

AUUGN

The Journal of AUUG Inc.

Volume 16, Number 1
February, 1995

Sun RPC: Performance issues

1995 AUUG Elections: information

UNIX: The OS made in heaven?

!AUUGN: Effective C formatting

... plus reviews, Chapter news, and more

AUUG
Inc.

UNIX & OPEN SYSTEMS USERS

AUUGN

Volume 16, Number 1
February, 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@munnari.oz.au

AUUG Executive

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

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

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

Treasurer: Frank Crawford
frank@atom.ansto.gov.au
ANSTO

Private Mail Bag 1
Menai NSW 2234

Committee Members:

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

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

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

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

Rick Stevenson
rick@stallion.oz.au
Stallion Technologies Pty. Ltd.
56 Sylvan Rd.
Toowong, QLD 4066

AUUG Business Manager

Catrina Dwyer
catrina@softway.sw.oz.au
PO Box 366
Kensington NSW 2033
Tel: (02) 959 3656
Fax: (02) 957 6706

Table of Contents

Editorial 3

AUUGN Submission Guidelines 3

The President's Report 5

Readers' Mail 5

Scientific Computing - A forgotten discipline? 5

Opinion: The Operating System they use in Heaven... 6

Phil McCrae

Conferences and Announcements 7

AUUG South Australia - SUMMER CONFERENCE 1995 7

AUUGWet 95 - Northern Territory Chapter of AUUG 7

Australian UNIX and Open Systems Users Group Annual Conference,
September 19-21, 1995 Sydney, Australia 8

SAGE_AU Third Annual Conference & General Meeting (SAGE-AU'95) 9

The Usenix Association 1995 Calendar
of symposia and conferences 10

9th Usenix Systems Administration Conference (LISA IX) 11

5th USENIX UNIX Security Symposium 13

Paper: Transparency and Performance Issues in Sun RPC 17

David Purdue

UNIX Tricks & Traps 22

Edited by Janet Jackson

Consistent binary support for multiple architectures
across a common filesystem *David Leonard* 22

For sites that have NIS netgroups... *Tim Cook* 22

Opinion: Death of the Net? 23

Chris Maltby

Book Reviews 25

Edited by Frank Crawford

*Reviewers: Zoltan Somogyi, Adrian Booth, Danny Yee, Craig Macbride, David J. Hughes,
David Conran, Lawson Hansen, Andrew Wenn*

Building a Successful Software Business 25

Computer Related Risks 26

Firewalls and Internet Security: Repelling the Wily Hacker 26

The Internet Book 27

Internet CD 28

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



The Linux Network Administrator's Guide	28
Motif Tools	30
PGP: Pretty Good Privacy	31
A Quarter Century of UNIX	31
X User Tools	32
Interlude: Yuletide Packets 34	
<i>Lucy Chubb</i>	
AUUG Chapter News 35	
From the Western Front	35
AUUG Inc. - Victorian Chapter (formerly SESSPOOLE)	35
Call for Articles for The Australian	36
AUUG Business 37	
AUUG Incorporated election procedures	37
AUUG MANAGEMENT COMMITTEE: Summary of Minutes of Meeting, 11th December 1994	39
Updated AUUG Regional Contacts, 1994 -1995	40
AUUG Inc. 1995 Annual Elections: CALL FOR PARTICIPATION	40
AUUG Inc. 1995 Annual Elections: CALL FOR NOMINATIONS	41
~/archive> !AUUGN 43	
AUUGN Volume 5, #1: Formatting C <i>Tim Long</i>	43
AARNET Mail service details 56	
Index to AUUGN Volume 15 60	
AUUG Institutional Members 67	
AUUG Membership applications & change notification 69	
Calendar of Events Inside back cover	

Contribution deadlines for AUUGN in 1995

- Vol 16, #2 (April): March 24
- Vol 16, #3 (June): May 19
- Vol 16, #4 (August): July 21
- 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@munnari.oz.au

Submission Guidelines

Submission guidelines for AUUGN contributions are regularly posted on the aus.org.auug news group. Alternately, e-mail to the above correspondence address, requesting a copy.

AUUGN Back-issues

A variety of back-issues of AUUGN are still available; for price and availability details, please contact the AUUG Secretariat, or write to:

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

Conference Proceedings

A limited number of copies of the Conference Proceedings from previous AUUG Conferences are still available, at \$50 each for members, and \$60 for non-members. Contact the AUUG Secretariat for details.

Mailing Lists

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

Disclaimer

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

Copyright Information

Copyright ©1995 AUUG Incorporated. All rights reserved. AUUGN is the journal of AUUG Incorporated, an organisation with the aim of promoting knowledge and understanding of Open Systems, including, but not restricted to, the UNIX[®] system, networking, graphics, user interfaces and programming and development environments, and related standards.

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

Trade marks

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

Editorial

Phil Anderson
 <auugn@munnari.oz.au>

Welcome to the first issue of AUUGN for 1995, and my first as well. The cover might've changed, and you'll find that a few things have been moved around, but all the features you've grown to love are still here.

At least, I assume you love them... fresh to the task of editing a publication such as this, I'd like to be sure that AUUGN is giving you what you want. If it is, great! ... but is there anything else you'd like to see here? If it's not bringing you satisfaction, what do you want from AUUGN? Drop me a line at the AUUG Inc. address, or send me some e-mail <auugn@munnari.oz.au> and let me know.

All of the contributors to AUUGN are members like you, who provide papers, articles and reviews out of the goodness of their heart. If you're one of our readers who hasn't contributed before, how about sending us something? Maybe you've something to say about the UNIX/Open Systems world, or some meaty technical titbit. If you've a piece of worldly experience from which others might benefit, or a paper you've written, why not share it with the rest of us? The Humble Editor doesn't bite... much.



At work, I'm a UNIX user. Although I think the whole box 'n' dice has a long way to go before it's really user friendly, I like many of the things that the operating system lets me do. I admire the Linus Torvalds and the Free Software Foundations of this world, for their altruism and solid ideals. But you know what really bites? At home, I'm a DOS/Windows user. Why? Because I can buy the applications I need, to do the jobs I do, at a halfway reasonable price. They're quite user-friendly, and I have a wide range of products from which to choose. I can also use my PC as an entertainment console for a reasonably low cost, with a huge choice.

UNIX may be a technically superior operating environment, but it requires its users to have either a pile of technical savvy, or a heap of money to get all of the user-friendly bells and whistles. That seems to be a recipe for trouble.

Who *will* the UNIX users of the next ten years be? Will they still be developers, students and researchers, or will the likes of a Plan 9 take hold amongst those groups? Can the proponents and marketeers of UNIX consolidate their foothold in the commercial arena? How many home PCs will be running Linux, or FreeBSD, or another no-cost incarnation instead of Windows9x or OS/2? Does UNIX have a future in the ever-expanding domestic market?

Does it have a future at all in its current incarnations? Keeping a foothold in the commercial world often has more to do with how many folks use your product, than it does with how powerful or technically elegant that product is. Look at the VHS versus Betamax video conflict; How many of us still use Beta because it's technically superior? Could the same happen to UNIX?

Will the Open Systems concept become so prevalent in software and hardware development that, as consumers, most of us won't know, or indeed care about the operating environment we're using? I'd like to hear your thoughts.



Thanks must go to my predecessor, Jagoda Crawford, for her help and support in getting under weigh, and for allowing me to bother her for the odd word of wisdom. Much appreciated.❖

AUUGN Submission Guidelines

Please submit all articles, comment and notices for publication in AUUGN in the following forms:

Text

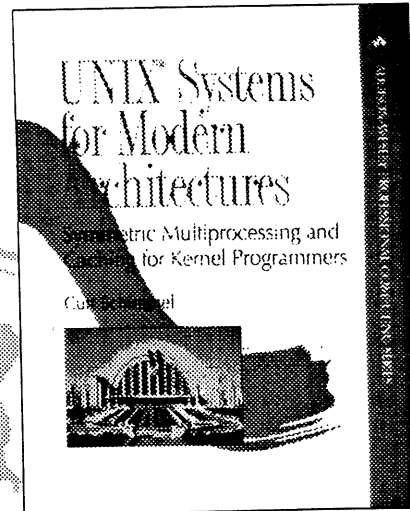
- ASCII format
- Text left-aligned, and unjustified
- Blank line between paragraphs
- If used, please put footnotes, notes and bibliography at the end of the file

Images

- Please provide images, diagrams and illustrations as separate files
- Make sure that your text contribution indicates where graphics should go in the contribution
- We can read/convert most graphics formats; make sure that if providing images as Postscript, that you use Encapsulated Postscript (EPS) format
- If the above is not possible, small hardcopy illustrations are acceptable

Where to send your submissions

- E-mail your submission to auugn@munnari.oz.au with AUUGN submission in the Subject line, or
- Post it on a disk (DOS formatted is preferable, though both MAC and UNIX disks are also accepted) to
 AUUGN Submissions
 PO Box 366
 Kensington NSW 2033



20% Discount to AUUG Members

UNIX Systems for Modern Architectures

Curt Schimmel

Written for the UNIX kernel developer, this book provides a complete yet comprehensible explanation of the operation of caches and symmetric multiprocessors, how they work together, and the issues operating systems must address in order to run on the machines that incorporate them.

After a review of UNIX kernel internals, Schimmel launches into a detailed description of cache memory systems, including several kinds of virtual and physical caches, as well as a chapter on efficient cache management. For each type of cache, the book covers the impact on the software and the operating system changes necessary for these systems. The next section details the operation of the tightly-coupled, shared memory, symmetric multiprocessor. It examines the problems these multiprocessors present to the operating system, such as race conditions, deadlocks, and the ordering of memory operations, and looks at how the UNIX kernel can be adapted to run on such systems. Finally, the book looks at the interaction between cache memory systems and multiprocessors and the new problems that this interaction presents to the kernel. Techniques for solving these problems are then explained.

CODE 63338-8 RRP \$59.95 AUUG Price \$47.95

NEW TITLES IN THE PROFESSIONAL COMPUTING SERIES:

- | | | |
|---|--------------|----------------------|
| Stevens, TCP/IP Illustrated, Volume 2: The Implementation | (Code 63354) | available Feb '95 |
| Lakos, Large Scale C++ Software Design | (Code 63362) | available May '95 |
| Pendergrast, Desktop Kornshell Graphical Programming | (Code 63375) | available August '95 |

YES: I would like to order:

QTY: _____ TITLE: _____

QTY: _____ TITLE: _____ QTY: _____ TITLE: _____

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

BANKCARD VISA MASTERCARD AMERICAN EXPRESS ID NO: _____

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

NAME: _____ SIGNATURE: _____

STREET ADDRESS: _____

STATE: _____ POST CODE: _____

Send your order to ➤

Addison-Wesley Publishing Company

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



The President's Report

Phil McCrae
 <pmc@syd.dit.csiro.au>

You have probably noticed that this issue of AUUGN has a different look to it. And for a good reason – AUUGN has a new editor! Phil Anderson has taken over the helm from Jagoda Crawford, who has spent the past three years in the job. We are deeply indebted to Jagoda for her tireless efforts over the past three years: we have had 6 quality issues per year, published on time. Quite often the job of an editor is a tedious one, as a lot of the work is routine – especially when you have to chase people the likes of me for articles! So, from all AUUG members, thanks Jagoda!

Despite his name, Phil is not an international cyclist - but I do believe he is a swimmer! He works for OzWare Developments in Melbourne, where he is publications manager, and chief technical writer. We welcome Phil to the position, and look forward to a new look AUUGN for 1995. AUUGN, of course, is only as valuable as the contributions the editor receives. So one new editor does not an AUUGN make! Phil will be looking for articles and snippets of information from the usual sources, so please support him as much as possible.



On other issues, the summer conferences are in full swing in most states, the winter conference, AUUG '95, is taking shape nicely, *and* AUUG has a web page! It's hosted on the Canberra Chapter machine. Try it on <http://www.auug.org.au>. Our thanks go to Lawrie Brown et al. in Canberra for their efforts.❖

Readers' Mail

Last issue's President's Report prompted this response from Associate Professor Brian Salter-Duke of the Northern Territory University's School of Chemistry and Earth Sciences ...

Scientific Computing - A forgotten discipline?

Phil McCrae's comment, "Very few computers are used for number crunching these days – possibly only supercomputers, and the odd spreadsheet calculation", is so far off reality that I am beginning to think there is a conspiracy to drive us poor scientific computing folk out of the computing world by pretending we do not exist. After all, the ACS has ignored us for years and now seems to think

computing is just a branch of business management. I thought AUUG would know better.

It is wrong in two ways. First, lots of computers are used for number crunching. Second, they are not only supercomputers. Most number crunching these days is done on UNIX workstations.

I have used the ANU Supercomputer and may use it again, but I do not use it now. Last year, three colleagues and I obtained a DEET Infrastructure grant that has purchased two DEC Alpha 3000/800s. Both are now flat out number crunching. I also have a small IBM RISC 6000 and access to a larger university one. I often drive these hard crunching numbers. My field is computational chemistry and it is a rapidly growing field. In the US the number of graduates employed in this area has been growing exponentially for 10 years. Most are in industry. Many workstation vendors have specifically targeted this group in their sales material. Computational chemistry does include some symbol manipulation and some CAD-like graphics, but it uses lots of CPU hours crunching numbers. Most of it is done on workstations. Some is done on fast PCs (you should have seen the fuss in our community about the Pentium floating point bug – it matters). The supercomputer is often the last resort for really big tasks. Computational chemistry is just one of the activities that is still happily crunching numbers. What about environmental modelling, weather forecasting, etc. All these activities are growing, not declining. The very complex modelling of all aspects of our complex world will show a vast expansion in the next few decades.

The move here to workstations from mainframes and supercomputers has implications for AUUG. People like me are now also system administrators and we need to know a lot more about the operating system than we did 20 or more years ago (I go back 35 years to Mercury Autocode!). The operating system is now almost always UNIX.

While the majority of PCs are text manipulators, I estimate a majority of workstations spend a significant part of their time number crunching. The people who run them need the support of AUUG. Just my \$0.02 worth.❖

Opinion:

The Operating System they use in Heaven...

Phil McCrae, CSIRO Division of Information Technology
<Phillip.McCrea@syd.dit.csiro.au>

I don't know quite how mind associations work. People have modelled the brain as a huge associative memory with billions of processors, variously interconnected such that the topography of the links between processors represent concepts, and the reduction in strength of these links represents loss of memory with the passage of time.

