Less

by Paul DuBois
O'Reilly and Associates 1995
200pp.
ISBN 1-56592-132-1
Reviewed by Jamie Honan
<jhonan@mpx.com.au>

I've always looked on in bewildered amazement at the arcane antics of the csh user. The history editing mechanism seems baroque to the outsider. In return, I've received condescending remarks about my DOS like use of the Bash "readline" capabilities.

DuBois lifts the lid on the mysticism of csh history editing. DuBois writes for the level of one who is a relative newcomer to UNIX, often covering diverse but related topics to csh. For example, the old chestnut, how to remove a file with unusual characters (such as spaces) in its name is given a whirl. The "cp" command is explained, as are some command line switches.

However, the material is detailed enough that everyone but a real expert should learn something. Other topics covered include quoting, 'backtic command insertion', special characters such as '~', '*', environment variables, background suspension and resumption of processes, aliasing, pushd and popd. Along the way various tips and traps are expounded.

The author highlights the advantages of obtaining and using tcsh, a freeware version of csh. This package fixes a number of bugs found in many csh implementations, and also provides the same command line editing facilities that Bash has, in addition to the traditional csh history editing.

Tcsh has command and file name completion (via the TAB key), and the facilities for tailoring this seem useful and probably not widely known.

This book would be worthwhile both as a reference and introduction if your site has standardised on csh. The description of tcsh's features should tempt many csh users to switch over, but doesn't seem compelling enough if you are already used to a completely different shell.

As a scripting language the author makes no bones about csh's inadequacy, referring directly to Tom Christiansen's damning Usenet FAQ "Csh programming considered harmful". DuBois endorses these sentiments and hence doesn't attempt to describe csh's scripting capabilities.❖

Exploring Expect

by Don Libes
O'Reilly and Associates 1995
566pp., $59.95
ISBN
Reviewed by Jon Wright
Guru Software Services
<guru@acslink.net.au>

"Exploring Expect" is another volume in the O'Reilly handbook series. Like all the others, it provides a complete and thorough reference on the topic at hand as well as an animal on the cover (a chimpanzee). Don Libes, author of "Obfuscated C and Other Mysteries" and co-author of one of my all time favourite books: "Life With UNIX", is actually the one who developed the Expect language. This combined with an easy-reading style, suggests that he is the ideal person to guide a programmer through the intricacies of the language.

Unlike many programming books, the whole idea of Expect is presented gradually and gently. New features are only introduced when there is no other way to solve a particular problem with the functions that have already been presented. Normally this style of approach would present major difficulties for those who like to dip into different sections at random. To overcome this problem, Don has provided extensive cross-referencing in the text, along with two complete indices. In addition, there are two tables of contents (a brief one for the start-to-finish reader and an extensive one for the random hopper), as well as a command appendix.

Expect is a very interesting (though initially non-intuitive) language. The book describes tcl/Expect, the first implementation of the language, but there is also some reference to other versions (e.g. perl/Expect). For those who have never come across the language, Expect resembles the idea of "chat" scripts in UUCP. A simple program consists of expect (i.e. where the program matches input) and send (i.e. where the program produces output) statements. By comparison chat scripts consist of wait-for-this-pattern followed by send-this-sequence with no real alternative actions available.

Expect does not suffer from any such limitation. Basically Expect is the ideal tool for automating (or semi-automating) the user interface any program. This can be done by either replacing the user altogether or more often by using expect as a buffer between the user and the program. Examples of Expect scripts that are discussed in the book include:

- Automating tip and telnet connections.
- Changing passwords (great when you don't have NIS+).
- Automatically accepting some fsck options but prompting the operator for others.
- Clever su scripts that keep the current directory and command history.
- Replacements for the UNIX "script" program that allow recording to be turned on and off during a session.

Moreover Expect is able to interface with programs that require user interaction (e.g., **tip**, **vi**, **fsck**, **dump**). There are various extensions available as well. One that is discussed in the book is Expect combined with tk to create Expectk. This extension allows a programmer to replace the text interface on a program with a Motif-based graphical front-end.

When you download Expect (the book explains how, what and where), it comes with most of the advanced programs discussed in the book. Some of these should be immediately useful. One that I particularly like is a program that allows multiple users to share a single shell at the same time - this is a fantastic teaching tool. There are also programs to play chess, generate passwords, create timeouts in shell scripts, retrieve RFCs, etc.

Overall, I would rate this book as an invaluable reference for those who use (or would like to use) Expect. As a complete novice, this book enabled me to write a useful (saleable) Expect program in less than a week.❖

OSF Distributed Computing Environment: DCE Security Programming

by Wei Hu
O'Reilly and Associates 1995
300pp.
ISBN 1-56592-134-8
Reviewed by Michael Paddon
<mwp@munnari.oz.au>

The Open Software Foundation's Distributed Computing Environment (DCE) provides sophisticated network security and authentication services. Wei Hu's DCE Security Programming is intended as an introductory text and tutorial to this complex subject. The information in the book is current with DCE 1.0, although the author goes to some effort to point out where this differs from the newer DCE 1.1. The last chapter covers this issue in detail.

The first two chapters provide one of the best introductions to basic network security concepts that I've seen. Although the details are heavily DCE specific, the cryptographic techniques and trusted third party authentication mechanisms are of general interest and are concisely explained. The second chapter focuses in great detail on the Kerberos security server which forms the keystone of the DCE security service.

Chapter 3 deals with the DCE security API, which is both complex and large. This is, as you might expect, a little dry, but it does present the information in a very structured and accessible fashion.

Perhaps the most useful part of the book, from the point of view of an applications programmer, is Chapter 4 along with the appendices. Here, we are shown a complete example of how to build a secure application (an employee database, in this case), from design through to actual source code. If this is your first time out with DCE, you will find this alone is worth the price of the book in terms of time saved.

The last few chapters deal with access control lists, and with writing an ACL manager. This is probably the most difficult subject covered, due the complexity of the underlying technology, and the lack of good reference materials. This section is liberally illustrated with code fragments and it does a very good job of covering all the salient details.

Overall? DCE Security Programming is certainly not for everyone, but I would highly recommend it to anyone who needs to implement a truly secure client-server application. Even if you are not going to use DCE (and there are very few alternatives as mature or as complete), you will find this book a mine of information on how such security can and should be implemented.❖

AUUG Business

UNIX & OPEN SYSTEMS USERS

AUUG Election 1995: Returning Officer's Report

David Purdue
Returning Officer
Tuesday, 20th June, 1995

This report gives the results of the 1995 election for the AUUG committee, along with the election of K Robert Elz as an honorary life member, and two changes to the AUUG constitution.

There was only one nomination for the position of President, and so **Michael Paddon** was declared elected unopposed.

There were two nominations for Vice President: Michael Paddon and Glenn Huxtable. As Michael Paddon was already elected as President, **Glenn Huxtable** was declared elected unopposed.

There was only one nomination for the position of Treasurer, and so **Stephen Boucher** was declared elected unopposed.

A postal ballot was held from 15th May to 14th June 1995.

Ballots for this election were sent to the AUUG post office box, where they were collected by the AUUG secretariat and forwarded to me on Monday, 19th June 1995. That evening the votes were counted by myself and Brenda Parsons, the Assistant Returning Officer, at the offices of Sequent Computer Corporation in North Sydney.

A total of 209 ballots were received, of which 11 were rejected:

3 had not filled out or signed the declaration, 5 were student members, who are not voting members, 1 was a newsletter subscriber, 2 were not on the list of financial members provided by the Secretary.

This left 198 valid ballots to count - given the current AUUG membership of 774 Individual members and 342 Institutional, this corresponds to a voter turnout of 19.5%.

The first count was for the position of Secretary. The result was:

David PURDUE . . .	87
Peter WISHART . .	106
No vote or Informal .	5

Peter Wishart is declared elected as Secretary of AUUG, and takes no part in the election of General Committee Members.

The count for General Committee Members was carried out using the Hare Clark system described in the Election Procedures. The following five candidates were elected, listed in the order elected:

Lucy CHUBB
Frank CRAWFORD
Philip MCCREA
David PURDUE
Chris MALTBY

As no candidates remained for the positions of Returning Officer and Assistant Returning Officer, no counts were held for these positions.

On the election of K Robert Elz as an honorary life member, the result was:

Yes	170
No.	15
Abstain/Informal . .	13

Hence Robert is elected as an honorary life member.

Two changes to the Constitution of AUUG were proposed. On the first proposal, ***that membership periods be brought into line with current practise***:

Yes	182
No.	1
Abstain/Informal . .	0

This change is approved. On the second change, ***the introduction of the role of Immediate Past President to the Committee***:

Yes	166
No	16
Abstain/Informal	1

This change is also approved. It should be noted that not everyone who voted returned the constitutional change paper.

This concludes the AUUG Election for 1995. My thanks to Brenda Parsons, Peter Wishart and the AUUG Secretariat (ACMS) for their assistance in running the election.

Editor's note: Due to an oversight, the minutes of the 1994 AGM weren't published in these pages when they should have been. Neither wanting to deprive you of this information, nor to fly in the face of constitutional requirements, we present them now for your edification...

AUUG Minutes of the 1994 Annual General Meeting Auditorium, World Congress Centre, Melbourne Wednesday 8th September 1994

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

Apologies: Michael Tuke.

One hundred (100) AUUG members signed the AGM attendance record as being present at the meeting.

Meeting opened at 6:05 pm.

1. Minutes of Last Meeting

The President (Phil McCrea) advised that the minutes had been published in AUUGN Vol 14 No. 6 Dec 1993.

Motion: That the minutes of the 1993 AGM as published in AUUGN Vol 14 No. 6 (December 1993) be taken as read and accepted.
Moved: Greg Rose/Peter Chubb. CARRIED.

2. Returning Officers Report

A full report of the elections had been published in AUUGN. There were 224 votes, 4 were set aside (no indication of members name), 1 was in the incorrect envelope, 2 (no signature). All positions were filled except for assistant returning officer.All candidates for that position had been elected to a higher position and were therefore ineligible for election to the position.

The results of the election were: President: Phil McCrea Vice-President: Glenn Huxtable Secretary: Peter Wishart Treasurer: Frank Crawford Committee Members: Chris Maltby, Michael Paddon, Rick Stevenson, Lucy Chubb, Stephen Boucher. Returning Officer: David Purdue. Assistant Returning Officer: vacant.

Motion: That the returning officers report be accepted.
Moved: John OBrien/ John Wright. CARRIED.

3. President's Report

The president noted a few highlights from the past year:

- the Kirk McKusik tour had been very successful. AUUG was considering having other such tour but probably not tied to summer conferences since this made organisation significantly more difficult.
- summer conferences this year had been great successes.
- an office had been opened in North Sydney coinciding with the appointment of a new business manager, Catrina Dwyer.
- the corporate sponsorship programme now had three members: Sun, Digital and IBM. Each had contributed $5,000. AUUG was looking to a maximum of 10 corporate sponsors.
- the Australian articles had given AUUG a high profile in the media.
- Jagoda Crawford had done a sterling job as AUUGN editor over the past few years, but she was now indicating that she wanted someone to take over the job in 1995. The meeting recorded its thanks to Jagoda for her contribution to AUUG during her time as AUUGN editor. It was noted that some secretariat support might be considered for the AUUGN editor to reduce the workload.

Motion: That the Presidents report be accepted.
Moved: David Purdue/ Merik Karman. CARRIED.

4. Secretary's Report

The Secretary showed a number of graphs with membership data. These will be published in a future AUUGN.

There had been a significant increase in membership over the last year (around 26% overall). This was probably due to the higher national profile through the articles in the Australian and greater local activities through chapters. It was noted that there had been a significant increase in ordinary (individual) memberships. Rising from 488 last year to 635 this year. There had been a 7% growth in membership just in the period since July 1994.

Membership renewals sent to members were now requesting an update of all information held on members. This would make it easier to keep this information current.

AUUG was looking at options for replacing the current membership card.

Motion: That the Secretary's report be accepted. **Moved:** Michi Henning/Lawrie Brown. CARRIED.

5. Treasurers Report

The Auditors report had been received at 5:20pm on the day of the meeting, so the Treasurer had not had much time to study it.

Income for the year was just over $300,000. There was a loss from last years conference of around $40,000 but a profit from the exhibition of around $60,000.

The opening of an office for the business manager had required AUUG to purchase a number of assets.

Chapter activities had made a surplus of around $42,000 (this figure did not include the Victorian chapter, whose details were still with the auditor). Summer conferences had made around $7,000.

A full Treasurers report would be published in a future AUUGN.

There were comments that AUUG might be selling itself too cheap. The Kirk workshop fee was only $100, the value was much greater than that and AUUG should look at charging more for such events. Members of the management committee noted that a deliberate attempt had been made to keep the fee down and to keep conference fees down, in general. There was discussion about reviewing the conference, methods of charging, fees etc. There was a consensus that raising prices would impact people's ability to attend and so any such move should be considered very carefully. Other options should be explored to raise funds, e.g., the corporate sponsorship scheme. It was noted that AUUG makes significant profit from tutorials, both in chapters and at AUUG94. These should be exploited more.

Motion: That the Treasurers report be accepted. **Moved:** John OBrien/ James Worsley.CARRIED.

6. AUUG96 Tender

The management committee was looking at tenders for running the 1996 conference and exhibition. A request for tenders had been sent out in June 1994. Eleven tenders were now being evaluated.

7. Membership Survey

A membership survey had been conducted and had received over 400 replies. Full analysis of the survey had not yet been completed, but the following points were noted:

- the profile of AUUG members had changed. It was now predominantly systems admin and consultants. The management sector had significant reduced.
- this was the first survey to cover chapters and it showed that they were very successful. Peer contact and good technical content were listed as two reasons for the success.
- the most informative publications for members were UNIX World, Open Systems Review and AUUGN. o the most popular form of UNIX was SunOS.

8. Other Business

There was discussion about the need to consider other venues for the national conference (e.g., Brisbane). It was noted that Brisbane posed problems for the exhibition since many vendors were not currently keen on sites outside Sydney and Melbourne.It was also noted that the responses to the AUUG96 call for tenders had shown that conference management companies did not seem willing to consider sites outside Sydney and Melbourne. There was some discussion on separating the conference from the exhibition.

9. Next Meeting

The next meeting will be held in conjunction with AUUG95 at the Convention Centre, Sydney, in September 1995. The exact date and time will be advised to members.

The meeting closed at 7:05 pm.❖

AUUG Institutional Members

as at 30/07/95

AAII
Aberfoyle Resource Limited
ACAY Network Computing Pty.Ltd.
Actrol Parts
Adept Software
Advanced Software Engineering
Alcatel Australia
Amalgamated Television Services
Amdahl Australia Pty Ltd
Andersen Consulting
ANI Manufacturing Group
Animal Logic Research Pty. Ltd.
Ansett Australia
ANSTO
Anti-Cancer Council of Victoria
ANZ McCaughan
AT & T GIS
Atlas Computer Systems
Attorney-General's Department
Ausnet Services Pty. Ltd.
AUSOM Inc.
Australian Archives
Australian Bureau of Statistics
Australian Centre for Remote Sensing (ACRES)
Australian Customs Service
Australian Defence Industries Ltd.
Australian Electoral Commission
Australian Film Television and Radio School
Australian Information Processing Centre Pty. Ltd.
Australian Medical Enterprise
Australian Museum
Australian National Audit Office
Australian National University
Australian Submarine Corporation
Australian Taxation Office
Australian Technology Resources (ACT) Pty. Ltd.
Australian Technology Resources Pty. Ltd.
Australian Tourist Commission
Australian Wool Research & Promotion Organisation
AWA Defence Industries
B & D Australia
Bartwon Water
Bay Technologies Pty Ltd
BHA Computer Pty. Limited
BHP Information Technology
BHP Minerals Exploration
BHP Petroleum
BHP Research - Melbourne Laboratories
BHP Research - Newcastle Laboratories
Bond University
Burdett, Buckeridge & Young Ltd.
Bureau of Meteorology
Butterworths
Bytecraft Pty. Ltd.
C.I.S.R.A.
Cadcom Solutions Pty. Ltd.
Cape Grim B.A.P.S
Capricorn Coal Management Pty. Ltd.
CelsiusTech Australia
Central Queensland University
Centre for Open Systems Pty. Ltd.
Chief Secretary's Department
CITEC
CITR Pty. Ltd.
Clarity International
Classified Consulting Pty. Ltd.
Clegg Driscoll Consultants Pty. Ltd.
Co-Cam Computer Group
Coal & Allied Operations
Cognos Pty. Ltd.
Com Net Solutions
Com Tech Communications
Comcare Australia
Commercial Dynamics
Commercial Industrial Computer Services Pty. Ltd.
Communica Software Consultants
Composite Buyers Ltd.
Computechnics Pty. Ltd.
Computer Sciences of Australia Pty. Ltd.
Computer Systems (Australia) Pty. Ltd.
Compuware Asia-Pacific
Comsys International Pty. Ltd.
Concord Repatriation General Hospital
Continuum Australia
Copper Refineries Pty. Ltd.
Corinthian Engineering Pty. Ltd.
CSIRO Division of Information Technology
CSIRO Division of Manufacturing Technology
Curtin University of Technology
Cyberdyne Systems Corporation Pty. Ltd.
Cyberscience Corporation Pty. Ltd.
Cybersource Pty. Ltd.
Daedalus Integration Pty. Ltd.
Data General Australia
Datacraft Limited
Datacraft Technologies
Dawn Technologies
DB Bain Group Services Pty. Ltd.
Deakin University
Defence Housing Authority
Defence Service Homes
Department of Business & Employment
Department of Communications and the Arts
Department of Conservation & Natural Resources
Department of Defence
Department of Defence (TC Section)
Department of Education QLD
Department of Environment & Natural Resources
Department of Family Services & Aboriginal & Islander Affairs
Department of Health & Community Services
Department of Lands, Housing & Local Government
Department of Public Administration
Department of State Services
Department of the Treasury
Dept. of Industrial Relations, Employment, Training & Further Education
DEVETIR
Dialix
Digital Equipment Corp. (Australia) Pty. Ltd.
Dominion Systems Pty. Ltd.
DSTO, Lab 73
EASAMS (Australia) Limited
Edith Cowan University
Electricity Trust of South Australia
Electro Optics Pty. Ltd.
Engineering Computer Services Pty. Ltd.
Environmental Resources Information Network (ERIN)
Equity Systems Pty. Limited
Ericsson Australia
ESRI Australia Pty. Ltd.
Execom Consulting
Executive Computing
FGH Decision Support Systems Pty. Ltd.
Financial Network Services
Fire Fighting Enterprises
First State Computing
Flinders University
Fremantle Port Authority
Fujitsu Australia Ltd.
G.James Australia Pty. Ltd.
GEC Alsthom Information Technology
GEC Marconi Systems Ltd.
Genasys II Pty. Ltd.
General Automation Pty. Ltd.
GIO Australia
Golden Circle Limited
Great Barrier Reef Marine Park Authority
Haltek Pty. Ltd.
Hamersley Iron
Hannan Group Computer Services
Hansen Corporation Pty. Ltd.
Heath Insurance
Hermes Precisa Australia Pty. Ltd.
Hitachi Data Systems
Honeywell Australia Ltd.
Honeywell Ltd.
Hong Kong Jockey Club Systems (Australia) Pty. Ltd.
HSB
I.P.S Radio & Space Services
IBA Healthcare Pty. Ltd.
IBM Australia Ltd.
Ideas International Pty. Ltd.
In-Mar-Tech Australia Pty. Ltd.
Independent Systems Integrators
Informatel Online
Information Technology Consultants
Informed Technology
Insession Labs Pty. Ltd.
Insurance & Superannuation Commission