Those of you who know the Welsh comedian, Max Boyce, would be aware of his description of rugby as "the game they play in heaven!". No-one but the most ardent AFL fans would dare dispute this, of course... although the recent form of the Welsh rugby team may well have caused Max to change his tune – if not his religion!

Somehow the presence of both the Pope and Bill Gates in Sydney at the same time recently conjured up the phrase in my mind: "UNIX – the operating system they use in heaven"! Again, very few people—other than the uninformed and perhaps a few VMS die-hards who still retain that incredible ability to pronounce file names with '\$' characters in the middle of them—would dispute this.

All except for Mr. Gates, that is! I know for a fact that Bill Gates is not a VMS zealot, and no-one could possibly call him uninformed. So why does he insist in thrusting his own operating system on the world that, by and large, the IT community does not need nor want?

It set me off thinking about Microsoft in the wider context. We actually know why Microsoft is spurning the "Operating System they use in Heaven" in favour of their own home-cooked alternative: it's all to do with the bottom line. In the world of commodity purchasing, it is the best selling product (or technology) that eventually becomes the standard, purely by market forces. The organisation whose technology dominates the commodity market stands to gain healthy royalty revenues and so forth.

That Microsoft has cornered the desktop market cannot be denied, although IBM and Apple are still plugging away with their own desktop environments. Even this writer has succumbed and is penning these few thoughts in a Windows environment...

In the non-desktop arena—generally referred to as the midrange market—Microsoft has to date been a non-player, but they have designs on this area with NT –

under its various code names. This is the area where UNIX is firmly entrenched, and every single computer manufacturer now offers UNIX. It is a non-issue: UNIX will not be moved from its position of dominance of the midrange. I honestly believe that Microsoft have left their run too late.

Let's face it, the midrange is the database market. It is the 'server' side of client/server technology. Microsoft has reached a position of dominance at the desktop (i.e. client side) by owning both the operating environment *and* the desktop application software market – or at least a healthy proportion of it. Again, this writer will admit to being a user of Word – and quite a satisfied one at that!

Who are the dominant players in the server market? Certainly all the database vendors are there. In fact, if you ignored the database vendors, there really wouldn't be a server market at all! Databases such as Oracle, Sybase, Informix, Ingres and so forth are not tied to any particular operating environment – thanks to Open Systems (read: UNIX). This is the world where standards *do* matter: users want the freedom to be able to change their hardware if they wish. Changing from vendor A to vendor B is a relatively painless task if the implementation has been on UNIX. As an aside, changing an application from one relational database to another *is* a different matter: even though they are all SQL based, they all have enough attractive proprietary extensions to make portability a major problem.

Microsoft are not only at loggerheads with the rest of the IT industry in the server environment area – they are trying to establish their own proprietary standards on the market place in other areas as well. One such area is Objects. Everyone (even Microsoft) agrees that having a set of software 'objects' that can be used over and over again in different programs would be a good thing. Virtually the rest of the IT world have thrown their support behind the Object Management Group's definition of reusable 'objects', the Common Object Request Broker Architecture (CORBA). Our colleagues at Microsoft, however, have their own object description—OLE—which they are trying to make a standard purely by market forces. And before I get calls from Microsoft lawyers, let me point out that I am quoting from learned journals such as IEEE Computer, and Object Magazine...

We can all do our bit, of course, to level the playing field by sticking to both UNIX and the wider concepts of Open Systems that UNIX has spawned – even Microsoft could not hold out against TCP/IP, which after all is the standard communications software on UNIX.

Is UNIX the operating system they use in heaven? You bet!❖

Conferences and Announcements

AUUG SOUTH AUSTRALIA SUMMER CONFERENCE 1995

Friday March 24th, 1995
9:00 am to 4:30 pm

Location:

ETSA Auditorium, 41 The Parade, Norwood SA 5067

Theme:

The Internet - Legal, Technical and Security Issues

Speakers:

Will include:

Richard Prior, Bonnins, Legal issues

Simon Hackett, Internode, Security issues

Rates:

Status	Code	Fee
Full-time student	S	\$40
AUUG Member *	M	\$60
Non-member	N	\$80

* Members discounts are also available to ACS and ACADS members. Registration includes fees, lunch, morning and afternoon teas.

Payment:

Payment should be by cheque, crossed (not negotiable) and made payable to **AUUG Inc. - SA Chapter**.

Alternative payment arrangements can be made, if required, by contacting the conference treasurer (see contact details, below). Receipts will be available at the Conference.

Please fax the registration slip below or email the details described there to

adick@stnecss.telecom.com.au

Please forward payment to:

c/o Alastair Dick, Telecom Australia, Locked Bag 342, GPO Adelaide SA 5800

Ph: 08-230-7062 Fax: 08-410-0023

Email: adick@stnecss.telecom.com.au

Enquiries:

Kathryn Jones, Systems Services

Ph: 08-212-2800 Fax: 08-231-0321

Email: klj@syserv.com.au

AUUGWet 95 Northern Territory Chapter of AUUG

18-21 April, 1995

The NT Chapter of the Australian UNIX users group (AUUG) will be holding a Wet season conference (summer conference for southerners) during April 95. The theme for this conference is:

Security and the Internet

Presentations on any area of interest to the open systems community are sought.

AUUGWet has been running for four years now and provides a good opportunity to visit the Top End. We normally get about 70 people at the conference.

But why should you come to Darwin? Well:

To visit the Top End. Free food and drink. Access to the School of Information Technology kit:

- 1 x Piper Cherokee 6 (airplane).
- 1 x GPS.
- 1 x slightly soiled Toyota 4x4.

Free air tickets:

We hope to able to bring in a few speakers from outside Darwin. We would of course provide you with the air tickets. Free accommodation. We can probably find somewhere for you to stay in Darwin free of charge.

Contact:

Phil Maker pjm@cs.ntu.edu.au

Euan Pryde edp@it.ntu.edu.au (089-466546)

AUUG - AUSTRALIAN UNIX AND OPEN SYSTEMS USER GROUP
1995 SUMMER CONFERENCE SOUTH AUSTRALIA
Friday March 24th, 1995

REGISTRATION DETAILS

Organisation:

Fax Number:

Attendee Name:

Phone No.:

E-mail addr:

Code(S/M/N): Fee: \$

Member(Society):

TOTAL \$ _____

Call for Papers:

Australian UNIX and Open Systems Users Group Annual Conference, September 19-21, 1995 Sydney, Australia

"The Internet Means Bu\$iness"

Every 30 minutes another network is connected to the world wide Internet. What was a research experiment of the 80's is now part of every production level communications service inventory in the '90s. It is quickly becoming the only viable method of servicing hundreds and thousands of people in Australia and millions world wide, from corporate and government users to academics and researchers as well as the private individual. Despite its size, the Internet resists inertia and continues to grow, assuming dynamic new roles at each stage in its evolution.

AUUG '95 solicits papers on all aspects of the Internet phenomenon, in particular, its relevance to business use and applications combining UNIX and Open Systems technologies with the Internet's open communications services.

Events:

AUUG '95 will be a four day conference, commencing September 18, 1995. The first day will be devoted to tutorial presentations, followed by three days of papers, work-in-progress and product update sessions, and BOFs.

Tutorials:

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

Papers:

AUUG '95 provides dual Technical and Management tracks for the presentations. To share your innovative implementations, applications, and similar areas submit your abstract for the technical track. We are also interested in your experiences, case studies, strategic issues, and the like. If your topic better fits these areas submit your abstract for the Management track. This should not, of course, discourage papers which are appropriate for both audiences at once.

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

Prize for the Best Student Paper:

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

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

A great success in '94, these brief 15 minute sessions are designed to report on current work with fundamental aspects highlighted, and to allow the chance for vendors to 'advertise' products. Product specification sheets should be submitted with your abstract.

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

Are you interested in discussing particular problem areas, sharing arcana on favourite programs, using the internet, or other controversial topics? During the lunch hour and at the end of each presentation day, time slots for BOFs will be available. We distinguish two types of BOF; general interest and vendor sponsored. Please contact the Committee if you would like to organise a Birds-of-a-Feather Session. There may be some facilities charge to vendor sponsored events.

Speaker Incentives:

Presenters of papers are afforded free conference registration. Product Update, Work-in-Progress and panellist presenters get single day registration and can 'top up' to get full attendance.

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

Form of Submissions:

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

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

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

Acceptance:

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

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

Programme Committee:

Geoff Huston - AARNet (chair)
 Ian Hoyle - BHP Research
 Hugh Irvine - connect.com.au
 Phil McCrea - Consultant
 Frank Crawford - ANSTO
 Andy Linton - AARNet

Relevant Dates:

Abstract and outlines due: March 6, 1995
 Notifications to authors: April 3, 1995
 Final Papers due: August 4, 1995

Addresses:

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

AUUG '95 Programme
 PO Box 468
 Paddington, NSW 2021
 AUSTRALIA

e-mail: auug95.papers@munnari.oz.au

Phone: +61 2 959-3656

Fax: +61 2 923-1036

Tutorial abstracts to: auug95.tutorials@munnari.oz.au

Announcement and Call for Papers:

**SAGE_AU THIRD ANNUAL
 CONFERENCE AND GENERAL
 MEETING
 (SAGE-AU'95)**

Dates:

Wednesday July 12 to Friday July 14, 1995

Venue:

The University of Wollongong

Call for Papers

The System Administrators Guild of Australia (SAGE-AU) will be hosting its third annual conference

in conjunction with its 1995 Annual General Meeting. The theme of the conference is:

System Administration - Interacting with Users

System administration staff are often accused of being "user-unfriendly". Is this because they are always overworked and underpaid, or does the amount of time they spend interacting with computers deskill them in terms of human-human interaction? Does the nature of the profession tend to attract people who interact with machines better than people? What can system administration staff do to improve their human interaction skills?

This conference will focus on these issues.

SAGE-AU'95 is hereby calling for papers and tutorial presentations on any and all topics related to system administration, particularly those specifically relating to the theme of the conference, as described above.

Conference details

SAGE-AU'95 will be a three-day conference, running from Wednesday July 12, 1995 to Friday July 14, 1995.

The first day (Wednesday) will be dedicated to tutorials on tools and techniques to aid system administration.

The AGM will be held at the end of the second day.

All other times will be allocated to presentations or discussions.

A conference dinner will be held on Thursday evening.

The conference will feature a small trade show on the second and third days, focusing on system administration tools.

Papers

Time slots are available for 15 minute, 30 minute and 60 minute presentations. 5 minutes should be reserved for questions from the audience.

Fifteen minute time slots are less formal, and are to allow people to talk briefly about some topic of interest or problem without having to prepare a formal paper (Work in Progress).

People presenting a 30 or 60 minute talk will receive a 50% discount on conference registration fees.

People presenting a 15 minute talk will receive a 25% discount on conference registration fees.

If you wish to present a paper, send an abstract to the address below by the due date. Please indicate whether you are asking for a 15, 30 or 60 minute time slot.

Abstracts should be 100 - 200 words in length.

Papers should have a technical orientation, and should not contain advertising. People performing 30 minute and 60 minute presentations will be expected to provide a paper for inclusion in the conference

Conferences and Announcements

proceedings. The due date for papers to be included in the proceedings is given below.

Tutorials

Tutorial sessions will be either half-day or full-day in duration. People wishing to present tutorials should submit an abstract of the material they wish to present and an indication of whether they require a half-day or full-day time slot. Tutorials should be run in lecture format. Suggested topics include:

- Computer and Network security/Kerberos/Network Authentication
- PC/Apple/Mainframe Interoperability
- NFS/Automount/AMD Configuration and Operation

Tutorial presenters will be paid \$400 for a half-day tutorial and \$800 for a full-day tutorial and will receive free conference registration. SAGE-AU will reimburse tutorial presenters for reasonable costs of handout materials.

As with papers, tutorials should have a technical orientation, and should not contain advertising.

Exhibition/Trade Show

On the second and third days of the conference (Thursday and Friday) SAGE-AU'95 will host a small, technically-oriented trade show focusing on systems administration tools.

If you or your company are interested in participating in the trade show, please contact the organisers for details.

Registration

The registration fee for SAGE-AU'95 is \$120. Non-members may register for \$170. Non-members who register for SAGE-AU'95 and successfully apply for membership of SAGE-AU will have their first year's membership fee waived.

The registration fee includes one ticket to the conference dinner. Additional tickets to the conference dinner may be purchased separately.

Conference registration forms will be available in February 1995.

The registration fee for tutorials is \$100 (half-day) or \$200 (full-day).

Registration forms for tutorials will be available approximately six weeks before the conference date.

Travel

To encourage interstate attendees, SAGE-AU will be offering travel subsidies for interstate travellers. The travel rebate will be \$120 for attendees from Western Australia, South Australia, Tasmania and the Northern Territory, and \$60 for attendees from Victoria and Queensland.

Deadlines

Applications to present tutorials and papers must reach the organisers by April 28, 1995.

To be included in the conference proceedings, papers must reach the organisers by June 23, 1995. If submitted electronically, the preferred format is Postscript.

Addresses

Send all inquiries regarding the conference, as well as paper and tutorial abstracts to:

Peter Gray
Department of Computer Science
The University of Wollongong
Wollongong NSW 2522
Australia

e-mail: pdg@cs.uow.edu.au

Request for general information about SAGE-AU and membership applications should be addressed to:

Frank Crawford
c/- Computing Centre
ANSTO
Private Mailbag 1
Menai NSW 2234
Australia

e-mail: frank@atom.ansto.gov.au

THE USENIX ASSOCIATION 1995 CALENDAR OF SYMPOSIA AND CONFERENCES

USENIX/SAGE TUTORIALS ON INTERNET SECURITY, UNIX POWER TOOLS, FIREWALLS, AND THE LAW AND COMPUTERS

March 12-13, 1995

Sponsored by the USENIX Association, SAGE (the System Administrators Guild), and UniForum
UniForum Exposition & Conference, Dallas, Texas

2ND USENIX SYMPOSIUM ON MOBILE AND LOCATION-INDEPENDENT COMPUTING

April 10-11, 1995

Ann Arbor, Michigan Program Chair: Jim Rees, CITI,
University of Michigan

4TH SYSTEM ADMINISTRATION, NETWORKING, AND SECURITY SYMPOSIUM (SANS IV)

April 24-29, 1995

Sponsored by The Open Systems Conference Board in cooperation with USENIX Association's Special Technical Group, SAGE (System Administrators Guild) and The Washington Area UNIX User's Group

Washington, D.C. Program Chair: Rob Kolstad, BSDI, Inc.

5TH USENIX UNIX SECURITY SYMPOSIUM

June 5-7, 1995

In cooperation with The Computer Emergency Response Team (CERT), IFIP WG 11.4, and UniForum Salt Lake City, Utah Program Chair: Fred Avolio, Trusted Information Systems, Inc.