Integration Design Pty. Ltd.
Intelligent Network Development
Internode Systems Pty. Ltd.
James Cook University
Joint House Department
JTEC Pty. Ltd.
Keays Software
Knowledge Engineering Pty. Ltd.
Laboratory Systems Pty. Ltd.
Labtam Australia Pty. Ltd.
Land Information Centre
Land Titles Office
Leeds & Northrup Australia Pty. Limited
Legent Australia Pty. Ltd.
Logica Pty. Ltd.
Lotus Development
Lyons Computer Pty. Ltd.
Macquarie University
Main Roads Western Australia
Maintain Axis Computers
Matcom Technologies
Mayne Nickless Courier Systems
Mayne Nickless Information Tech. Services
Memtec Limited
Mentor Technologies Pty. Ltd.
Mercedes-Benz (Australia)
Message Handling Systems
Metal Trades Industry Association
Mincom Pty. Ltd.
Minenco Pty. Ltd.
Mitsubishi Motors Australia Ltd.
Mitsui Computer Limited
Moldflow Pty. Ltd.
Motorola Communications Australia
Motorola Computer Systems
MPA International Pty. Ltd.
MUA Pty. Ltd.
Multibase Pty. Ltd.
Multiline BBS
Multiuser Solutions Pty. Ltd.
National Library of Australia
National Resource Information Centre
NCOM Services
NEC Australia Pty. Ltd.
Northern Territory Library Service
Novell Pty. Ltd.
NSW Agriculture
NSW Public Works, Information Services
NSW Teachers Federation Health Society
Object Design Pty. Ltd.
Object Oriented Pty. Ltd.
Object Technology International Pty. Ltd.
Office of State Revenue
Office of the Director of Public Prosecutions
ONA
Open Software Associates Ltd.
Opentec Pty Ltd
OPSM
OSIX Pty. Ltd.
Pacific Semiconductor Pty. Ltd.
Pacific Star Communications
Patinum Technology Australia
Peter Harding & Associates Pty. Ltd.
Petrosys Pty. Ltd.
Philips PTS
Port of Melbourne Authority
Powerhouse Museum
PPIT Pty. Ltd.
Primary Industries & Energy
Process Software Solutions Pty. Ltd.
Prospect Electricity
pTizan Computer Services Pty. Ltd.
Pyramid Data Centre Systems
Qantek
QLD Department of Transport
QLD Electricity Commission
Quality Bakers Pty. Ltd.
Quality By Design Pty. Ltd.
Redland Shire Council
Rehabilitation Tasmania
Renison Golfields Consolidated Ltd.
Rinbina Pty. Ltd.
Royal Melbourne Institute of Technology
SCEGGS Redlands Ltd
Sculptor 4GL+SQL
Security Mailing Services
SEQEB Business Systems
Siemens Nixdorf Information Systems Pty. Ltd.
Smallworld Systems (Aust.) Pty. Ltd.
Smorgon ARC
Snowy Mountains Authority
SoftGen Pacific Pty. Ltd.
Software Plus (Australia) Pty. Ltd.
Softway Pty. Ltd.
South Australian Co-operative Bulk Handling
Specialix Pty. Ltd.
St. Gregory's Armenian School
St. John of God Hospital
St. Vincent's Private Hospital
Stallion Technologies Pty. Ltd.
Standards Australia
Stanilite
State Library of Victoria
State Revenue Office
Steelmark Eagle & Globe
Sterling Software
Storage Technology of Australia
Strategic Information Technologies Pty. Ltd.
Sunburst Regency Foods Pty. Ltd.
Sydney Electricity
Sydney Ports Authority
Systek Corporation Pty. Ltd.
System Builder Development Pty. Ltd.
Systems and Management Pty Ltd
Systems Development Telecom Australia
TAB of Queensland
TAFE NSW, Information Systems Division
Tandem Computers
Technical Software Services
TechNIX Consulting Group International
Telecom Australia
Telecom Australia Corporate Customer
Telecom Payphone Services
Telecom Research Laboratories
The Far North QLD Electricity Board
The Fulcrum Consulting Group
The Knowledge Group Pty Ltd
The Preston Group
The Roads & Traffic Authority
The Southport School
The University of Western Australia
Thiess Contractors Pty. Ltd.
Thomas Cook Ltd.
TNT Australia Information Technology
Toshiba International Corporation Pty. Ltd.
Tower Technology Pty. Ltd.
Tradelink Plumbing Supplies Centres
Transport Accident Commission
Triad Software Pty. Ltd.
Turbosoft Pty. Ltd.
TUSC Computer Systems
Unidata Australia
Uninet Consulting Pty. Ltd.
Unisys Australia Ltd.
University of Adelaide
University of Melbourne
University of New England
University of New South Wales
University of Queensland
University of South Australia
University of Sydney
University of Tasmania
University of Technology, Sydney
Unixpac Pty. Ltd.
Vanguard Computer Services Pty. Ltd.
Vanoco Pty. Ltd.
Victoria University of Technology
VME Systems Pty. Ltd.
Walter & Eliza Hall Institute
Wang Australia Pty. Ltd.
Water Board
WCS Australia Pty. Ltd.
Wesfarmers Limited
Western Mining Corporation
Westrail
Woodside Offshore Petroleum
Workers' Compensation Board of QLD
Workstations Plus
XEDOC Software Development Pty. Ltd.
Zircon Systems Pty. Ltd.
Zurich Australian Insurance

❖

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

20% off for AUUG members

UNIX FOR THE IMPATIENT

Second Edition

Paul Abrahams and Bruce Larson

0-201-82376-4
650 pages/ Paperback

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 ì 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

STANDARDS

An Update on Standards Relevant to USENIX Members

by Nick Stoughton
USENIX Standards Report Editor
<nick@hoskyns.co.uk>

The following reports are published in this column:

- PASC
- POSIX.0
- POSIX.1
- POSIX.1g
- POSIX.21
- POSIX.4
- POSIX System Administration
- POSIX Conformance Testing
- Programming for the Real World

Our Standards Report Editor, **Nick Stoughton**, welcomes dialogue between this column and you, the readers. Please send any comments you might have to

A Report From The Chair

by Lowell Johnson,
Chair IEEE Portable Applications Standards Committee

PASC, the IEEE Portable Applications Standards Committee, has been undergoing a lot of changes over the last year or two. The most notable have been the decline in participation, the completion of 27 standards (undoubtedly part of the reason for the declining participation), and the smaller, more focused projects which are now being started. There are more changes coming in the future, but I will defer reporting on them for the present.

The primary topic of this report, and probably the most visible change, is the new organization structure adopted by the SEC (the Sponsor Executive Committee of PASC) at its last meeting on April 27 in Irvine, CA. A reorganization of some sort has been in the works since the October, 1993 meeting in Bethesda, MD. However, I think a little organizational history is appropriate before we get into the new structure itself.

When PASC (then TCOS – the Technical Committee on Open Systems) was younger, there was basically a new working group created (with a chair on the SEC) for each new project approved. Once we had over six or seven groups we started assigning some new projects to existing groups and some to new groups. If a new project was given to an existing group, it was usually because that was where most of the people who would be working on the project were located.

As we grew larger, a significant number of projects were eventually going to either modify or amend existing projects or standards; the first and most complex case being POSIX.1. Special committees had to be created to coordinate the activities of the various working groups. ISO had defined three projects where much (but certainly not all) of the PASC standards would reside when they achieved IS status. They are:

- 9945-1 System Application Program Interfaces
- 9945-2 Shell and Utilities
- 9945-3 System Administration

IEEE then gradually began renumbering the PASC projects to better align them with their eventual ISO homes: for example, all projects that would end up in ISO 9945-1 were renumbered to the form 1003.1x where x was just the next available letter in the alphabet. A complete list of all project numbers (old and new) is appended at the end of this paper.

We then had a very large SEC, with working groups that were not organized in the most logical fashion. As the PASC participation declined and many projects became smaller, the SEC became disproportionately large in relation to the total PASC membership. At its largest, the SEC had about 38 voting members, plus several non-voting liaisons, and other senior people who attended without an official vote. Generally the SEC meetings were attended by about 50 active participants (not counting non-participating observers), out of a total meeting attendance that had dwindled to about 120. Clearly a reorganization and restructuring of the SEC was needed.

A reorganization was initially proposed at the October, 1993 PASC meeting in Bethesda, and has been more or less hotly debated since then. We finally put out an official letter ballot to the then current voting members of the SEC. The results were in before our April, 1995 meeting in Irvine, CA, which broke the issue into three main questions:

- Should the basic reorganization happen?
- Should the SEC Functional Chairs (formerly called Vice Chairs) be granted continued SEC voting status?
- Should the PMC continue to exist? (The Project Management Subcommittee reviews projects and makes recommendations to the whole SEC, but has no absolute authority of it own).

I have paraphrased these a bit for simplicity, but the ballot resulted in the approval of all three questions. However, it was generally accepted that the precise organization specified in the ballot was not perfect, and since this issue had been so contentious, I announced in January that the reorganization would not take effect until after the April meeting, and that motions could be made to "fine tune" the organization during the April SEC meeting itself. There was considerable debate, and strict Robert's Rules of Order were followed, but in the end, there were only three real changes made:

- a very small number of projects were rearranged under the new working groups.
- the office of Coordination Functional Chair was removed.
- the structure of the PMC was changed to allow up to half the members to be from outside the new SEC, and to allow the PMC Chair a vote on the SEC.

The last point is particularly important since it allows PASC members who do not have a vote on the new SEC to have a significant voice in the project approval mechanism. I sincerely urge people to consider participating through this mechanism.

After all this preamble, here is the new PASC organization:

PASC 1995 Organization

System Services	1003.1 (standard)
	1003.1a (addendum)
	1003.1b (RT - std)
	1003.1c (threads)
	1003.1d (more RT)
	1003.1f (TFA)
	1003.1h (SRASS)
	1003.1i (corrigenda)
	1003.1j (advanced RT)
	1003.1k (Serial media)
Shell and Utilities	1003.2 (standard)
	1003.2a (standard)
	1003.2b (more S & U)
	1003.2d (batch)
	1003.2e (Serial media)
System Administration	1387.1 (SA Interfaces)
	1387.2 (standard)
	1387.3 (User mgmt)
	1387.4 (print)
Language Bindings	1003.5 (standard)
	1003.5a (more Ada)
	1003.5b (Ada RT)
	1003.5c (XTI)
	1003.5d (sockets)
	003.9 (standard)
	2003.5 (test methods)
Security	1003.1e (system APIs)
	1003.2c (S&U)
	1003.22 (framework)
Profiles	1003.0 (POSIX Guide)
	1003.10 (Supercomp)
	1003.13 (RT profiles)
	1003.14 (multi-proc)
	1003.18 (POSIX profile)
	1201.2 (drivability)
Test Methods	1003.3 (standard)
	2003r (.3 revision)
	2003.1 (standard)
	2003.2 (.2 & .2a TMs)
	2003.4 (RT TMs)
	1326 (standard)
	1326.1 (standard)
	1326.2 (standard)
	1328 (standard)
	1328.1 (standard)
	1328.2 (standard)
Distributed Services	1003.1g (XTI/Sockets)
	1003.21 (RT Dist. Comm)
	1224 (standard)
	1224.1 (standard)
	1224.2 (standard)
	1238.1 (standard)
	1238.2 (standard)
	1327 (standard)
	1327.1 (standard)
	1327.2 (standard)
	1351 (standard)
	1353 (standard)

[This table was purposely terse so that it could fit on a single page or slide. A complete listing of the PASC projects is provided as an addendum to this article for reference purposes. For those of you who are not yet comfortable with the numbering changes, I have included both the old and new numbers of the affected projects, so the list looks a bit longer than you may have expected.] .

One of the major ramifications of this new working group structure is the changes (both direct and indirect) in the SEC. Obviously the SEC becomes much smaller. However, it must be emphasized that the SEC meetings are open, so anyone may attend. It is my intention to allow everyone present the chance to speak, but I reserve the right to place reasonable limitations on the time a non-member may hold the floor. Besides the right to vote, SEC members are also the only ones who can make a formal motion.

The new SEC has the following structure:

Voting Members of the SEC:

8 New Working Group Chairs
4 SEC Officers (Chair, V. Chair, Secretary, Treasurer)
3 Functional Chairs (Balloting, Logistics, Interpretations)
1 Project Management Committee Chair
4 Institutional Representatives (rotating membership)

This yields a total of 20 votes in the reorganized SEC. However, one person is currently both the Vice Chair and Secretary, but only votes once, so there are effectively only 19 votes. This is quite a change from the 36 to 38 we had a year or two ago.

There used to be six Institutional Representatives (IRs) before the reorganization, but since the number of voting IRs cannot exceed 25% of the total non-IR voting membership, it had to be reduced to four. As stated above, we will continue to allow all IRs (and anyone else) to attend the SEC meetings, but only four will be allowed to vote. The IRs did not wish to select which of them would be allowed to vote, so the SEC held a ballot to determine the voting IRs. The two highest vote getters would sit for two years and the next two would have one year terms. Another election will be held in one year for the two expiring positions. The two expiring IRs may (and probably will) stand for re-election, but any other IR who so wishes may also try for the next term, which will be a full two years. Then every year we will hold another election for the two expiring two year terms.

The vote was very close, and in fact there was a tie for second and third, so a coin toss was used to select which IR would receive a two year term and which the one year term. As a result, EurOpen and X/Open won the 2 year terms, while USENIX and OSSWG (a US Navy user group) received the one year terms. I wish to express thanks for the input and support from all the other IRs and invite them to continue to attend our meetings and stand for election at the April, 1996 meeting.

It is our intention that the SEC will become mostly an administrative body with the bulk of the technical decisions being made in the new "large" working groups. Some people have dubbed these the "super groups," but that is strictly unofficial.

Another result of the reorganization is that the steering committees all disappear. Their work of coordinating the various interacting projects will now be done in the plenary sessions of the new working groups. A few special cases (such as testing amendments and security) will be worked out between the groups.

It is our hope that these changes will result in faster and more efficient SEC meetings, but the primary benefits should be to reduce the administrative work that must be done by each project "leader." Nothing was mandated, but it was suggested that each working group have a vice-chair for each project, which may effectively be the chair, or the technical leader for that project. With the new organization, only the eight working group chairs should have to deal with the paperwork involved in the standards process, thus leaving other people more time to do the technical work.

Only time will tell how effective this new organization is, but it is an evolving process after all, and more changes are probably already looming in the future.