USENIX CONFERENCE ON OBJECT-ORIENTED TECHNOLOGIES (COOTS)

June 26-29, 1995

Monterey, California Program Chair: Vince Russo, Purdue University Tutorial Program Chair: Doug Lea, SUNY Oswego

TCL/TK WORKSHOP 95

July 6-8, 1995

Sponsored by Unisys Inc. and the USENIX Association Toronto, Ontario, Canada

9TH USENIX SYSTEMS ADMINISTRATION CONFERENCE (LISA IX)

September 18-22, 1995

Sponsored by SAGE, the System Administrators Guild Monterey, California Program Chairs: Tina Darmohray, Lawrence Livermore National Laboratory, and Paul Evans, Synopses, Inc.

For more information, please contact

USENIX Conference Office
22672 Lambert St., Suite 613
Lake Forest, CA USA 92630
Tel: +1 714 588 8649
FAX +1 714 588 9706
E-mail: conference@usenix.org

Announcement & Call for Participation:

9th USENIX SYSTEMS ADMINISTRATION CONFERENCE (LISA IX)

September 18-22, 1995

Marriott Hotel, Monterey, California

Co-sponsored by USENIX, the UNIX and Advanced Computing Systems Professional and Technical Association, and SAGE, the System Administrators Guild

Important Dates

REFEREED PAPER SUBMISSIONS

- Extended abstracts due: **May 1, 1995**
- Notification to authors: **June 5, 1995**
- Final papers due: **August 1, 1995**

- Registration materials available: **July, 1995**

The USENIX Systems Administration (LISA) Conference is widely recognized as the leading technical conference for system administrators. Historically, LISA stood for "Large Installation Systems Administration," back in the days when having a large installation meant having over 100 users, over 100 systems, or over one gigabyte of disk storage. Today, the scope of the LISA conference includes topics of interest to system administrators from sites of all sizes and kinds. What the conference attendees have in common is an interest in solving problems that cannot be dealt with simply by scaling up well-understood solutions appropriate to a single machine or a small number of workstations on a LAN.

The theme for this year's conference is "New Challenges," which includes such emerging issues as integration of non-UNIX and proprietary systems and networking technologies, distributed information services, network voice and video teleconferencing, and managing very complex networks. We are particularly interested in technical papers that reflect hands-on experience, describe fully implemented and freely distributable solutions, and advance the state of the art of system administration as an engineering discipline.

Tutorial Program

Monday and Tuesday, September 18-19, 1995

The two-day tutorial program at the conference offers up to five tracks of full- and half-day tutorials. Tutorials offer expert instruction in areas of interest to system administrators of all levels, from novice through senior. Topics are expected to include networking, advanced system administration tools, Solaris and BSD administration, Perl programming, firewalls, NIS, DNS, Sendmail, and more.

To provide the best possible tutorial offerings, USENIX continually solicits proposals for new tutorials. If you are interested in presenting a tutorial at this or other USENIX conferences, please contact the tutorial coordinator:

Daniel V. Klein
Tel: +1 412 421 0285
FAX: +1 412 421 2332
E-mail: dvk@usenix.org

Technical Sessions

Wednesday through Friday, September 20-22, 1995 The three days of technical sessions consist of two parallel tracks. The first track is dedicated to presentations of refereed technical papers. The second track is intended to accommodate invited talks, panels and Works-in-Progress (WIP) sessions.

Conference Topics

Papers addressing the following topics are particularly timely; papers addressing other technical areas of general interest are equally welcome:

Dealing with differences in UNIX implementations - migration and interoperability among BSD, SVR4, OSF and others □ Integration of UNIX-based with non-UNIX-based and proprietary systems and networking technologies (Mac, NT and DOS PCs) □ Application of emerging technologies (Mbone, Mosaic) to system administration □ Administration and security of distributed information services (WAIS, gopher, WWW) and network voice and video teleconferencing (Mbone) □ Experience supporting mobile and location-independent computing □ Experience with large (1000+ machine) networks, especially networks of SVR4-based systems □ Real-world experience with implementations of proposed system administration standards □ Unusual applications of commercial system administration software packages □ Application of operational planning techniques to system administration including measurements and metrics, continuous process improvement, automation, and increasing productivity □ File migration, archival storage and backup systems in extremely large environments □ Innovative tools and techniques that have worked for you □ Managing high-demand and high-availability environments □ Migrating to new hardware and software technologies □ Administration of remote sites that have no technical experts Supporting MIS organizations on UNIX □ Real-world experiences with emerging procedural/ethical issues, e.g., developing site policies, tracking abusers, and implementing solutions to security problems □ Networking non-traditional sites (libraries, museums, K-12)

Refereed Paper Submissions

An extended abstract is required for the paper selection process. Full papers are not acceptable at this stage; if you send a full paper, you must also include an extended abstract. "Extended" means 2-5 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 or measurement tool.

Submissions will be judged on the quality of the written submission, and whether or not the work advances the state of the art of system administration. For more detailed author instructions and a sample extended abstract, send email to lisa9authors@usenix.org, or call USENIX at +1 510 528 8649.

Note that the USENIX organization, 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 unread. All submissions are held in the highest confidence prior to publication in the conference proceedings, both as a matter of policy and as protected by the U.S. Copyright Act of 1976.

Authors of an accepted paper must provide a final paper for publication in the conference proceedings. At least one author of each accepted paper presents the paper at the conference. Final papers are limited to 20 pages, including diagrams, figures and appendixes, and must be in troff, ASCII, or LaTeX format. We will supply you with instructions. Papers should include a brief description of the site, where appropriate.

Conference proceedings, containing all refereed papers and materials from the invited talks, will be distributed to attendees and will also be available from the USENIX following the conference.

Where To Send Submissions

Please submit extended abstracts for the refereed paper track by TWO of the following methods:

- E-mail to: lisa9papers@usenix.org
- FAX to: +1 510 548 5738
- Mail to:
LISA 9 Conference
USENIX Association
2560 Ninth Street, Suite 215,
Berkeley, CA USA 94710

To discuss potential submissions, and for inquiries regarding the content of the conference program, contact the program co-chairs at lisa9chair@usenix.org or at:

Tina M. Darmohray
Lawrence Livermore National Laboratory
PO Box 808 L-510
Livermore, CA USA 94550
Tel: +1 510 423 5999
FAX: +1 510 422 7869
E-mail: tmd@usenix.org

Paul Evans
Synopsys, Inc.
700 East Middlefield Road
Mountain View, CA USA 94043
Tel: +1 415 694 1855
FAX: +1 415 965 8637
E-mail: ple@usenix.org

Invited Talk Track

If you have a topic of general interest to system administrators, but that is not suited for a traditional technical paper submission, please submit a proposal for a second track presentation to the invited talk (IT) coordinators:

Laura de Leon, Hewlett-Packard
 Tel: +1 415 857 5605
 FAX: +1 415 857 5686
 E-mail: deleon@hpl.hp.com

Peg Schafer, BBN
 Tel: +1 617 873-2626
 FAX: +1 617 873 4265
 E-mail: peg@bbn.com

Program Committee

Program Co-chair: Tina Darmohray, Lawrence Livermore National Laboratory

Program Co-chair: Paul Evans, Synopsys, Inc.

Paul Anderson, University of Edinburgh
 Kim Carney, Massachusetts Institute of Technology
 Rob Kolstad, Berkeley Software Design, Inc.
 Bryan McDonald, SRI International
 Marcus Ranum, Trusted Information Systems, Inc.
 John Schimmel, Silicon Graphics, Inc.

Vendor Display

Wednesday, September 20, 1995

Well-informed vendor representatives will demonstrate products and services at the informal table-top display. If your company would like to participate, please contact:

Zanna Knight
 Tel: +1 510 528 8649
 FAX: +1 510 548 5738
 E-mail: display@usenix.org

Birds-of-a-feather Sessions

Birds-of-a-Feather sessions (BOFs) are very informal gatherings of attendees interested in a particular topic. BOFs are held Tuesday, Wednesday, and Thursday evenings of the conference. BOFs may be scheduled in advance by telephoning the USENIX Conference Office at +1 714 588 8649 or via e-mail to conference@usenix.org. They may also be scheduled at the conference.

For Registration Information

All details of the conference program, conference registration fees and forms, and hotel discount and reservation information will be available in July, 1995. If you wish to receive registration materials, please contact:

USENIX Conference Office
 22672 Lambert Street, Suite 613
 Lake Forest, CA USA 92630
 Tel: +1 714 588 8649
 FAX: +1 714 588 9706
 E-mail: conference@usenix.org

Announcement and Preliminary Call for Papers:

5th USENIX UNIX Security Symposium

June 5-7, 1995

Salt Lake City Marriott Hotel,
 Salt Lake City, Utah

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

In cooperation with: The Computer Emergency Response Team (CERT), IFIP WG 11.4, and UniForum (pending)

Important Dates

REFEREED PAPER SUBMISSIONS

- Extended abstracts due: Feb. 13, 1995
- Program Committee decisions made: March 8, 1995
- Camera-ready final papers due: May 1, 1995
- Registration Materials Available: March 1995

Program Committee

Program Chair: Fred Avolio, Trusted Information Systems, Inc.

Steve Bellovin, AT&T Bell Laboratories
 Bill Cheswick, AT&T Bell Laboratories
 Ed DeHart, CERT
 Ed Gould, Digital Equipment Corporation
 Marcus Ranum, Trusted Information Systems, Inc.
 Jeff Schiller, MIT
 Gene Spafford, COAST Laboratory, Purdue University

Overview

The goal of this symposium is to bring together security practitioners, researchers, system administrators, systems programmers, and others with an interest in computer security as it relates to networks and the UNIX operating system.

This will be a 3 day, single-track symposium. The symposium will consist of tutorials, refereed and invited technical presentations, and panel sessions. The first day will be devoted to tutorial presentations. Two days of technical sessions will follow the tutorials.

Tutorials [June 5]

This one-day tutorial program is designed to address the needs of both technical and management attendees. The tutorials will supply overviews of various security mechanisms and policies. Each will provide specifics to the system and site administrator for implementing numerous local and network security precautions, firewalls, and monitoring systems.

Keynote and Technical Sessions [June 6-7]

The keynote address by Stephen T. Walker, Founder and President of Trusted Information Systems, will begin the technical sessions program. Mr. Walker will speak on information security and privacy in computing. Mr. Walker is an electronics engineer and computer systems analyst with over 25 years of experience in system design and program management; particularly extensive is his experience with the design and implementation of large scale computer networks and information systems. He is recognized for his pioneering work on the DoD Computer Security Initiative, the establishment of the National Computer Security Center, and the formation of the Defense Data Network. He is a member of the Computer System Security and Privacy Advisory Board, established by the Computer Security Act of 1987.

The technical sessions program, in addition to presentations of refereed papers, will include invited talks, and possibly panel sessions. There will also be two evenings available for Birds-of-a-Feather sessions (BOFs) and Works-in-Progress Reports (WiPs). The program committee invites you to submit proposals, ideas, or suggestions for these presentations; your suggestions may be submitted to the program chair via email to: security@usenix.org or by post to the address given below.

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

Symposium Topics

Presentations are being solicited in areas including but not limited to:

User/system authentication File system security
Network security Security and system management
 Security-enhanced versions of the UNIX operating system Security tools Security incident investigation and response Computer misuse and anomaly detection Security in heterogeneous environments Configuration management to support security Security-related testing methods
Case studies

Refereed Paper Submissions

Submissions must be received by Feb. 13, 1995. Full papers should be 10 to 15 pages. Instead of a full paper, authors may submit an extended abstract which discusses key ideas. Extended abstracts should be 5-7 pages long (about 2500-3500 words), not counting references and figures. The body of the extended abstract should be in complete paragraphs. The object

of an extended abstract is to convince the reviewers that a good paper and presentation will result.

All submissions will be judged on originality, relevance, and correctness. Each accepted submission will be assigned a member of the program committee to act as its shepherd through the preparation of the final paper. The assigned member will act as a conduit for feedback from the committee to the authors. Camera-ready final papers are due May 1, 1995.

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

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

The UNIX Security Symposium, like most conferences and journals, requires that papers not be submitted simultaneously to another conference or publication and that submitted papers not be previously or subsequently published elsewhere. Papers accompanied by "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, both as a matter of policy and in accord with the U.S. Copyright Act of 1976.

Where to Submit

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

Preferred Method: email (Postscript or ASCII) to: securitypapers@usenix.org

Alternate Methods: Postal delivery to:

Fred Avolio
Trusted Information Systems
3060 Washington Road
Glenwood, MD 21738

Tel: +1 301 854 6889
Fax to: +1 301 854 5363

Registration Materials

Materials containing all details of the technical and tutorial programs, registration fees and forms, and hotel information will be available beginning in March 1995. If you wish to receive the registration materials, please contact USENIX at:

USENIX Conference Office
22672 Lambert Street, Suite 613
Lake Forest, CA USA 92630
Tel: +1 714 588 8649
Fax: +1 714 588 9706
email: conference@usenix.org

Combine Business with Pleasure

UniForum NZ '95

The Conference

17-20 May 1995
Masterton, New Zealand

Featuring:

Technology Convergence	Hypermedia
The Internet	Security
Open Networking	User Tools

Registration forms available mid-March 1995.
Contact: Julie Jones, UniForum New Zealand
P.O. Box 27-149 email: jj@integral.co.nz
Mt Roskill, Auckland Ph: 64-25-958-245
New Zealand Fax: 64-9-629-2015



Moving to OPEN Systems?

Softway is Australia's largest open systems software house.
We understand the needs of the open systems marketplace,
and have extensive expertise in the following areas:

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

79 Myrtle Street	Phone: (02) 698 2322
Chippendale	Fax: (02) 699 9174
NSW 2008	Internet: enquiries@sw.oz.au



Specialists in:

- UNIX Personal Productivity software
- UNIX System Management software

- Distributed computing
- Object-oriented design & programming

**Master distributors for
the EASE range of
UNIX administration
tools.**

Suite 1, 214 Bay St. Brighton 3186
Victoria, Australia
Tel: (03) 596 2166. Fax: (03) 596 1134
E-mail: info@ozware.com.au



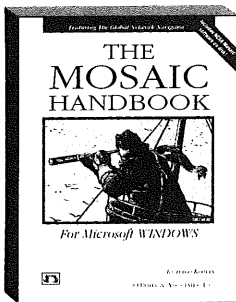
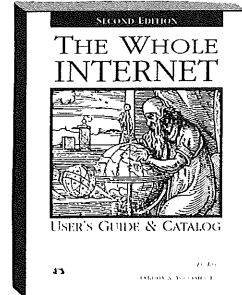
The Art of Information

New Books

THE WHOLE INTERNET USER'S GUIDE AND CATALOGUE 2nd Edition

Author: Ed Krol Price: \$49.95 ISBN: 1-56592-063-5

The best book about the Internet just got better! This is the second edition of O'Reilly's bestselling introduction to the Internet, a resource of almost unimaginable wealth.