PASC Projects

1003.0	Guide to POSIX Open System Environment
1003.1	System API C-Binding (Standard approved 9/90)
.1a	System API Extensions – C-Binding (PAR revised 3/94)
.1b	New # of old 1003.4 – Real Time (Standard approved 9/93)
.1c	New # of old 1003.4a – Real Time Threads (Standard approved 6/95)
.1d	New # of old 1003.4b – Real Time API Extensions
.1e	New # of old 1003.6.1 – Security API Extensions
.1f	New # of old 1003.8 – Transparent File Access
.1g	New # of old 1003.12 – Protocol Independent Interfaces
.1h	Reliable, Available, Serviceable Systems (PAR approved 9/94)
.1i	Technical corrections to 1003.1b-1993 (Standard approved 6/95)
.1j	Advanced Realtime Extensions – C (PAR approved 10/94)
.1k	Serial Media APIs (PAR approved 6/95)

.1 LIS Language Independent form of 1003.1 (see P1372)

1003.2 Shell and Utilities (IEEE Standard approved 9/92)

.2a User Portability Extensions (published with 1003.2)

.2b Shell and Tools – Corrections and Extensions (PAR revised 3/92)

.2c New # of 1003.6.2 – Security Shell & Utility Extensions (PAR 9/93)

.2d New # of part of 1003.15 – Batch Shell & Util Extensions

.2e Serial Media Shell and Tools (PAR approved 6/95)

1003.3 General POSIX Test Methods (Standard approved 3/91)

.3.1 Test Methods for System API (approved 10/92 – see 2003.1)

.3.2 Test Methods for Shell and Tools (see 2003.2)

1003.4 Real Time Extensions [published as 1003.1b] (Standard approved 9/93)

.4a Real Time – Threads Extension [will be published as 1003.1c]

. 4b Real Time System API Extensions [becomes 1003.1d]

1003.5 Ada Binding to System API (9945-1) (Standard approved 6/18/92)

.5a Technical corrections to 1003.5-1992

.5b Ada Realtime Bindings (old .20 for .4 & .4a) (PAR approved 7/91)

.5c Ada Binding for Protocol Ind. Interfaces – XTI (PAR approved 10/94)

1003.6 Security Extensions (original document split in 2 parts)

.6.1 Prot, Audit, & Control Interface – Amend to 1003.1 [see 1003.1e]

.6.2 Protection and Control Utilities – Amend.to 1003.2 [see 1003.2c]

1003.7 System Administration Interface [see 1387.1 – changed 10/93]

.7.1 Sys Admin Print Management (PAR approved 1/92) [see 1387.4]

.7.2 Sys Admin Software Administration (PAR app 1/92) [see 1387.2]

.7.3 Sys Admin – User & Group Account Management [see 1387.3]

1003.8 POSIX Transparent File Access Interface (see 1003.1f)

.8a Shell and Tools: TFA Utilities (PAR approved 1/92)

1003.9 Fortran 77 Bindings to POSIX (Standard approved 6/92)

1003.10 Supercomputing Appl. Environment Profile (Standard approved 6/95)

1003.11 Transaction Processing – Support withdrawn by SEC 4/93

1003.12 Protocol Independent Interfaces – C [became 1003.1g]

1003.13 Real Time Application Profiles

1003.14 POSIX Multiprocessing Application Environment Profile

1003.15 Batch Queuing Extensions (to be split, see 1003.2d for .2 part)

1003.16 System API – C-Binding to the LIS (PAR withdrawn 7/93)

.16a System API Extensions – C-Binding to LIS (PAR withdrawn 7/93)

1003.17 Directory Services [replaced by 1224.2, 1326.2, 1327.2, 1328.2]

1003.18 POSIX Environment Platform AEP (the "minimum" POSIX profile)

1003.19 Fortran 90 Binding to 1003.1 (Support withdrawn 9/93)

1003.20 Real Time Ada Bindings (PAR approved 7/91) [became 1003.5b]

1003.21 Real Time Dist Systems Communications (PAR approved 3/93)

1003.22 Guide to Open Systems Security Framework (PAR approved 3/93)

1201.1 Interfaces for User Portability (support withdrawn 9/94)

.2 Recommended Practice on Driveability

.x Direct Balloting of X Library (never officially submitted)

1224 OSI Abstract Data Manipulation – LIS (Standard approved 3/93)

.1 OSI X.400 Messaging API -LIS (Standard approved 3/93)

.2 Directory Services API – LIS (Std approved 3/93)

1237 Support withdrawn by the SEC 10/90

1238 Common OSI API

.1 OSI FTAM API (PAR revised 6/94)

1326 Test Methods for 1224 (Standard approved 3/93)

.1 Test Methods for 1224.1 (Standard approved 3/93)

.2 Test Methods for 1224.2 (Standard approved 3/93)

1327 OSI Abstract Data Mn API – C-Binding (Standard approved 3/93)

.1 OSI X.400 Messaging API – C-Binding (Standard approved 3/93)

.2 Dir Services API – C-Binding (Standard approved 3/93)

1328 Test Methods for 1327 (Standard approved 3/93)

.1 Test Methods for 1327.1 (Standard approved 3/93)

.2 Test Methods for 1327.2 (Standard approved 3/93)

1351 OSI API – ACSE & Presentation Layer (Standard approved 9/94)

1352	LIS Test methods for 1351
1353	OSI API – ACSE & Presentation (C-Binding) (Standard approved 9/94)
1354	C Language Test Methods for 1353
1372	System Interfaces (old 1003.1 in LIS form)
1387.1	System Administration Umbrella document
.2	System Admin. Software (old # 1003.7.2)
	System Admin. User / Group (new number of 1003.7.3)
.4	System Admin. Printing (new number of 1003.7.1)
2003	Test Methods for OSE (revision of 1003.3-1991) (PAR approved 7/92)
.1	Test Methods for System API (Standard approved 10/92)
.2	Test Methods for Shell and Tools (in recirculation ballot)
.4	Test Methods for Real Time Extensions (PAR approved 1/94)
.5	Test Methods for System API – Ada Binding (PAR approved 9/94)

POSIX.0: Guide to POSIX OSE

Kevin Lewis <klewis@granpa.enet.dec.com> reports on the April 24-28, 1995 meeting in Irvine, CA

Draft 18 of the POSIX.0 guide was submitted to the IEEE Review Committee (RevCom) at its March 15 meeting. RevCom noted that one change was made to the draft without that change being circulated to the ballot group. Thus, much to the appreciation of the fatigued ballot coordinator who is also the author of this snitch, RevCom gave approval conditional upon the recirculation of the change and upon the receipt of no new objections.

This recirculation was completed on April 24. One new negative ballot was submitted but this ballot simply restated an objection that had been previously been circulated. The ballot summary resulting from this last recirculation is as follows:

58 affirmative
9 negative
9 abstentions
93% return
86% affirmative

At this point, the ballot coordinator has made contact with the IEEE RevCom Secretariat to review the recirculation results. It is the humble (and I believe accurate) opinion of this writer that the conditions set by RevCom on POSIX.0 have been met and full approval should be rapidly forthcoming. The next step, upon receipt of full approval, is submittal of draft 18 to JTC1 for final balloting.

The future meeting plan for POSIX.0 is to have a one day meeting at the July meeting in Nashua to declare victory and for the group to enjoy a dinner together. As far as future work is concerned, a study group on User Profiles has spawned out of POSIX.0 and is now operating on its own. It appears that it will become a working group by the end of 1995.

I think it is now safe to say the POSIX.0 Guide Project is done (or to quote Jason Zions, "It's cooked!")

Report on POSIX.1: System API

Shravan Pargal <pargal@ironwood.cray.com> reports on the April 24-28, 1995 meeting in Irvine, CA

This meeting in Irvine was a busy one for the POSIX.1a working group. Draft 12 of POSIX.1a was sent out to the ballot group in November, with the ballot closing on March 15th, 1995. Ballots were in and needed to be worked on during this meeting. The following were the primary agenda items at this meeting:

- Removable media plenary
- Ballot resolutions

The first issue tackled after the meeting was called to order was a plenary with the removable media group. Chuck DeBaun (Fermilab) presented a proposal to the working group for a set of functions for serial media control. The proposal covered `open()`, `close()`, `read()`, and `write()` issues related to serial media. In addition, there was a proposal for additional interfaces to formalize and expand `ioctl` like functions (to be known as `mtio` functions) as related to serial media.

The working group split up into smaller groups of two or more members to tackle the large number of ballots that needed resolutions. Each group was assigned a chapter to work on to resolve "easy" ballot requests. In addition the groups tried to sort the ballots by issue, forming sub-groups within each chapter. The groups then brought lists of issues back to the working group as a whole for discussion. After working on ballot classification and resolution for two days, the group discussed the state of the returned ballots. Ballots returned seemed to be split into three roughly equal chunks: checkpoint/restart, resource limits, and everything else. The working group decided (thanks to a suggestion from Nick Stoughton and Chuck Karish), to split the POSIX.1a project authorization request (PAR) into three PARs, submitting checkpoint/restart and resource limits as separate standards. Two new PARs will be submitted to the SEC 45 days in advance of the next meeting. However since the earliest this could be approved is in July, the working group decided that a final decision would be held off until the July meeting.

In between ballot resolution, discussions were held on proposals for `getshconf()` submitted by David Willcox and signals submitted by Alan Rowe. The `getshconf()` interface was defined to meet a need to separate POSIX.1 and POSIX.2 symbols and yet be able to get one from the other in a portable form. The signals discussion will continue through e-mail.

The POSIX.1a WG will continue to work on resolving ballot objections and interpretations resulting from the Draft 12 ballot. A final decision to go ahead and form the new project teams for checkpoint/restart and resource limits will be made at the next meeting.

Report on POSIX.1g: Protocol Independent Interfaces

Chris Harding <charding@datlog.co.uk> reports on the April 24-28, 1995 meeting in Irvine, CA

The POSIX.12 group (now known as POSIX.1g) was formed to standardize a protocol-independent interface for process-to-process communication within POSIX. Two candidate interfaces were presented to it: the X/Open Transport Interface (XTI) and Berkeley sockets. In spite of intense – sometimes very heated – discussion, the group could not choose between them. Also, it felt that both of these interfaces were at too low a level, and that a simpler interface was really what was needed.

In its early days, the group was well-attended. It had plenty of resources, and thought that it had plenty of time. It decided to specify a Detailed Network Interface (DNI), which would be either XTI or sockets, and a Simple Network Interface (SNI), which would be a new interface that it would develop.

In line with then current POSIX practice, it would have to create a language-independent specification (LIS) and test methods for the SNI and the DNI, and create a new C language binding for the SNI, as well as tidying up the DNI C language binding.

Attendance at POSIX meetings dwindled. POSIX.12, like all the other groups, suffered from this. Completing the program within a reasonable time became impossible.

The group therefore decided to focus its efforts:

- The SNI was deferred until the DNI was done.
- The argument between XTI and sockets was cut short; the group decided to standardize both C language bindings, and to create a single LIS for them.
- Because no one else in POSIX would do it, the group also decided to standardize `poll()` and `select()`, which are needed by many applications of XTI and sockets.
- XTI was to be specified in conjunction with OSI and Internet protocols. The sockets interface was to be specified in conjunction with OSI and Internet protocols, and for local inter-process communication.
- Test methods for XTI were available from X/Open; these would have to be tidied up, and new test methods for sockets and the LIS would have to be written.

On this basis, the group made good progress.

The POSIX Sponsor Executive Committee then dropped its requirements to produce LIS and test methods for POSIX specifications. The POSIX.12 LIS was by this time nearly complete. The group decided to continue with its LIS (although it was not clear exactly how this would eventually be published). No work had been done on test methods, so the group was happy not to have to worry about them anymore.

At about this time, the name of the group was changed to POSIX.1g.

A ballot group was formed which is much larger than the working group. It reviews drafts but does not meet. The POSIX.1g working group has been fortunate because the ballot group contains a number of people who have been prepared to spend considerable time and effort in reviewing the drafts.

A draft (draft 4.0) containing the two C language bindings and the LIS was prepared and submitted to ballot. There were 1188 ballot objections and comments on draft 4.0, and it was well short of the 75% approval that is needed for it to be accepted as an IEEE standard.

By the time of the April 1995 POSIX meeting, the standard had been refined in ballot through draft 4.1, which received 452 objections and comments, draft 5.0 (299 objections and comments) and draft 6.0 (165 objections and comments).

It had also been slightly expanded in scope, since several ballot objections called for functionality to be added. The main additions were:

- two new sockets functions that are needed because POSIX does not include all of the features of BSD UNIX.
- `getaddrinfo()` translates the name and/or location of a service into socket addresses and associated information.
- `isfdtype()` determines whether a file descriptor is associated with a socket.
- `sockatmark()` determines whether a socket is at the out-of-band mark.

- various Internet address-resolution functions, such as gethostbyname().

- support for a minimal 7-layer OSI stack (mOSI) under XTI.

The question of supporting New Generation IP has been raised, but the group felt that this was not yet sufficiently mature for standardization.

There were just four regular (more or less) attendees at the April meeting: Reed Adams, Greg Bussiere, Steve Case (the group vice chair), and Chris Harding. A number of people who had been strong contributors at previous meetings were unfortunately unable to come this time, because of funding limitations, pressure of other work, or job changes. They included Bob Durst (the group chair), Mike Karels, Karen Olsen, Kurt Matthys, Keith Sklower, and Steve Wise.

The meeting was almost exclusively devoted to resolution of ballot comments.

There have now been a total of over two thousand comments and objections. It is much easier to handle them on line than on paper. The group has increasingly been making use of laptops, with megabyte ballot-comment files being passed around on floppies. This time, all work was computer-based, and the comments were never even printed out.

There have been progressively fewer comments and objections on each draft (a good sign!). This time, in spite of the reduced attendance the group produced provisional resolutions for almost all comments on the current draft, and also cleared up most of the comments that had been deferred from previous drafts (there were quite a few of these – particularly from the first balloted draft).

The most controversial technical issue has been whether XTI and sockets should both be required, or whether a conforming implementation can support just one of them. The ballot group was polled some time ago on this, and was split 50/50. A compromise was reached: XTI and sockets are specified as separate options, but conforming implementations must support both. A profile for a minimal realtime implementation, for example, might however require just one of the options. (An implementation could then conform to the profile without conforming to the base standard – an interesting situation.) There were a number of objections against this compromise, all of which called for the requirement to support both XTI and sockets to be dropped. However, other balloters had said that they would object if the requirement were dropped. It transpired that dropping the requirement would turn more "yes" votes to "no" than it would turn "no" votes to "yes." The requirement was therefore kept.

One major decision was to move the LIS into an informative annex. The ultimate fate of the LIS is still not clear. It provides a logical basis for putting forward XTI and sockets as parallel interfaces. Also, it provides a complete specification of how the API relates to the underlying protocols; nether the XTI nor the sockets specification does this. But it does not add anything to the definitions of the XTI and sockets functions. And there have been objections to including it as normative material, even though the C language bindings specifications have always had priority in case of any conflict.

Another question that was resolved concerns the form of the draft. There is a POSIX Document Structure Plan, which calls for drafts to be laid out as though they were to be printed as part of the complete POSIX document set. This would probably mean POSIX.1g starting at chapter eighty-something. Its purpose is to ease the integration of all the parts of POSIX. But integration is proceeding one project at a time, and the turn of POSIX.1g will not come for some years (no one is quite sure exactly when it will be). The organizations that fund participants' attendance at POSIX.1g meetings are not prepared to continue this funding indefinitely, and the group wants to be done by the end of this year. The current draft is organized as a self-contained document (but with one chapter that describes changes to POSIX.1). The group now intends to submit it to the IEEE Standards Board in this form.

With the changes decided on at the meeting, over 75% of balloters should vote affirmatively.

But there will still be a further two or (more probably) three ballot recirculations. These will tidy up loose ends, and ensure that balloters are happy with the actions that have been taken to meet their objections. The first of them will be a 10-day recirculation starting on June 19th. A subsequent recirculation could take place between the July and October POSIX meetings. If necessary, there could be a further recirculation before the end of the year.

Report on POSIX.21: Realtime Distributed Systems Communication

Craig Meyers <bcm@SEI.CMU.EDU> reports on the April 24-28, 1995 meeting in Irvine, CA

The goal of the POSIX.21 working group is to develop a language-independent specification (LIS) and Ada language binding for interprocess communication in the realtime distributed systems domain. The group is expected to address realtime needs in the following areas: protocol management, endpoint management, connection management, unicast data transfer, multicast data transfer, multicast group management, transaction services, event and error management, and performance monitoring. The interface will also provide support for the use of priorities, synchronous and asynchronous interactions, and the ability to bound the blocking that a calling task is able to accept through the use of timeouts. Similar to POSIX.1g, the interface will be developed in a protocol-independent manner with mappings provided to existing protocols. It is expected that a mock ballot of the LIS will be initiated after the July POSIX meeting.

The following is a summary of the major topics discussed at this meeting:

- buffer management, including allocation of buffers to application vs. service provider space.
- endpoint management; purge operations were removed and hard vs. graceful deletion of an endpoint was discussed.
- directory services LIS and models, including location-independent and location-dependent.
- event management; including ways in which the application becomes aware of events (poll vs. callback, for example).
- overall data transfer models in an attempt to harmonize individual models (e.g., unicast, multicast, and broadcast) within a larger framework.
- performance measurement; several issues were resolved.
- prototyping; those interested in this topic had separate meetings and discussed implementation considerations for the initial work.

A transcription of the minutes are available by anonymous FTP from *ftp.sei.cmu.edu* in the file *pub/posix/admin/min-95-04*. Persons interested in working with the group are requested to contact Craig Meyers at *bcm@sei.cmu.edu*.

Report on POSIX.4

Joe Gwinn <GWINN@SUD2.ED.RAY.COM> reports on the April 24-28, 1995 meeting in Irvine, CA

What is the POSIX.4 Working Group doing?

The big news is that as part of the general reorganization of the POSIX working groups, the POSIX.4 Working Group no longer exists, having been largely replaced by 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 will be the first under the new organization. Expect some chaos. In spite of this internal organizational motion, the same people are working on the same documents as before.

The POSIX.4 WG has five major projects: POSIX.1c, or pthreads (formerly known as POSIX.4a), POSIX.1d (formerly known as POSIX.4b), POSIX.1i, POSIX.1j (informally known as "POSIX.4d"), and POSIX.13, discussed below.

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, for almost $100 US. Ouch! Anyway, the IEEE accepts credit card phone orders.