THE MOSAIC HANDBOOK for MICROSOFT WINDOWS (Bk/Dsk)

Author: Richard Koman Price: \$59.95 ISBN: 1-56592-094-5

Using Mosaic, a user can move from document to document, viewing text, graphics, video, audio - without having to worry about where the information is located.

from O'Reilly,

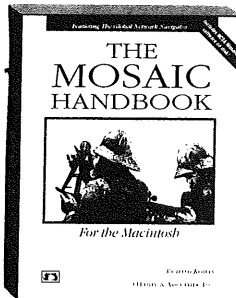
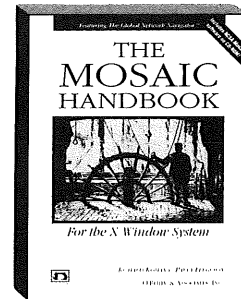
the Leading



THE MOSAIC HANDBOOK for the X WINDOW SYSTEM (Bk/CD)

This book comes with the public domain version of NCSA Mosaic on CD. The disk also contains the Mosaic Handbook Home Page, with links to all of the servers mentioned in the book.

Author: Koman & Ferguson Price: \$59.95 ISBN: 1-56592-095-3



Internet Publisher

THE MOSAIC HANDBOOK FOR THE MACINTOSH (Bk/Dsk)

Comes with Enhanced NCSA Mosaic on disk, a commercial version of NCSA Mosaic that improves upon what's available on the Internet, plus Mosaic Handbook Home Page, with links to all of the servers mentioned in the book.

Author: Richard Koman Price: \$59.95 ISBN: 1-56592-096-1

AVAILABLE FROM AUSTRALIA'S LEADING BOOKSELLERS

or call WoodsLane for the stockist nearest you

A.C.N. 003 667 549

NSW: Dymocks George St Sydney 2240411
 Dymocks Pocket & Technical
 Wynard Sydney 2240444
 Hotline Chatswood Sydney 4152265
 Dymocks North Sydney 9566771

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

QLD: Dymocks Edward St Brisbane (07) 220 0152

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

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

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

ARIEL2

WoodsLane Pty Ltd Phone: (02) 979 5944 Fax: (02) 9997 3348 EMail: woods@tmx.mhs.oz.au

Paper:

Transparency and Performance Issues in Sun RPC

David Purdue
<davidp@knowledge.com.au>

The Knowledge Group
Level 4, 153 Walker Street
North Sydney, NSW, 2060

Abstract

Sun RPC is an important implementation of a Remote Procedure Call mechanism, because it is included in almost all UNIX versions. This paper uses the construction of a sample Sun RPC program to examine issues of transparency and performance in Sun RPC.

1. Introduction

Sun RPC^(SUNM90) is Sun Microsystems Computer Corporation's implementation of a Remote Procedure Call mechanism. It is an important example of RPC because it is used on almost all UNIX systems,^(BLAC92, p. 281) the source code is available,^(COMP94) it is used to implement Sun's Network File System (NFS), which is itself a widely used distributed file system, and it has been recommended as an Internet standard.^(SUNM88, SUNM87)

In this paper I shall provide an assessment of Sun RPC in terms of how well it provides a transparent remote procedure call mechanism, and in how much use of Sun RPC affects performance as opposed to using lower level network programming primitives.

1.1. An Example

In this paper I will be illustrating the points I make with an example. The example service is a function that provides an arbitrary number of random numbers. It is conceivable that this may be a desired distributed service, as mathematical calculation or simulation programs may need a source of truly random numbers, and there may be a machine on the network that is attached to an atomic source, or some other source of truly random numbers. It is a useful example for this project, because it allows me to construct messages of different sizes, for use in evaluating performance.

Having written a random number generating function and a program that uses it, I altered the program so

that it would access the random number function over the network. Two such versions of the server and client were produced, one that uses Sun RPC, and another that uses the BSD socket interface to TCP.^(STEV90, Ch.6) Thus a comparison can be made in terms of transparency in the programmer's interface (going from local procedure call to RPC), and ease of programming (RPC versus socket programming).

The source code for these examples can be obtained by anonymous ftp from <ftp.auug.org.au>. At the time of writing I do not know which directory.

2. An Overview of Sun RPC

The Sun RPC package consists of four components:

- RPC and XDR protocols

The protocols used for client/server communication, and a language for describing the client/server interface

- **rpcgen**

A program that translates the description of the client/server interface into server and client stub routines

- a runtime library to make it all work
- the portmapper

A Sun RPC program that handles some transparency issues

These elements are examined in more detail below.

2.1. RPC and XDR protocols

The XDR protocol^(SUNM87) is a mechanism for encoding arbitrary data structures into a byte stream in a machine independent manner. The protocol also defines a language used to describe data structures that are to be sent with the XDR protocol.

At its simplest, XDR described issues such as byte ordering for integers¹ and the format for floating point numbers.² Character strings are sent with an ASCII encoding.³ The standard goes on to describe how arrays, variable length arrays, structures and unions can be encoded. Although there is no direct support for pointers (as noted in TANE92, p. 425, modern pointers only really make sense in the address space of the caller) XDR does support encoding for the recursive data structures they are often used to construct, such as linked lists or trees. This uses the "Optional-data" construct, and the syntax for describing it looks like the C syntax for pointers. This construct, however, does not support data structures of arbitrary complexity, for example it is not trivial to encode doubly linked lists.

1. Big-endian, which happens to be the same as most Sun machines.

2. IEEE Standard 74-1985, another coincidence?

3. You get the message. :-)

XDR is an implicit encoding, that is, the encoded byte stream contains no explicit data typing information. In the context of remote procedure calls this makes sense, as the lack of type information reduces the overhead of the protocol, and the procedure caller and the procedure being called should agree on the number and type of arguments, so that there is no ambiguity in how to the byte stream should be interpreted.

RPC (SUNM88) extends the XDR protocol and language by adding messages that represent procedure calls and replies. Individual procedures are identified by three numbers: the program number, used to identify a service; the version number, so that changes to the interface for a particular service can be recognised; and a procedure number identifying the required procedure in that service.

The RPC language allows a programmer to define the interface to the service offered by assigning program, version and procedure numbers and describing the arguments and results for each procedure.

The RPC protocol provides messages for procedure calls and server replies, and includes the option of an authentication mechanism. There are four authentication mechanisms defined in the protocol standard (Null, UNIX, Short and DES), but the protocol is open ended so other mechanisms can be defined.

RPC can be implemented over any transport layer protocol, but the lower protocol may affect the semantics of RPC, for example by imposing a maximum message size.

In my example, the RPC description of the random number service is shown in the file random.x.

2.2. rpcgen

The rpcgen program takes the RPC description of a service interface and produces four files: a client stub, a server stub, a file of functions for converting data structures to and from their XDR format, and a header file that describes the functions and data types described.

It is instructional for the programmer to examine the generated header file, as the data structures that rpcgen generates from the XDR description may not be exactly what the programmer expected. For example, look at random.h.

2.3. Runtime Library

As many operations in implementing a remote procedure call are identical from procedure to procedure, and as the stubs generated by rpcgen do not provide full transparency, there are a number of library routines used to support access by user programs to the RPC protocol. The programmer can use these, for example, to control which protocol RPC

will be layered over, or adjust the timeout used by the call, or get more detail on an error condition.

2.4. The Portmapper

The RPC protocol describes a message format, but leaves the delivery of these messages to whatever lower level protocol RPC is layered on. In the case of the TCP/IP suite of protocols, an RPC server must listen on a UDP or TCP port. The information the caller provides - namely program, version and procedure numbers - is not sufficient to locate the server. This is listed as a virtue in the protocol specification, (SUNM88,Sec. 5) as it allows independence from the binding software.

A standard piece of binding software supplied with Sun RPC is the portmapper. The portmapper is itself an RPC service (program number 100000, version 2) that listens on a well known port (port 111 for TCP and UDP). When RPC server programs start, they register their existence with the portmapper. Client programs tell the portmapper what program number, version and protocol (TCP or UDP) they want, and the portmapper replies with the port number that the server is listening on. The client can then use the protocol specified to contact that port.

3. Transparency Issues

In an ideal world, we should simply be able to replace our local procedure call with a client stub on the client end and a server stub on the server end, as shown in figure 1. In practise this is not possible, and in this section I examine some of the reasons for this. Many of the issues I examine here are raised in references. (TANE92, STEV90, ARCH89)

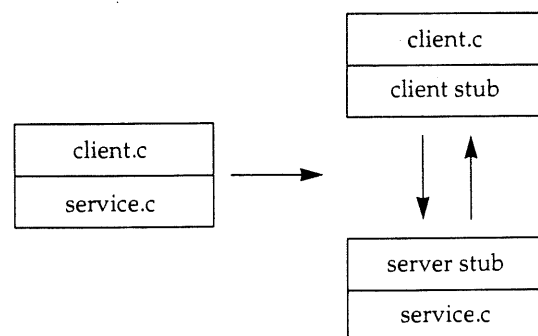


Figure 1 - Ideal local to distributed progression.

On the server side, it was sufficient to wrap the service procedure with another function to interface it the server stub generated by rpcgen. This wrapper is shown in `rpc_server_wrap.c`. On the client side, however, some more significant changes were required because RPC needs some initialisation before the service can be called. These changes are

highlighted with “`#ifdef RPC_CLIENT`” in `rpc_client.c`.

3.1. Service Numbering

When programmers specify the RPC interface, they assign a service number to the service they create. Thus it is the programmer’s responsibility to ensure that the number he or she uses does not clash with numbers being used by any other service on that machine (or, potentially, any other machine in the network).

It should really be the system’s responsibility to ensure unique numbers are assigned, if numbers are used at all. In the ANSA system, ^(ARCH89) for example, a service registers its properties with a trader, and clients look for servers with the required properties.

3.2. Transport Protocol

In theory, Sun RPC could be layered over any transport protocol; but in practice the run time libraries supplied only support TCP and UDP. A particular protocol must be chosen by the programmer, and the choice of protocol can affect the semantics of the call.

If UDP is chosen, then total size of arguments and the total size of results must each be less than the maximum UDP packet size, which in most implementations is 8192 bytes. TCP has no size restriction on parameters or results.

The RPC library supports a broadcast procedure call, so any server on the local network can reply, but only for RPC over UDP.

3.3. Parameter Passing

Sun RPC supports the passing of only a single parameter and the return of only a single result, though these parameters and results could be structures containing several data items. The only parameter passing model supported is call by value.

Of course, remote procedures cannot have side effects - they cannot access global variables.

3.4. Data Representation

The XDR protocol and XDR functions generated by `rpcgen` handle any differences in data representation from machine to machine. The client and server stubs automatically use these routines to encode and decode parameters and results. But, as noted in section 2.1, there are limits on the data structures that can be represented in XDR.

Note that the representation chosen in XDR mainly suits Sun machines, so that XDR is inefficient when used to transmit data between two machines that, for example, both use little-endian integers.

3.5. Binding and Location Transparency

In the local case, the caller knows which procedure it is using as they were bound together at compile time.⁴

In a remote procedure call, the caller must find the process providing the service it desires.

In Sun RPC this is accomplished by calling the portmapper service. The RPC library contains functions that will do a lookup on the portmapper when the client is initialised, so the client can then send its RPC requests to the appropriate TCP or UDP port.

Note, however, that the caller must specify which machine it wants to connect to find the service. If it is not known in advance which machine the service will be on, then the programmer must include code to send a broadcast RPC to the portmapper on all machines.⁵

3.6. Replication Transparency

A system designer may choose to have several replicated server instances for reasons of performance or reliability. The Sun RPC protocol provides no direct support for this. The system designer will have to ensure clients broadcast when looking for a service (see last section), and will have to come up with a way for the replicated servers to stay in step with each other.

3.7. Failure Handling

In the local case, the client need only examine the return code to determine the success of the operation. In the RPC case, as can be seen in `rpc_client.c`, the return status of the RPC library function must be examined to determine if a return code was even received from the server. Thus a whole extra layer of error checking and response must be added in on the client side.

If TCP is used as the transport protocol, then the client will get an error return if the server breaks its connection (crashes). The client has no way of knowing if the crash occurred before or after the request was executed. The TCP protocol handles timeouts and retransmissions.

If the UDP protocol is used, then the client stub automatically provides timeout and retransmission (the client programmer can set the value of the timeout). The RPC protocol includes a transaction identifier, and the server stub can be instructed to remember responses sent and to send a cached response again if it receives a request it has seen

4. Shared libraries are the exception to this. The functions in the shared library are bound at run time. However, you can trust the contents of a shared library as much as you trust the binary you are running, as they are just as easy to replace. You may not have control over the remote server that your RPC’s bind to.

5. Although the portmapper does have a procedure to automatically handle broadcast RPC (it looks up the server the RPC is destined for and sends it to that server itself if the server is registered, and drops the request if it is not) this only works for RPC over UDP. In the general case, the programmer must do a broadcast RPC to find which machine can provide the service, then initialise the client to talk to that machine.

before. This implements an at-most-once semantics in place of the default at-least-once semantics.

The SunRPC implementation is not resilient to failures of the portmapper. If the portmapper fails, new servers will not be able to register themselves, and clients will not be able to find running servers, so all RPCs will time out unsatisfied. In practice, even if the portmapper was restarted, all running servers would have to be restarted so they could register themselves with the new portmapper, so the only way to recover properly is to reboot the system.

3.8. Security

In the local case the caller of a function can be sure that it is calling an authorised provider of the service, and the procedure can be sure it is called by an authorised user, because they are linked together at compile time. With a remote call, neither party can be sure.⁶

To assure clients and server that they are talking to authorised servers and clients, Sun RPC includes an authentication mechanism. The client sends its credentials and a verifier to the server with its RPC call, the server returns its own verifier with the results. The standard authentication methods provided by the library are Null, UNIX, Short⁷ and DES, but it is easy to add new methods.

Using the authentication mechanisms is not transparent; it requires some extra programming on the client and server sides.

In addition, the portmapper introduces security risks as it can be used, in certain circumstances, to spoof the origin of a request. (CHES94, p. 35) It is recommended that firewalls should block packets sent to the portmapper.

4. Performance

Each of the client/server combinations - the local case, the Sun RPC version and a version that uses TCP sockets - were run. The clients were all run on a Sun 2 workstation, the servers on a Sun IPX, and the two were connected with a thin wire Ethernet.

The tests were run at night, so both machines and the network had no other significant load. This helps to reduce the variance between runs. Each run of the program requested 1,048,576 random numbers in total, meaning 4 Mbytes of data was transferred between the server and the client. Each program was run ten times, with requests for blocks of 32, 128, 512, 1024 and 4096 bytes of random numbers to be sent, and the execution time for each of these was averaged. The results of this experiment appear in table 1 below. By noting that in

6. In fact, many classic attacks on networks of Sun systems relied on providing a false RPC service.

7. Short is a shorthand version of UNIX, used as short cut when the client has already identified itself using UNIX authentication.

all cases 4 megabytes of useful data were transferred, we can work out effective transfer rates in kilobytes per second, these are shown in table 2.

Block size	local	tcp	rpc
32	118.12	296.41	468.52
128	32.38	82.92	128.23
512	10.36	25.87	41.89
1024	6.51	16.39	27.59
4096	4.05	9.21	17.91

Table 1 - Run times for different procedure link methods

Block size	local	tcp	rpc
32	34.7	13.8	8.7
128	126.5	49.4	31.9
512	395.4	158.3	97.8
1024	629.2	249.9	148.5
4096	1011.4	444.7	228.7

Table 2 - Transfer rates for different procedure link methods

These results are illustrated graphically in figure 2. Figure 3 compares the transfer rates for tcp and Sun RPC.

As can be seen from figure 3, using Sun RPC gives approximately a 50% penalty in data transfer rates. This runs contrary to expectation, as the RPC and XDR protocols should not introduce that much extra traffic, and there should not be excessive amounts of processing on either side to convert formats, so I would expect the transfer rates for TCP and RPC to converge for large message sizes.

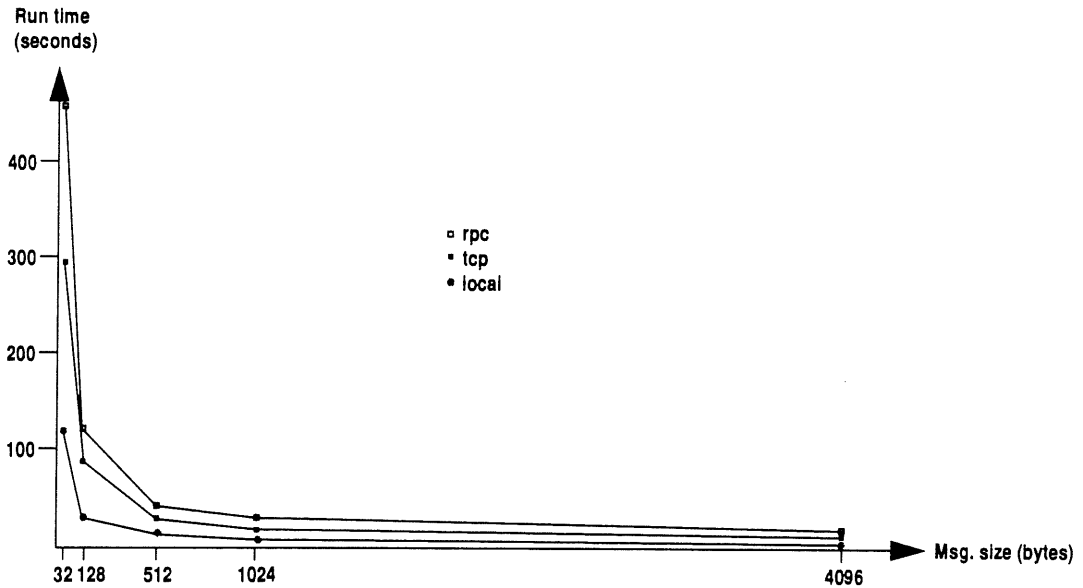
An examination of the RPC implementation is required to explain this, and this goes beyond the time I can devote to this paper. However, profiling suggests that the reason for the extra time taken by the RPC version is due to the number of calls made on XDR routines to encode and decode the integers sent.

5. Conclusions

The Sun RPC mechanism makes network programming easier, however it falls far short of making remote procedure calls transparent to the programmer.

Due to the performance penalty, Sun RPC should not be used in applications that require large amounts of data transfer between clients and servers.❖

Figure 2:
Run time versus
packet size



Transfer rate
(kbytes/sec)

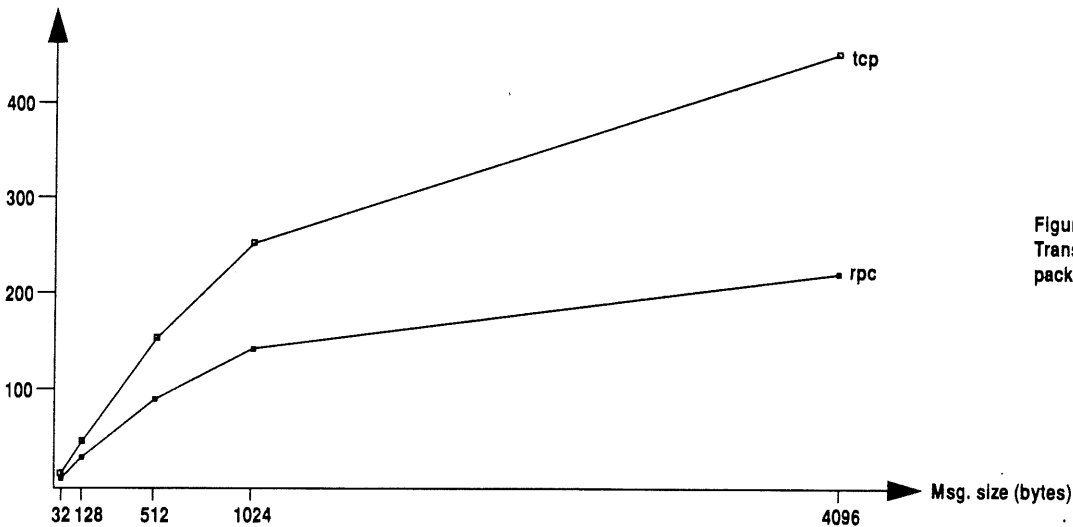


Figure 3:
Transfer rate versus
packet size

References

ARCH89

Architecture Projects Management Ltd.: *ANSA: An Engineer's Introduction to the Architecture* TR.03.02 Cambridge UK November 1989

BLAC92

Ulysses Black: *TCP/IP And Related Protocols* McGraw-Hill Incorporated, New York, NY, USA 1992

CHES94

William R. Cheswick, Steven M. Bellovin: *Firewalls and Internet Security Reading*, Addison-Wesley, MA, USA 1994

COMP94

Computer Systems Research Group, University of California at Berkeley: *4.BSD-Lite Source CD-ROM* O'Reilly & Associates, Incorporated, Sebastopol, CA, USA April 1994

STEV90

W. Richard Stevens: *UNIX Network Programming*

Prentice-Hall Incorporated, Englewood Cliffs, NJ, USA 1990

SUNM87

Sun Microsystems: *XDR: External Data Representation Standard* Internet Network Working Group RFC1014 Internet Request For Comments, June 1987

SUNM88

Sun Microsystems: *RPC: Remote Procedure Call Protocol Specification Version 2* Internet Network Working Group RFC1057 Internet Request For Comments, June 1988

SUNM90

Sun Microsystems: *Network Programming Guide* Sun Microsystems Incorporated Part Number: 800-3850-10 Manual set for SunOS 4.1.x, Mountain View, CA, USA March 1990

TANE92

Andrew S. Tanenbaum: *Modern Operating Systems* Prentice-Hall International Incorporated, Englewood Cliffs, NJ, USA 1992

UNIX Tricks & Traps

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

Consistent Binary Support for Multiple Architectures Across a Common Filesystem

David Leonard
<D.Leonard@fnarg.net.au>

A trick I picked up from Scott Merillees has come in infinitely useful over the last few years, especially since I have been working in an NFSed environment on (and off) about five or six different platforms.

Most of us have a personal `~/bin` directory, maybe even a `~/lib` (and some even have our own `~/etc`!)

If your `~/bin` contains a binary compiled under, say, HP/UX, then it more than likely won't work when you move over to your Alpha.

The blindingly simple and obvious solution to this problem is outlined as follows: compile your binary for each architecture, place in an architecture dependent directory and correct your search `PATH` at login.

The `uname(1)` program can give you the machine architecture, as well as the operating system name, and version, which you can then use to form your `PATH` environment variable.

`uname` is found on virtually all UNIXes these days; and even if it's not, you could quickly write your own.

For example:

```
$ uname -m; uname -s; uname -r
mac68k
NetBSD
1.0
$ PATH=$PATH:$HOME/bin/arch/'uname -m'; export PATH
```

Or, to be really over the top:

```
$ PATH=$PATH:$HOME/bin/arch/'uname -m'/'uname -s'/'uname -r'
$ export PATH
```

My environments are not that aggressive, so only a single depth `~/bin/arch` satisfies my needs. This directory contains further subdirectories `alpha`, `RISC`, `sun4` and `mac68k`.

Sun's `uname -m` sometimes returns `sun4c` or `sun4m` but you can get around this with appropriate symbolic links.

For example, I have `hunt(6)` compiled for three architectures:

```
48 /home/leonard/bin/arch/alpha/hunt*
92 /home/leonard/bin/arch/RISC/hunt*
80 /home/leonard/bin/arch/mac68k/hunt*
```

For sites that have NIS netgroups...

Tim Cook
<tim@deakin.edu.au>

The following program tells you what netgroups you (or other users) are in.

```
#!/bin/sh
# netgroups -Like groups(1), but for netgroups
#
# $Id: netgroups,v 1.1 1994/05/16 06:19:40 tim Exp $
# $Source: /src/usr/local/bin/netgroups/netgroups,v $
#

prog="netgroups"

PATH=/bin:/usr/bin:/usr/ucb ; export PATH

case $# in
0 )
;;
1 )
USER="$1"
;;
* )
echo "usage: $prog [USER]" 1>&2 ; exit 1
;;
esac
case "$USER" in
USER="`whoami 2>/dev/null`"
;;
esac
case "$USER" in
echo "$prog: who are you?" 1>&2 ; exit 1
;;
esac
ypmatch "$USER". "*" netgroup.byuser | sed 's/,/ /g'
```

Please send your contributions for this column to the UNIX Tricks & Traps/User Support Mailbox
Sub-editor:

Janet Jackson
janet@dialix.oz.au
phone/fax: (09) 272 5061.

Suggestions for topics are also welcome.

Opinion:

Death of the Net?

Chris Maltby

In a small-scale rerun of the infamous Internet Worm panic, the *New York Times* published a front page story on another Internet security crisis. The *Sydney Morning Herald* reprinted the story in its next edition.

The story contained the usual mixture of fact, fantasy and hype. Non-technical readers might be forgiven for assuming that the long-predicted "death of the net" was at hand. According to the report, most home computers were at (some unspecified) risk of attack.

Sticking to the facts, so far as they are known, the story is a lot less exciting. It certainly seems as if a few hackers (motivation uncertain) have penetrated some systems presumed to be safe in that they were "protected" by various firewalls. A second hack was then applied to some of the penetrated systems to allow authorised sessions in progress to be hijacked (via some kernel table patching).

To get really boring, it is necessary to point out that the method of attack has been documented since at least 1987 in work by Robert T. Morris (Sr.) and Steve Bellovin (both at Bell Labs at the time). It is yet another manifestation of the hoary old problem of service authentication based on the requesting host's IP address alone. For an example, consider the contents of your system's `/etc/hosts.equiv` file.

The current hack is a refinement of the widely known source-routing hack, in which packets with a trusted source address are supplied with an optional "source route" which takes them to an untrusted spoofing host. By selecting your apparent source address appropriately, you can make the system under attack give you a login shell without asking you for a password.

The refinement is that by being able to predict in advance the contents of the necessary handshake packets, packets with bogus source addresses can be sent without the telltale source route option. The prediction is possible because packet sequence numbers are not randomised by the originating host.

So how should it be fixed. First and foremost, the net needs to move away from host address based authentication to something which uses some form of secure cryptography (say Kerberos with a public key extension). This will protect your hosts even if your firewall is breached.

Second, and more easily implemented, is to filter out incoming packets on untrusted network interfaces which should only have come from trusted networks or hosts. Unfortunately, it's only very recent revisions

of most router software which support any kind of inbound filtering. The outbound filtering they do support is inadequate against this attack.

However, your Internet service provider can install outbound filtering for your link, though you may not want to fully trust this. A second router is another possible way to set up this kind of security. A firewall built around a UNIX host may need some extensive kernel modification to make it safe, unless it trusts nothing on the network, not even itself; this is hard. Remember that filtering rules based solely on the packet source address without knowledge of where the packet appeared (screend, SOCKS, etc.) are not enough either.

Last of all, and least importantly, some improvement to the assignment of TCP sequence numbers to new connections to reduce guessability would be handy, though not enough in itself. Be cautious of vendors offering this as a "solution".

Once your system is penetrated, there is no way to prevent a hacker from hijacking any existing connections and using them to penetrate those systems as well. For Sun systems, the hijack is made easy by some useful (to legitimate administrators) but dangerous software available from UNIX source archives.