POSIX.1c (pthreads) passed International Standards Organization (ISO) Committee Document (CD) balloting in February 1995 with only one editorial comment, and Draft 10 will be submitted unchanged to the IEEE Standards Board for approval at their June 1995 meeting. Approval appears likely, at last. (*Editor's note: POSIX.1c was approved at this meeting*). After approval, POSIX.1c will also be merged into POSIX.1-1990 (or, more likely, IEEE Std 1003.1b-1993), which will take a year to get to a published merged book, if history is any guide. This published standard will likely be called "IEEE Std 1003.1c-1995," and, with luck, will be available in late 1996.

POSIX.4b (now POSIX.1d) is in ballot resolution, which is almost complete, except in the areas under discussion with the members of the Common Reference Ballot. Interrupt Control has been a particularly contentious area. The current plan is to recirculate after the April 1995 POSIX meeting, if possible.

POSIX.1d, around 130 pages in length, 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 to both measure and bound 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") went to ballot in November 1994, and is currently in ballot resolution. (*Editor's note: this standard too was approved at the June 1995 IEEE RevCom meeting, making it the fastest standard on record from Project Inception in July 1994 to completed standard in June 1995!*)

POSIX.1i consists of correcting the POSIX.4 interfaces to remedy clashes detected when POSIX.4 was merged into POSIX.1-1990 to yield POSIX.1b-1993, as well as minor problems discovered by early implementors of POSIX.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.4d" (now POSIX.1j), approved as a project by the Sponsor Executive Committee (SEC) at the October 1994 POSIX meeting, is being polished in the POSIX.4 WG. Draft 3 underwent mock ballot at the January 1995 POSIX meeting, and Draft 4 has been sent out in the March 15th POSIX mailing. POSIX.1j was voted out to ballot at the April 1995 POSIX meeting. Ballot will be after the July 1995 POSIX meeting. Watch out for the ballot group invitation from the IEEE. I will announce the opening of the ballot group when dates are available. Actual first ballot is scheduled for 8 September 1995 to 8 October 95; the ballot group will therefore close something like 1 September 1995.

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 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; and Persistent Notification for Message Queues, an option for POSIX.1b-1993 Message Queues.

POSIX.13 ballot resolution is nearly complete, and POSIX.13 will probably be recirculated after the April 1995 POSIX meeting.

POSIX.13, 100 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 POSIX.1b-1993 (POSIX.1 plus POSIX.4) and POSIX.1c (threads), 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.

A number of new items, as yet un-numbered and homeless, are under consideration, including a Trace Interface, and various issues relating to time and clocks.

Trace Interface. The purpose of these 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 POSIX.1b-1993, section 14.1.4, which currently defines CLOCK_REALTIME and TIMER_ABSTIME, are under discussion. The new kinds of clock are tentatively named CLOCK_SYNCHRONIZED and CLOCK_FUNDA MENTAL. 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.

CLOCK_REALTIME specifies only that the resolution be 20 milliseconds or finer, a complex way of saying that one second resolution isn't quite good enough for realtime. The value was chosen to allow platforms with the usual 100-Hz or 60-Hz clock tick rate to satisfy the requirement.

TIMER_ABSTIME is used to cause a timer to be absolute (wait until 10:00), rather than relative (wait 10 seconds).

CLOCK_SYNCHRONIZED gives user-controlled access to a local (to a platform) clock synchronized with an external clock (or set of clocks). Synchronization could be with a local GPS receiver, or by the Network Time Protocol (NTP) to a network timeserver. The details are implementation dependent. Synchronized clocks are allowed to run slightly fast or slow, and to make small (up to one second) jumps forwards or backwards, to account for leap seconds and the like. If NTP is active on a workstation, the standard `gettimeofday()` returns synchronized time; the point of having a named clock kind is to allow the user to express the true requirement.

CLOCK_FUNDAMENTAL gives user-controlled access to a local (to a platform) clock whose value steadily and smoothly increases. CLOCK_FUNDAMENTAL is uniform, monotonic, and continuous (to the resolution of the clock implementation), and also allows direct implementation of Ada95 (section D.8) Monotonic Time. Fundamental time always runs at the same (perhaps slightly inaccurate) rate, and never jumps. It may, however, give the same value twice, if called faster than the tick rate of the clock implementation. A clock cannot be both fundamental and synchronized, although a fundamental clock and a synchronized clock can certainly be approximately equal. Fundamental time is useful when the issue is the measurement, tracking, or prediction of the trajectory of a physical object; and for the accurate measurement of intervals. The details are implementation dependent.

Under discussion is which clock the various time-related POSIX.1b-1993 APIs will follow, where no argument to specify a clock is provided. Relative timers (wait 10 seconds) will most likely use or act like they use CLOCK_FUNDAMENTAL, while absolute timers (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.

Report on POSIX System Administration: Software Administration

Jay Ashford <ashford@austin.ibm.com> reports on the April 24-28, 1995 meeting in Irvine, CA

The April 1995 meeting of the POSIX Software Administration group marked an historic event – P1387.2 became the first POSIX system administration project to complete its work. System administration in general has been a big challenge for standardization due to the exceptionally rich and diverse heritage of administrative tools, techniques, and policies. Blending these together into a coherent standard is a very challenging task.

Software administration is certainly one of those areas with a rich and diverse legacy. P1387.2 provides the basis for standardized software administration. It includes a media organization for install images, along with commands to install, remove, configure, and verify software. Commands are also included to create distribution images and to merge distribution images. Along with these commands is an extensive description of their behavior. Provision is also made for tracking what software is installed and what its level is. All of these commands provide distributed function so that installation (removal, configuration, etc.) may be directed on one system to occur on any number of systems throughout a network.

Key to this process is the organization of the distribution images and the installed software. The units include files, filesets (groups of files), and products. This is made a bit richer by the use of sub-products (collections of filesets and/ or other sub-products within a specific product) and bundles (collections of filesets, sub-products, products, or other bundles anywhere within a distribution or installed software).

One very valuable addition to this effort is a set of concessions to operating systems not based on POSIX.1 and POSIX.2. In particular, there are exception conditions that allow such systems as DOS to conform to P1387.2.

P1387.2 has passed its ballots within IEEE, is being sent to the IEEE Standards Board for its approval, and has been sent to ISO for both registration as a Committee Draft and for ballot to progress to Draft International Standard.

Copies of the current draft, P1387.2/Draft 14a, April 1995, are available.

New work in this area is planned to begin at the October POSIX meeting. The suggestions to date include a guide to best use of the current standard, a profile for DOS (and related) systems, version and patch/fix management, policies in distributed management (especially related to what constitutes "success") and associated recovery policies, file sharing and client management, hierarchical distribution, scheduling, and queueing and queue management. There is some leaning toward having each of these efforts supported in a separate POSIX project so that each can proceed atits own pace. No firm decision, however, has been made in that regard.

In addition to the work being done with POSIX, there is a complementary effort underway within the System Management Working Group at X/Open. The X/Open effort is striving to provide the necessary specifications to permit distributed interoperability. This is needed because P1387.2 does not specify the means by which distributed function is to occur.

Report on POSIX Conformance Testing

Kathy Liburdy <kathy.liburdy@eng.clemson.edu> reports on the April 24-28, 1995 meeting in Irvine, CA

With 2003r still in ballot resolution, the newly formed Automated Testing Study Group was the primary source of conformance testing activity and entertainment at the April meeting. The agenda began with a presentation by John Hill on Formal Description Techniques (FDTs) and discussion generated by this presentation dominated the remainder of the meeting.

Central to the discussion were the JTC1 directives which encourage the use of FDTs for new and existing standards in general. Soon after John's description of the directives, members of the study group suggested that the group draft a recommendation to the SEC encouraging PASC to adopt a policy of following the JTC1 directives. The nearly instantaneous polarization of positions regarding this idea suggests that unfounded beliefs may be contributing, at least in part, to the perception of FDTs. In fairness to the image of FDTs, the following two statements are presented as myths:

- Formal Descriptions are Language Independent Specifications. Not always. Formal descriptions can be either programming language dependent (e.g., ADL is heavily biased toward the C language) or programming language independent (e.g., Z is not based on any programming language).
- Formal Descriptions guarantee automated testing. They do not. While some formal description languages directly support automated testing (e.g., CATS), others were created with different goals influencing their design (e.g., Z). The impact on test development from formal descriptions which are not directly translatable into executable tests is not clear.

Closer to reality is the following statement regarding FDTs: FDTs encompass a rich and diverse collection of formal specification languages. In the same way that programming languages may be characterized by different features and limitations, so may formal specification languages.

Formal specification languages share in common a mathematical basis which permits precise, unambiguous descriptions. A precise, unambiguous description is necessary (but not sufficient) for automated translation of descriptions into tests.

While the basic relation of automated testing to FDTs is clear, the relation of FDTs to standards development in general is not. The multitudes of domains in the software standards arena coupled with the diversity of characteristics of formal specifications languages presents an unenviable burden for anyone (or any group) attempting to justify a sweeping statement such as "Use FDTs for all standards." Pragmatic issues such as the impact on participation by volunteers to develop the standards and the understandability of the final product should be thoroughly studied in addition to the acclaimed benefits of FDTs.