For further information on this, and many other security threats, browse the WWW and FTP archive at the Computer Emergency Response Team's site cert.org, or join the mailing list for CERT advisories. And above all, don't panic.❖

AUUG BOOK CLUB & PRENTICE HALL AUSTRALIA

20% DISCOUNT TO AUUG MEMBERS

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

- The Internet Book**
ISBN: 0 1315 1565 9, Paper, 1994, \$44.95*

*Deduct 20% from listed retail price

Name: _____ Position: _____
Organisation: _____
Address: _____
Telephone: _____

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

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


Company purchase Order No. _____

Please charge my: Bankcard Visa MasterCard

Credit Card No:

Expiry Date: _____ Signature: _____

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

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



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

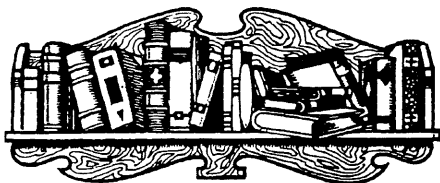
A Paramount Communications Company

Book Reviews

Frank Crawford, Coordinator
<frank@atom.ansto.gov.au>

Welcome back, I hope you all had a good Christmas, certainly some of our reviewers did. There are a stack of reviews both for books offered by AUUG for review, and others that members felt sufficiently motivated to write about. A number of the books were in manuscript form, so the reviewers may not have been able to comment on features like CD-ROMs, etc. I hope to shortly have a few more books for review, especially those from Prentice-Hall and Addison-Wesley, so watch out for a notice. The current practice is to post a note to the newsgroup aus.org.auug when new books are 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. ❖



Building a Successful Software Business

by Dave Radin
O'Reilly and Associates 1994,
371 pp
ISBN 1-56592-064-3
Reviewed by Zoltan Somogyi
<zs@cs.mu.oz.au>

The stated objective of this book is to help technical people set up their own small business selling software. It drives home the point that technical excellence is far from enough to ensure the success of a software company, and aims to teach the reader about the other necessary ingredients. These include: identifying a market identifying prospective customers getting your message across to them choosing distribution channels managing cash flow and taxes obtaining funding

The book is very readable. Most topics seem to be described at the right level of detail. Several chapters, e.g. the one on distribution channels, contain a section describing each possible alternative (retail resellers, wholesale distributors, mail order, shareware, etc.) followed by a table listing the advantages and

disadvantages of each alternative to make choice easier. Many points are illustrated with anecdotes involving well-known companies and stories drawn from the author's own experience.

Much of the material covered in the book is only common sense, but this should not prevent you from reading the book. Most of us wouldn't think of many of these issues until after we have been burned by them, by which time it is likely to be too late. For example, many of us would boast that we are not swayed by advertisements, but would be stumped when asked to come up with another way of communicating with potential customers.

I have learned several things from the book:

- Why trade show are not as good a source of customers as you might expect.
- Why seminars are a good source of customers but an expensive one.
- How to judge the quality of prospects and keep track of them as they turn into leads and hopefully customers.
- Why it is important to keep track of the success rate of your marketing efforts and how it can be done.
- Why it is easy to make a mess of order processing, and how to avoid the pitfalls.
- And even some tips on physical packaging.

The book's discussion of legal and accounting issues (particularly those relating to taxes) will be less useful for readers here than readers in the US, but they still serve the purpose of alerting the unwary to the presence of minefields, such the possibility of being taxed for income you haven't got yet.

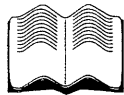
Unusually for a book of this type, this one discusses ethics in several places, mostly in ways I agree with. Two exceptions are 'cold' calls in telemarketing, which the book cans on business and not on ethical grounds, and the user market segmentation, selling the same product at different prices to different people. While segmentation may be a necessary tactic and may be defensible in some cases (e.g. quantity discounts reflect the reduced per-copy cost of the selling process), the book does not mention that in other cases it is not only unethical but also counterproductive.

One instance that will be familiar to many AUUGN readers is the premium that many software companies ask for delivering an application on a UNIX platform instead of a PC or a Mac. The assumption this is based on, that sites using UNIX have more money to spend than sites using PCs or Macs, is untrue in many cases, particularly for universities.

I would recommend this book to anyone who is starting up a software business or is thinking about doing so. Even if you think you know everything you need to know, it is good to have a checklist such as this

book to make sure. It may give you ideas you'd never have thought of, such as buying shares in your competitors just to get on their shareholders' mailing lists.

I would also recommend the book to anyone presently working for a small software company. The topics covered by this book are quite soft, and I know that some AUUGN readers will not like this fact, but even the most technically oriented programmers should know the business facts of life, if only for self-defence.❖



Computer Related Risks

by Peter G. Neumann
Addison-Wesley
ISBN 0-201-55805-X

Reviewed by Adrian Booth, Arena Technology <arena@dialix.oz.au>

For those who don't know, Peter G. Neumann (pgn) is the moderator of comp.risks, an excellent newsgroup containing (usually) serious discussion of the dangers of technology and its application.

pgn has summarised many, many cases where technology has caused errors, from the humorous (a monthly electricity bill for four billion dollars) to the disastrous (too many to choose from).

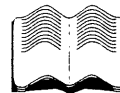
In many cases though, it isn't the technology itself that is at fault: other possible causes range from problems with the initial conceptualisation of a system, through design and implementation, maintenance and patches, and (of course) simple human error.

For example, the (initially) much-vaunted Patriot antimissile systems used in the Gulf War were only designed to work for 14 hours at a time (i.e. in their initial specification); this coupled with drifting clocks (a software implementation flaw) and 100-hour operational shifts (operational conditions) led to their becoming less and less accurate over time, until they were missing incoming missiles by over 600 meters (p. 34).

After an introductory chapter on the nature of risks, subsequent chapters cover reliability and safety problems (in such fields as communications, space, and civil aviation) and security vulnerabilities (intentional misuse, security accidents, and financial fraud).

A complete chapter then covers "Threats to Privacy and well-being", followed by excellent discussions of the broad field of risks, from both system- and human-oriented perspectives. A closing chapter discusses Implications and Conclusions.

I cannot recommend this book highly enough, even though after reading it you may never want to travel by plane, car, ship or train again, and may be putting all your cash under the bed! A quote from another review appears on the back cover: "This sobering description of many computer-related failures throughout our world deflates the hype and hubris of the industry." Run, do not walk to your nearest bookstore and buy a copy now.❖



Firewalls and Internet Security: Repelling the Wily Hacker

by William R. Cheswick and Steven M. Bellovin
Addison-Wesley 1994,
306 pages
ISBN 0-201-63357-4

Reviewed by Danny Yee
<danny@cs.su.oz.au>

Cheswick and Bellovin have written the first book that deals specifically with the security of whole networks rather than of individual hosts. Based on their experience administering the Internet firewall at AT&T, as well as on existing papers and reports, *Firewalls and Internet Security* tells you how to connect a network to the Internet without exposing all your computers to nefarious attacks. It begins with an introduction to security issues and a review of TCP/IP protocols from the point of view of security, but the reader is assumed to have a good understanding of the TCP/IP, an understanding of basic security concepts and some knowledge of UNIX.

The core of the book is a detailed look at how to set up and run a firewall. This begins by covering the mechanics of setting up a packet filter, application and circuit gateways, the uses and abuses of tunnelling and the general limitations of firewalls. A long chapter then goes into some detail in describing the application level gateway setup at AT&T. Also contains a brief discussion of user authentication and a description of useful tools such as connection libraries, network monitors and logging programs. (They recommend doing a lot of logging.) Also discussed are counter-intelligence measures, decoys and lures, and how to use standard hacking tools to test your security yourself. The stress throughout is on keeping things simple, in traditional UNIX style.

Cheswick and Bellovin then look at how things actually work in practice. Here they present a general typology of network attacks, an account of their encounter with the infamous 'Berferd' hacker in 1991, and some statistics on penetration attempts from their logs. I'm a bit unsure about some of the conclusions

they draw from the latter (see below), but it's good to see some statistics being published.

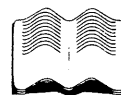
To round things off there are chapters on legal issues (if you watch a hacker instead of kicking him off at once, are you responsible for any damage he does while using your system to connect elsewhere?) and cryptography. The appendix contains a list of free resources - software packages and information sources - available to those trying to maintain secure networks, a port by port analysis of TCP and UDP protocol weaknesses and some suggestions for vendors and manufacturers of networking hardware and software.

This is great stuff, and I have only one quibble. I feel Cheswick and Bellovin are a little too paranoid in places, not in their evaluation of possible threats or in the precautions they suggest, but in their evaluation of the intensity of hacking activity. So attempts to rlogin in to their gateway as root, while they may be "evil", are almost certainly due to bored university undergraduates - I should think it's the last thing a competent hacker would try.

(Of course competent hackers probably have more sense than to attack a hardened target like AT&T at all, let alone head on.) Attempts to log in as guest, demo or visitor are surely signs of cluelessness, and hardly deserve to be labelled "attacks" or "evil". And a graph which is supposed to show that hackers are less active on weekends, to me suggests instead that most of their "penetration" attempts are from company employees or university students who don't even have net access on weekends. Using the term "hacker" instead of "cracker" for those up to no good is one thing; debasing the term to include everyone capable of typing "rlogin research.att.com -l root" is another. It's a far cry from that to being able to mount sequence number attacks on TCP connections.

Firewalls and Internet Security has no rival; while much of the information it provides is available elsewhere, no comparable summary exists. Anyone in charge of installing or administering an Internet firewall would be insane not to get a copy. And while some of it is irrelevant to smaller sites, much will be useful to anyone concerned with TCP/IP network security. That said, it should be pointed out again that this is not an introductory book on security; not only does it assume a solid knowledge of internet protocols, but it doesn't deal with anything except external network threats. Of course anyone with pretensions to being an Internet hacker will also want to read this book (if only to find out why they shouldn't try to crack AT&T :-)) and it can be read just for enjoyment. As well as being extremely informative, Firewalls is also extremely entertaining, with the authors managing to inject some light-heartedness into their

subject while still respecting its seriousness. I finished my copy within a day of receiving it. ❖



The Internet Book

by Douglas E. Comer

Prentice-Hall 1994,

312pp

ISBN 0-13-151565-9

Reviewed by Craig Macbride

<craig@rmit.edu.au>

When a book has a subtitle of "Everything you need to know about computer networking and how the Internet works", it's setting itself an enormous task. In this case, it does come pretty close.

For the person who knows nothing about computer networks, The Internet Book starts by explaining telephony, telegraphy, Morse code, modern digital communications, local area networks, wide area networks and then moves on to a history of the Internet. Most readers of AUUGN would, I suspect, not find much that's new to them in this, except perhaps for some of the historical parts.

After TCP/IP and IP addresses are explained, along with the naming of machines and domains, the rest of the book is concerned with what services the Internet supplies and how to use them. Electronic mail, mailing lists, public mailing lists, newsgroups, FTP, Telnet, Gopher, WWW, Archie, Veronica and WAIS are all covered in detail. Examples are included of how to access each of these services on the Internet and how to use them. There is even a list of archie servers worldwide.

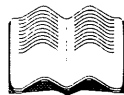
In one appendix, there are a large list of news groups and what they are likely to contain. Maybe it is a sign of the politically correct times that the author gives a much broader breakdown of the subgroups of some news groups, such as soc.religion than, say, alt.sex. Quite a number of groups within rec.arts are mentioned, but not the most widely read one.

The Internet Book manages to cover all of this with as little reference to machine types and operating systems as possible, which is quite an achievement. At some points, it is inevitable that specific programs are used and it seems a shame to me that the ugly "rn" is used as the example news reader.

I found it odd that a few things went unmentioned. FSP is not mentioned at all, even in the glossary. Its rise to any significant use may have occurred after the book was published. More surprising was that the "r" commands (rlogin, rcp and so on) are not mentioned at all. Furthermore, the author actually refers to Telnet as

“remote login” throughout one chapter. Thankfully, he doesn’t refer to FTP as “remote copy” to create further confusion.

All in all, this is a very comprehensive book to guide the virtually computer illiterate person all the way to understanding and using the Internet. ❖



Internet CD

by Vivian Neou

Prentice Hall

1994, 260 Pages (180 text, 80 appendices and index)

ISBN 0-13-123852-3

Reviewed by David J. Hughes, Bond University

<Bambi@Bond.edu.au>

Have you ever got to the end of a book, placed it back on the coffee table and just sat there asking yourself “Why?”? That’s the kind of reaction that “Internet CD” aroused in me. It’s hard to tell exactly at whom this book is aimed. In its introduction the book states that it is intended for experienced computer users who want to learn about the Internet. It then adds to that statement by saying that if you are a Windows user you should have enough knowledge to be able to add a Program Item and run a program from the Program Manager. This loose definition of an “experienced user” jumps from someone who can click on an icon under Windows to someone who understands IRQ settings and base IO addresses of the comms hardware of a PC. In short it’s hard to know where this book is supposed to fit in.

Accompanying “Internet CD” is a CD-ROM containing “All the Internet software and documentation you’ll ever need, on one packed CD-ROM”. It is pretty much packed to the brim with PC related software including Eudora for Windows, UUPC, WAIS, Gopher, Trumpet, Crynwr Packet Drivers and other bits and pieces. It also includes a copy of Linux, a full set of RFCs from the time of printing, mailing list archives (NameDroppers, BIND and TCP/IP), and various information sources (IETF, IEN, FYI etc.).

The structure of the book is basically as follows:

- roughly 15 pages describing various Internet services (FTP, telnet, archie, WAIS, e-mail etc.)
- 35 pages covering the installation and setup of a Windows based PC running a packet driver, an IP stack, and various applications
- 15 pages on compiling and installing gopher, WAIS and archie on a UNIX box
- 12 pages on using a text searching tool provided to enable searching of the RFCs, etc.

- about 30 pages on setting up Linux
- a 10 page chapter on setting up UUPC
- a 25 page reprint of the Gopher User’s Manual
- Various appendices, including a list of service providers and a couple of FAQs.

I can’t understand how anyone can describe the functionality of FTP, telnet, WAIS, archie, gopher, veronica and WWW in 15 pages. The book then spends 12 pages describing how to use a simple text searching utility provided with the RFCs on the CD. There just doesn’t appear to be a reasonable balance of information here. This imbalance becomes more apparent when you realise that the World Wide Web is covered in two short pages. I’m still trying to decide how configuring a PC for a UUCP connection fits in to the scheme of things.

When I received this book and noticed the SRI logo on the front I thought that it would be either a good book for someone who doesn’t know anything about the net or possibly a good reference for more experienced users. Unfortunately, it’s neither. The lasting impression of “Internet CD” is of a collection of bits and pieces tossed together like a limp garden salad and lobbed at the growing commercial Internet bandwagon. ❖



The Linux Network Administrator’s Guide

by Olaf Kirch

O’Reilly and Associates 1994,

335 pages

Reviewed by David Conran

<lucifer@dstc.edu.au>

Though the title suggests this book is ‘just’ for Linux, it covers most initial setup of network services in a fairly generic manner, thus making it useful not just for Linux, but for all the UNIXs out there. Of course there are some differences between the UNIXs and the book doesn’t point out where Linux varies from the pack, but even a young/novice System Administrator will spot the differences with ease. The book details how to setup almost every basic Internet service that any user could dare to want, the exceptions are the very advanced/non-standard services like multi-cast, AFS etc. But if you are setting these up, you probably won’t need this book. It is more focused on getting a user or a new System Administrator up and running on a network or the Internet as quickly as possible while explaining what everything does, be it from a UUCP mail/news feed to direct Ethernet in a corporate LAN.

The first two chapters, *Introduction to Networking & Issues of TCP/IP Networking*, introduce the network novice to a brief history of UNIX style networks from the dawn of time to the current. It explains the different types of network hardware and their uses & protocols. What makes up TCP/IP and how it actually works. Then it goes on to explain the pedigree of Linux's networking code and from where to obtain it. Network interfaces, IP addresses and structures and resolution are then discussed and explained. The nature of routing and how things find one another in the DNS hierarchy are detailed, while all the time explaining how all the configuration files are setup and where to make changes.

The next chapter, "Configuring the Networking Hardware" explains how to configure Linux kernels to detect & use different networking capable hardware varying from serial ports for SLIP/PPP to parallel ports for PLIP and parallel port Ethernet adapters to the huge array of PC Ethernet cards. It explains what all the kernel options do with regards to networking (though with the rate at which Linux develops, this is always out of date) and how to make the various cables necessary for each of the network style interfaces.

"Setting up the Serial Hardware" gives detailed instructions on how to prepare the serial ports for use with networking, like what the devices are called and how to set the baud rates, flow control and which UARTs you should have.

"Configuring TCP/IP Networking" puts the above theory into practice, including how and when to set the hostname, what you put in `/etc/hosts` and `/etc/networks` files, how to configure interfaces with `ifconfig`, setting up host and network routes and gateways.

Then it lists some of the tools to examine you setup such as `netstat`, `ping` and `arp`. Most the rest of this book could be considered generic enough to work with any UNIX system.

Chapter 6, "Name Service and Resolver Configuration" contains how to hand build you DNS database from scratch, what each line of `/etc/resolv.conf`, `named.boot` and the rest of the DNS files does. It also shows how to use `nslookup` to check that you have it all working perfectly.

Chapter 7 & 8, "Serial Line IP" & "The Point-to-Point Protocol" are probably the most important for the majority of Linux users. This is because most of the Linux systems are home-based, forcing Home sysadmins to connect to networks over serial lines. These chapters explain how to setup each of the protocols in a step-by-step fashion to connect your Linux Box to the Internet. It also goes on to show you how to configure automatic dialup scripts so you will never have to the complex process manually. Not

content with the trivial case of dialling up a SLIP/PPP server, the book goes on to explain & show the reader how to setup their own SLIP/PPP server.

Chapter 9, "Various Network Applications". Here the book explains all the network daemons that are likely to be running on a UNIX box & of course, how to set them up. This includes `inetd`, `tcpd` (TCP wrapper), services & protocol files, Remote Procedure Calls (RPC) and the `r*` commands.

Chapter 10, "The Networking Information System" or to the older folk of UNIX, "Yellow Pages" is explained and how to configure a system to be a client of a NIS server and of course, where to get the software from.

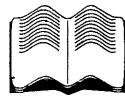
Chapter 11, "The Networking File System", (NFS) is described. How it works, what limitations Linux's implementation has (or had, the kernel just keeps on improving :) how to check if you system has NFS built in to the kernel. It shows the reader how to mount a remote NFS file-system including all the common mount options and what their effects are. It then goes on to explain what daemons and files are necessary to be an NFS server and how to setup the permissions securely.

The next Chapter, 12, "Managing Taylor UUCP" is huge! It explains more of the history of UUCP and theory and use of this old protocol. It goes on to explain all of UUCP's programs and configuration files, it tell you what you need to know to be able to understand UUCP naming, how it all works, what restrictions you can place on it, how to run it over TCP, file transfers, forwarding, dialup scripting, setting up `gettys`, account setups and how to protect yourself. It goes on to explain the low levels of the UUCP protocol and what to look for in the log files.

The next three chapters deal with Electronic Mail. Chapter 13 explains what email is, how it is delivered including Internet & UUCP addressing and routing (and hybrids of both) and how to configure the 'elm' mail reading program. Chapter 14 details all you need to know to setup an 'smail' server/client with UUCP or SMTP, config options, routing, aliases and mailing lists while Chapter 15 details how to do roughly the same things with IDA Sendmail, though as most real people on the net use Sendmail 8.6.9, this seemed a little pointless. (Especially as Linux now ships with the option of installing 8.6.9 instead of IDA)

The rest of the book (Chapters 16 through 19) detail and explain Usenet News. Starting with a brief history, some terminology and some of the logistics of news (e.g. 60+ Mb. per day) to how to configure and maintain C News (No mention of INN which seems more common in the Internet community). NNTP is described briefly, but imparting all you need to know to set yourself up as a secure NNTP server. Even briefer description of how to configure the various Newsreaders (`nn`, `trn`, `tin`) is supplied, but from my

experience, you will need more information and real experience of maintaining a news feed before you will be able to configure them comfortably and be able to answer all of their configuration questions. On the whole, the book is very useful for a System Administrator who has never had to look after a network of machines or machine connected to the Internet. The book is ideal for the home user who has purchased a Linux CD and has installed it on their home PC and then wishes to connect to the Internet via their home phone line. If you are an experienced UNIX System Administrator with lots of machines on the Internet, then my guess is this book is not for you. I found several parts of the book very interesting and topical to me as I had at work just gone through the pain of setting up a Linux box as a router with two Ethernet cards and I had to connect several 'home' PC's running Linux to our corporate network via PPP. Both of which took me a while and a bit of frustration, only to have this book arrive to be reviewed (which had simple instructions & information on how to do the above) a week after I had every thing setup. Such is life! :-)*



Motif Tools

by David Flanigan
O'Reilly & Associates
1994, 984 Pages + CD
ISBN 1-56592-055-9

*Reviewed by David J. Hughes, Bond University
<Bambi@Bond.edu.au>*

The promotional material from O'Reilly & Assoc. for this book portrayed it as the single most valuable set of tools for Motif programmers and I'd say that they're not far from the truth! The material covered in "Motif Tools" has the potential to save vast amounts of time and effort during Motif based GUI design and development.

The book itself is a very detailed tutorial and reference manual for the motif tools library (libXmt) written by David Flanigan of Dovetail Systems. Xmt is a collection of 8 custom widgets, 260 convenience routines and a handful of resource converters that offer a new level of flexibility and productivity to the Motif coder. Using these tools it is a very simple task to whip up a prototype or even production GUI for an application before the ink has dried on the project spec. It is really that good.

The new widgets offered by Xmt have been designed to take the hard work out of Motif programming. Anyone that has spent hours mulling over pages of form attachments trying to get the desired layout of a

dialog box will take great pleasure in using the XmtLayout widget. This widget offer the same type of flexibility and ease of use that people only expect from the Tk toolkit for Tcl. The Xmt chooser widget also takes the hard work out of radio boxes, check boxes and other related objects by offering a simple syntax for the creation of all the required elements rather than leaving the programmer to do all the work.

The other new widgets strive for the same goal: make it easy and efficient.

One of the nicest features of Xmt from a productivity point of view is nothing new to the X programming world, although it is new to the Motif programmer. Via the addition of a few new resource converters, Xmt allows just about your entire GUI to be specified as application resources. This not only includes widget appearance and layout but also widget creation. It is possible to create a fully functioning GUI, including callbacks to dialog boxes and other UI elements, without writing a single line of C. The development cycle is incredibly fast as there is no compilation or linking involved. It's a matter of execute, edit, execute. This type of functionality has been available via the Widget Creation Library but to have it tied in with the other features of the Xmt library makes development a breeze.

Many of the features provided by Xmt are there just to increase the ease of use of Motif. One example of this is the implementation of a very simple yet incredibly useful font handling mechanism. The programmer may specify a list of fonts to be used within the application, either within the code or via the resource file, and associate a name with each font. From then on, font information can be included in labels and other text strings to be displayed in the GUI. For example, the programmer may decide to use a 48 point Times font for dialog headers. If that font is called HEAD in the applications font list, a label string of "@f[HEAD]This is a header" will force the desired font to be used for the string. Fonts can be mixed to any extent allowing things like "@f[HEAD] This is in a @f[BOLD]bold @f[HEAD] font" to generate the desired effect. It's a simple idea but saves so much time.

"Motif Tools" is quite a large and daunting book weighing in at close to a thousand pages although it is structured to offer both tutorial and reference sections for all the features of the library. After a couple of days using it you'll find that all you need is a few post-it notes marking important pages and you'll be pumping out Motif code faster than you thought possible without a code generator.

The book comes with a CD containing a copy of the library's source code and some example programs from the book. If you want to have a look at the library before buying the book you can grab the code via FTP from <ftp://ftp.ora.com/pub/examples/xbook/Xmt/>

xmt200.tar.Z. The book comes in at about \$110 retail so you probably should have a look at the library before rushing to the bookshop. If you're anything like me it'll only take a couple of hours before you're heading off to Dymocks.❖



PGP: Pretty Good Privacy

by Simson Garfinkel
O'Reilly and Associates, Inc. 1994
Reviewed by Lawson Hanson
B & H Computer Services Pty Ltd

Divided into three main parts, the text of PGP: Pretty Good Privacy, is a highly readable, and fairly thorough treatment of cryptography in general, and Phil Zimmerman's PGP system in particular. The three parts are: Part I: PGP Overview and Background, providing excellent explanations of both the earlier Private Key, and the more recent Public Key cryptographic systems, including Digital Signatures, and some of the Privacy, Politics, and Patents Litigation concerns that surround the subject; Part II: Using PGP, describing in great detail how to use PGP to create your public and secret keys, use your pass phrase, maintain key rings, and digital signatures, to enable you to successfully encrypt and decrypt messages, in a security conscious manner; and Part III: Appendices, contains information about how to obtain and install PGP software on UNIX, DOS and Macintosh computers, and has a description of some of the mathematics of cryptography.

The book starts off with a fascinating discussion of the history of cryptography and cryptanalysis, covering many of the important turning points in its colourful past. The strengths and weaknesses of the various schemes is outlined, and the reader is brought to understand the necessity for the research that led to more recent discoveries involving the idea of public and secret keys in comparison to the older, more vulnerable private key systems.

Considerable bandwidth is given to covering the dilemma we face in regard to the need for private and business information security, versus the requirements of government and national security bodies, who sometimes need to have wire-tapping powers to catch criminals. Communications between commercial companies needs to be secure to prevent (or at least frustrate) industrial espionage; but at the same time, needs to be monitored to prevent large scale criminal fraud. It is reported that US telephone companies are being forced to provide wire-tap capabilities for the FBI and other law enforcement agencies, and that they face a \$10,000 per day fine for non-compliance. But

what of computer modem communications where the data has already been privately encrypted?

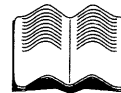
It appears that the PGP system has been very carefully thought through; for example, your secret key is usually encrypted with a pass phrase. This means that if your computer is stolen, or somebody gets hold of a copy of your secret key file, they will not be able to use it to read your encrypted mail, and neither will they be able to sign a bogus message from you!

People who have your public key can send you encrypted messages, but they can not read encrypted messages that other people have sent you, unless they also manage to get both your secret key, and its pass phrase. In general, by using the PGP system, you can be sure of three things:

- Since a file is encrypted using the recipient's public key, only someone with the recipient's secret key (and pass phrase) may read it.
- Because the message is signed with the sender's digital signature, the recipient can be assured that the sender is who they claim to be.
- When a message has a digital signature, PGP can detect whether there have been any alterations, and will warn you if there are.

Of course, for these three things to be assured, you and the recipient must be in possession of verified copies of the each other's public key. Garfinkel's text also discusses the purpose of public key servers, and the lengths to which one must go to be assured that what you get is what it claims to be, etc. The book is packed with lots of practical examples of how PGP is used in real life, and covers many of the options and the ways in which they may be combined to produce most effective results.

PGP is available for private use at no cost. Commercial use licenses are also available. If secure computer communications are important to you, then you should consider using PGP. With a 1024 bit key size, if you could somehow get 100 million high end (100 MIPS) personal computers to help crack the key, it would still take you an estimated 280,000 years to decipher. Any way you look at it, that makes PGP - Pretty Good Privacy.❖



A Quarter Century of UNIX

by Peter H. Salus
Addison-Wesley 1994,
256 pages
ISBN 0-201-54777-5
Reviewed by Danny Yee
<danny@cs.su.oz.au>

A Quarter Century of UNIX is an oral history of UNIX in the form of an annotated collection of reminiscences. It begins at the "birth" of UNIX, with Ken Thompson looking for a machine to play Space Travel on, then jumps back to provide the context, both in the history of computing in general and in the particular setup at Bell Labs. Part two describes the work done up to 1974, both on UNIX and on the tools and language (C) so closely associated with it. Part three tries to pin down some of the things that made UNIX unique: its style, the strong contributions by users and user groups, and the key role of some of its more famous tools. Parts four and five trace the expansion of UNIX: the development of BSD and the commercial UNIXes, the creation of SUN, the ambivalent relationship with DEC, and attempts at standardization. The final section offers an overview of the current status of UNIX (in its many different versions) and offers some suggestions as to where it is heading. There is also a very brief glance at some of the systems that it has influenced, including Bell Lab's new Plan 9 system. The finale has Dennis Ritchie, Brian Kernighan and others offering their ideas on what made UNIX work. Particularly noteworthy is the solid treatment of legal issues (three chapters altogether) and the coverage of events outside the United States (in Australia, Europe and Japan).

The format of A Quarter Century of UNIX with most of the text in the form of extended quotations (some scores of people are quoted from at length), runs the risk of discontinuity and lack of focus. Salus has chosen and edited his source material well, however, and inserted his own summary and exposition in appropriate places. The result is both informative and enjoyable, with a good balance between the personal and the technical.

I did spot a few minor inconsistencies in the text; on page 155 we read "It was 32V that became 3BSD in 1979", though the UNIX versions tree on page 61 shows no such influence - and errors - on page 253 we have "It was clear that AT&T hadn't objected to other derivatives: Linux, MINIX, etc. In the autumn of 1988...", implying that Linux existed in 1988 (and Linus' name is misspelt in the index, too). But these are just quibbles. A more weighty criticism would be that the book sometimes reads more like myth than history, with the participants portrayed like epic heroes. This may worry the historians, but in a way it is the legends and myths that are the most influential, so the distinction is perhaps moot.

You don't need a lot of technical knowledge to read A Quarter Century of UNIX, but the more you know about UNIX (and to a lesser extent about architectures and operating systems) the more you will get out of it ... if you've never used `awk`, for example, you will probably have little interest in reading about its origins and development.

The main audience will be programmers, administrators and users with extensive UNIX experience, but people in marketing and management might well learn a thing or two from it, and historians and sociologists studying the computer industry will find Salus' work an essential source of primary material. A Quarter Century of UNIX should be a great success; it's only surprising that it wasn't written years ago!❖



X User Tools

by Linda Mui and Valerie Quercia
O'Reilly and Associates Inc.,
Sebastopol, CA 1994,
719pages, includes CD-ROM,
price unstated
ISBN unstated.