Fortunately, this task is far removed from the charter of the Automated Testing Study Group. If enough interest in applying FDTs to standards development is generated by this study group and elsewhere, a new study group may be formed to study these issues.

The Automated Testing Study Group will reconvene in Nashua, NH to further discuss issues related to automated testing. The real issue at hand may not be the technical merits and theoretical issues of FDTs, but rather: do the current rules and politics of PASC allow a test methods standard to be developed using a formal specification language which supports automated testing? If the answer is no, then the study group should consider ways to open the door for testing standards which support automated testing. If the answer is yes, the study group should probably dissolve and send its members on their way to do real work.

Meanwhile, a stalled effort in the development of the 2003.5 standard and a funded effort to develop a test suite for POSIX.5 has created a reversal of previous PASC approaches to conformance testing: the work on the test suite implementation is getting ahead of the standardization effort! Testing standards in PASC have traditionally preceded the implementation of the test suite implementation. The (unplanned) approach by the POSIX.5 Test Suite/2003.5 Standard effort is more along the lines of conventional standards development: standardize based upon existing practice (if there is sufficient reason to standardize at all).

The value of a test methods standard remains questionable. In keeping with the model established by base standards, the standardization of "testing" interfaces might be more appropriate than attempting to standardize a restatement of the base standard. This observation is based on experience gained during the development of the test suite for POSIX.5 in cases where required behavior is stated, but interfaces to confirm the behavior are missing.

Another possibility which is emerging from the POSIX.5 test suite development is that of submitting the testing strategy used in the test suite implementation for consideration as a Recommended Practice. This approach has some appeal in that the open review provided by the balloting process may enhance the corresponding test suite. Regardless of the outcome, this effort has succeeded in raising some important questions about conformance testing in open systems standards. Hopefully, the POSIX.5 test suite experience will provide insights into these issues. The test suite is scheduled for completion in October 1995.

Report on POSIX.4 – Programming For The Real World

by Nick Stoughton
<nick@pert.co.uk>

[*As it turns out, this is a book review – Ed.*]

Those of you who have followed the Standards Activity column in *;login:* over the years will doubtless recognise the name Bill O. Gallmeister, one time vice chair of the POSIX.4 working group, and long time snitch on realtime work.

Bill's practice in writing snitch reports has clearly paid off in his new book, *POSIX.4 – Programming For The Real World*, published by O'Reilly. At some 550 pages, this is no bedtime story, but the style throughout the early chapters makes easy reading. The trouble with reading a book written by someone you know well is that you can almost hear him saying some of the phrases and explanations. Bill's character and personality have tainted those early chapters, but as an easy going Californian BWG (Bearded White Guy), that taint is in no way offputting.

The book consists of two principle sections: the first is a simple introduction to writing portable applications using the POSIX.1b interfaces, while the second half is a reference section to those interfaces, presented in nice familiar man page format. This format is just how I like it; after I'm done reading the introductory stuff, all I want is a good reference book. I rarely if ever go back to look at primers once I've read them. This book is laid out in a style that means I am likely to use it every working day for the rest of my career (or for a long time at least).

Bill points out that Realtime programming (that's the official POSIX spelling of realtime) is not something strange that only missile guidance people need worry about. True, many of the interfaces are strange for programmers, who are used to building compilers and editors and such like tools. But to people building end-user applications, predictability *is* vital.

In my personal experience, over the last 15 years or so I have gone from writing 100% system type applications, to a point now where about 50% of what I am designing and building is end-user applications. True, I haven't had to write too many embedded realtime systems, but I do fall into the class of people that Bill classes as real world programmers. As more and more computer systems are delivered to end users, more and more of us are going to have to get used to real world things. The only trouble is, having set the scene so well in the preface and chapter 1, he only uses embedded, real realtime examples for the rest of the story!

As I said, the book has two principle parts; a seven chapter "primer," written in an easy-to-read style, complete with exercises (though some of these do appear a little labored), followed by a reference section. These sections are of approximately equal size, and this balance is excellent. The reference section should probably have covered *all* of POSIX.1b, including the old, perhaps familiar, POSIX.1 interfaces, but most of these are now so second nature to most of the intended audience, I can understand their omission.

The explanation of feature test macros is at best weak, at worst wrong. However, this is, in my mind, a minor gripe.

Another minor problem is the choice of cover design: a series of pulleys and weights taken from an 19th Century English engraving entitled "Mechanics" suggests that the target audience is mechanical engineers. I think that's unfortunate, because the real target audience is people like us – hackers. Most mechanical engineers I know would have difficulty getting beyond the preface (OK, perhaps I'm being a little unfair), and faced with the pros and cons of using signals as an interprocess communications mechanism would happily go back to their strings and pulleys, drawing cartoons of Bill instead of weights. Nevertheless, since I once studied engineering (I even have a real engineering text book with a remarkably similar cover design at home), and I am English, I think I can live with this one!

The POSIX.1b standard incorporates all of POSIX.1, and the entire standard is covered. However, the intended audience is, as I said, advanced C programmers. This means he does not cover straightforward interfaces like `open`, `close`, `read`, `write`, `fork`, and `exec` in anything more than a cursory fashion. In fact, they are even missing from the reference section. The book concentrates on those new and unfamiliar realtime extensions that were made to POSIX.1 to make POSIX.1b.

The biggest complaint I can make about this book is the use throughout of the old term POSIX.4 – as Bill points out it has been renamed as POSIX.1b, but he finds that term too difficult to grasp. The standard I have on my bookshelf (in those rare moments when I'm not actually using it) is called POSIX.1b, formerly known as POSIX.4. The anachronistic use of the old title grates after a while, and I wish he had bitten the bullet along with all the rest of us in POSIX – we may not like the titles, but they are here to stay, and pretending that the renumbering didn't happen won't make them go away!

Over the years, plenty of people have bitched about what the POSIX.4 working group (and its still called that, even though their standard is part of POSIX.1) did. Bill does not shy away from that, but does clearly point out that though many of these interfaces are not traditional ones, that does not change their historical validity. The book makes almost no mention of the parallel activity of the POSIX.4 working group, the soon to be standard POSIX.1c (formerly POSIX.4a, Threads). This may be a problem in the long term, and I would encourage a future edition that does cover multithreading when that standard is a reality. However, in the short term, what is covered, POSIX.1b, is covered excellently, and I would thoroughly recommend this work to anyone involved in portable application development that might need to consider any realtime aspect.

AUUG Inc.

UNIX AND OPEN SYSTEMS USERS

AARNet

MAIL SERVICE

Dear Site Administrator,

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

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

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

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

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

AUUG
UNIX AND OPEN SYSTEMS USERS

AARNet

Mail Service Application

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

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

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

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

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

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

AUUG
UNIX AND OPEN SYSTEMS USERS

AARNet

MAIL SERVICE
AFFILIATE MEMBERSHIP
APPLICATION FORM

PLEASE PRINT CLEARLY!

Date: ____________________

Name of Organisation/Owner: ____

____

Signed: ______________________________ AUUG Membership No (if known): __________

Name: ______________________________ Position: ______________________________

on behalf of the organisation named above.

Address: ____

____

____

______________________________________ Postcode: ______________________________

Administrative Contact: __________________________ Title: ______________________

E-Mail: ________________________________ Phone: ( ) ______________________

Fax: ( ) ______________________

Technical Contact: ______________________________ Title: ______________________

E-Mail: ________________________________ Phone: ( ) ______________________

Fax: ( ) ______________________

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

Domain Names Requested: ____

____

Gateway Addresses: ____

____

____

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

* * * *

Send this page only to:

AUUG Incorporated
PO Box 366
Kensington NSW 2033

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

AUUG Inc.
UNIX AND OPEN SYSTEMS USERS

AARNet

MAIL SERVICE
AFFILIATE MEMBERSHIP
APPLICATION FORM cont'd.

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

AUUG Inc

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

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

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

____
Signature

____
Date

LOCAL CHAPTER DESIGNATE:

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

TO BETTER SERVE YOU, PLEASE PRINT YOUR CONTACT INFORMATION:

Name/Contact: ______________________________
Position/Title: ______________________________
Company: ______________________________
Address: ______________________________
______________________ Postcode__________
Tel: BH__________________ AH __________________
Fax: BH__________________ AH __________________
email address: ______________________________

Over for Institutional Membership

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

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

____
Signature

__________________ __________________
Title *Date*

Please charge $____________ to my

□ Bankcard, □ Visa, □ Mastercard

Account number: ______________________________
Expiry date: ______________________________
Name on card: ______________________________
Signature: ______________________________

AUUG Secretariat Use

Chq: bank ________________ *bsb* ________________
a/c ________________ *#* ________________
Date: ________________ *$* ________________
Initial: ________________
Date processed: ______________________________
Membership # ______________________________

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

MEMBERSHIP APPLICATION
INSTITUTIONAL

AUUG Inc

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.

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

AUUG Inc

UNIX® AND OPEN SYSTEMS USERS

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

(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 # ____________________