Reviewed by Andrew Wenn
Victoria University of Technology (Footscray).
<awenn@matilda.vut.edu.au>

Yet another X book from

O'Reilly and Associates, this time one aimed at the ordinary non-programming, non-guru end-user who wishes to learn just enough about X to customise their environment so they can become as productive as the users of other windowing systems. In short this is the type of book that I wish I'd had access to several years ago when I first started learning to use X. I have always liked O'Reilly publications and this one is no exception, it is so crammed full of information, hints and tips that even someone who is already very familiar with X is sure to find something of interest. Moreover the cramming has not been done at the expense of clarity.

The book is organised as a series of articles loosely grouped together by subject, the idea being that you can browse the articles of interest to you. Within each article, concepts that are likely to be unfamiliar are flagged with a pointer enabling the reader to find more information elsewhere in the text.

In most cases, I found this to be reasonably well implemented, but my pre-publication review copy contained quite a number of missing links, that I am sure will be fully in place in the published version. Although, Mui and Quercia have written a large number of the articles themselves, they have collected material from other experienced X users as well, so in effect, we have the combined wisdom of many users brought together in one large volume and presented in a non-daunting format. The editors are at pains to point out that this text will not tell you all about X, but

that it contains enough information to enable you to configure your working environment and application!

My review copy was also missing quite a number of graphics so I am unable to comment on the quality of overall layout of the book - I presume that it will be up to the usual high standards we have come to expect from this publisher. One other thing that I hope appears in the final version is an index - my copy had a fairly detailed table of contents but no index, something that I found very strange and rather annoying.

The published text will also come with a CD ROM containing much of the public domain software mentioned in the articles as well as the scripts for the various examples that are discussed. A casual browse through the book will reveal that there are CD ROM symbols on almost half the pages. It was a pity that my review copy did not come with the CD as it meant I was unable to assess the usefulness of many of the articles.

The 32 chapters of the text are divided into seven parts:

Part 1 consists of three chapters covering the basics of X, including a brief introduction to servers, clients, GUI's etc., how to start X with and without XDM and starting clients from the command line.

Part 2 has four chapters one on clocks (far too many mentioned if you ask me), another on alarms, calendars and other office management tools, another on some of the useful reference tools such as **tkman** and **xwebster** and the fourth chapter covers the gimmicky things for decorating you display and using valuable resources. Still it is surprising how many of us find them entertaining and useful(!).

Part 3, working with applications contains six chapters on mailers, x communication programs, networking tools (in the main X based tools such as **ftptool**, **xmosaic**, **xarchie** etc), editing files, x based games, and finally one on using **xterm** and **xclipboard** productively. As could be expected, none of the applications is discussed in great depth, rather there is enough information on how to start and configure the clients to your liking and often some handy hints on how to get the most out of each one.

Part 4, has four chapters one each on the **twm**, Motif, Openlook and **fvwm** window managers. As could be expected, each has details on menus and customisation and depending on the window manager, articles on focus types and keyboard short-cuts.

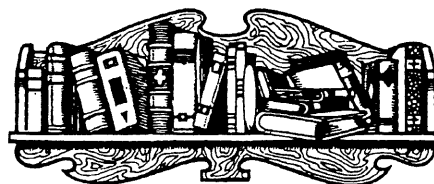
Part 5, User Environments, contains six chapters, four introductory ones covering startup scripts, remote clients, configuring applications and the keyboard and mouse. Two further chapters cover resources in more depth and advanced topics on the startup environment.

Part 6, Graphics consists of five chapters on bitmaps and icons, screen-dumps, drawing and viewing pictures, bitmap conversions and colour. Thus it will enable you to come up with either tremendous or horrible colour schemes depending on your artistic ability.

Part 7, the final section is for system administrators and would-be system administrators and contains three chapters on administration tools, configuring XDM, writing new tools (including some discussion of Tcl/Tk) and finally a chapter on how to use the CD-ROM.

Although it would be possible to read the book from cover to cover, it is really designed to be browsed, turning to a page and reading something that may interest you or make your life easier. Using this technique and imagining I was a fairly raw beginner, I found that I was able to read the simple articles, solve a problem and then progress to the more complex sections only if I needed or wanted to. For the most part my questions were answered in the first article I read.

I highly recommend this book to all X users be they just beginners or of many years standing. Help desk staff and system administrators should have a copy available as I am sure that it could be lent to end users thus solving at least one third of all commonly asked questions without consuming valuable time and resources. It contains a plethora of information and provided the cross referencing problems referred to above (as well as the typos) are corrected in the final published version it should be up to the high standards we have come to expect from this publisher. Highly recommended provided an index is included.❖



Interlude:

Yuletide Packets

by Lucy Chubb
<lucyc@suite.softway.oz.au>

This is the tale of an Internet elf, or how Santa went high tech.

What could seem more appropriate at Christmas time than a company (Softway) that is in the business of connecting corporate customers to the Internet, providing Santa with an Internet address?

What would Santa get from being connected to the Internet? Well, he could become a telecommuter. Think what this means to someone who would normally be stuck in the cold and snow of the North Pole for months at around this time of year. With an Internet connection (and a lot of busy elves to actually do the work) he could be basking on a tropical beach in Australia for a little while (until the Big Day when he actually did all the deliveries).

The origin of the mail was a surprise. Although it is not always easy to determine where a piece of electronic mail comes from, there was mail from all the following places: USA, Estonia, South Africa, Netherlands, Japan, Germany, McMurdo base (Antarctica), and even Australia (where the machine handling the mail is located). Although most of the messages were, mercifully, in English there was one in Estonian and another in Spanish. It took several days to work out what language the Estonian message was in. It turned into a truly international exercise.

The mail came from people of all ages (under 3 to over 80 - which is the oldest anyone admitted to being), from all different occupations - students, grandparents, researchers, and the military. The tone was sometimes serious, humorous, or sad.

Children mailed their Christmas wish list for Santa confident that he would listen. "Bigger children" had wish lists that (if fulfilled) would result in them spending considerable time and expense in repairing their floors even supposing Santa could have fitted the goodies down the chimney (a feat seemingly as difficult as Santa visiting all the homes on the planet in the one night). Some had wishes for themselves and some had wishes for other people. Some people wanted to wish the Internet Santa a Merry Christmas or to say what a good idea they thought it was. Others were lonely, hurt, and afraid - needing someone to talk to. What do you say to the child whose only Christmas wish is that dad would stop yelling and beating mum?

Quite a few people had questions for Santa, which gave the Internet elves a chance to exercise their creative juices. Here is a sample of the sort of things

some of the adults asked and the replies from various elves.

How do the reindeer like being on the Super-Highway? Santa answered: well, its a bit of a squash fitting them down the cable (especially the fibre), but once they're in they're quite happy about it. Its much like a rollercoaster: each sits in its packet and gets whisked along, which is much less effort than having to pull a great lunking sleigh.

The request for 365 pairs of slacks drew a suggestion from Santa that a new chest of drawers should be on the Christmas list also.

A number of people wanted Santa to provide theses for them - but Santa warning them that they might not get very good marks if the elves did it for them.



So, what conclusions has this Internet elf come to? Was it interesting? Yes. Was it worth while? Well, it was interesting. The Internet community is a "global village" on an enormous scale consisting of a bewildering variety of cultures and it is great to be part of it. The reaction of the Internet community has been generally positive, so we can be confident that it was appreciated. Would we do it again? Not without a lot more elves! I don't think the company could afford to do it again in this form because of the cost in manpower. The size of the event (putting Santa online) surprised some who decided to do it (and confirmed the fears of some who voted no) inundated as we were by mail from a bewildering variety of cultures.

It was surprising how quickly the exercise became large. While this was aided by wide media exposure, it could not have happened without media representatives and the general population having a significant background awareness of the Internet and how to use it. The largest volume of responses came from America - so what happened Australia? Are we more ignorant, in general, of the Internet or don't we write to Santa any more? ❖

AUUG Chapter News

From the Western Front

Janet Jackson, WA Chapter Sub-editor
<janet@perth.dialix.oz.au>

WAUG had a rather quiet meeting in December, maybe because of people being out of town, or maybe because it wasn't held at the usual venue. It would have to be that month that the paper meeting notices didn't get sent, presumably because some postal employee had had too much pre-Christmas spirit. Even more annoying to me is that our newsletter YAUN got lost too.

Anyway the December meeting was at IBM, who always put on excellent food (and beer these days, too) and usually have something interesting to show us. Indulis Bernsteins described in detail HACMP (High Availability Clustered Multi-Processing), which provides various ways for AIX boxes in a cluster to quickly and automatically take over each other's functions after a failure.

This software probably gets most use on high-end boxes, but Indulis showed that it can scale by successfully demonstrating it on a couple of antiquated RS-6000s, flicking off the switch of one CPU and watching the other take over its disk.

The speaker at the January meeting was Rob Chivers of AAA WinSoft Training on "MS Windows to UNIX Connectivity". The meeting notice advertised "...a general discussion on the issues involved in running UNIX applications from the MS Windows environment and integrating the two technologies." Disappointingly, it turned out to be a sales talk for a product, basically a fancy Windows terminal emulator (which looked to be quite fancy if that's what you're after). This could apparently talk to various different kinds of systems, including UNIX, over various kinds of connections.

This talk simply listed the features of the product and made no attempt to describe the technical issues involved in implementing such a thing even though the speaker said he had previously worked for the company that developed it.

Content aside, this talk provided a practical lesson for presenters and organisers, especially where projection panels are being used. Before you start, always test the exact combination of equipment you'll be using on the night, and preferably at the actual venue. Rob's slides were spoilt because the bottom few lines were missing, a casualty of some disagreement between his PC and the projection panel.

The WA Regional Group of SAGE-AU (the Systems Administrators Guild of Australia) has struggled on

through the Christmas period, when I think most systems administrators slink off overseas (i.e., to Rottneest) to get away from their users. Maybe we should have the Christmas meeting at Rotto next year, instead of at a restaurant, then we might get more than five attendees.

In January all but six members of SAGE-AU WA missed out on a very interesting talk on EDP Auditing from Alex Dermedoglou, the VP of the local chapter of the EDP Auditors Association. As usual at SAGE-AU meetings there was a lot of audience discussion.

But SAGE-AU needs more people, and I know there are a lot of systems administrators out there who don't realise the benefits of an association of your professional colleagues. And there's only one way you're going to find out.

Don't wait for the July conference! If you're a systems administrator (and maybe you don't know you are, thinking of yourself as a programmer or power user even though you spend most of your time looking after operating systems) do yourself a favour and get yourself along to a local SAGE-AU meeting.

Getting back to AUUG matters: by the time you read this the Summer Conference series will probably be underway. Perth organiser Adrian Booth has rounded up a good bunch of speakers this year mostly by asking people outright. The conference and tutorials are at the Freeway Hotel in South Perth, and will replace our usual monthly meeting.

WAUG, the WA Chapter of AUUG, meets at 6pm on the third Tuesday of each month. Our meeting organiser is Mark Baker, <waug-meetings@uniwa.uwa.edu.au>, (09) 491 6081. ♦

AUUG Inc. - Victorian Chapter (formerly SESSPOOLE)

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

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

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

Social meetings are currently held in the Bistro of The Grosvenor Tavern, 2A Equitable Place, Melbourne, starting at about 6:30pm. Venues for the technical meetings are varied and are announced prior to the event. The dates for the next few meetings are:

- Tue, 14 March '95 Social
- Wed, 19 April '95 Technical

Hope to see you there! To find out more about AUUG-Vic and its activities, contact the committee or look for announcements in the newsgroup aus.org.auug, or on the mailing list auugvic@clcs.com.au

The committee may be reached more directly on auugvic-exec@clcs.co.au.

AUUG-Vic committee:

President: Enno Davids
Metva
(03) 882 2333
enno@metva.technix.oz.au

Secretary: David Taylor
Monash University
(03)
davet@vaxc.cc.monash.edu.au

Treasurer: Neil Murray
Webster Computer
(03) 561 9999
neil@wcc.oz.au

Programme Chair: Michael Paddon
Kodak
(03) 353 2382
mwp@munnari.oz.au

Committee: Arnold Pears
La Trobe University
(03) 479 1144
pears@latcs1.lat.oz.au

Committee: Peter Lazarus
Legent Australia
plazarus@auspacific.legent.com

AUUG-Vic Email addresses:

General Membership: auugvic@clcs.com.au

Committee: auugvic-exec@clcs.com.au

Mailing list administration:
auugvic-request@clcs.com.au ❖

Call for Articles for *The Australian*

Lucy Chubb, co-ordinator
<lucyc@softway.sw.oz.au>
Tel: (02) 698 2322

The Australian newspaper runs an AUUG column every Tuesday, in its computer section. The aim of these articles is to inform the public and raise the profile of open systems within this country.

Having one's views published in a respected national paper also carries kudos and recognition for authors.

AUUG would like to ensure that all members of the open system community have access to this voice, and we are actively seeking a diverse spectrum of people and opinions.

If you are interested in being part of this process, please provide me with the following information:

- your name
- contact details
- a copy of your article

Submission guidelines

Your article should be between 600 and 800 words in length, and can address any issue that may be of interest within the open systems community. If you can't decide on an appropriate topic, please provide me with some professional details and I'll try and give you some ideas tailored to your expertise. Some typical subjects are listed at the end of this article.

If you have access to email, this is the preferred form of submission. Please format your article as a plain text file, with lines no longer than 79 characters, and with a blank line separating paragraphs. If you don't have email, please provide a hardcopy in a similar format (there's not much point doing any fancy typesetting).

All submissions are accepted on the understanding that they may or may not be used and that the material may be edited. AUUG will only submit your work to the Australian newspaper, although unless you advise us otherwise we will reserve the right to add your articles to a public FTP archive at some time in the future, and reprint them in AUUG's newsletter, AUUGN. The copyright on the material remains yours, your act of submitting material only gives us licence for the abovementioned purposes.

In practice, I submit your work to the Australian unedited and leave the decision of what to print up to them (I'm not in the business of being a thought police!). Usually a period of 2 to 4 weeks will then pass before you'll see your article in print; I maintain a pipeline of material to buffer me against the inevitable fluctuations in supply.

Please email or phone me if you have any further queries. ❖

AUUG Business



AUUG Incorporated ELECTION PROCEDURES

David Purdue, Returning Officer of AUUG Incorporated
<davidp@knowledge.com.au>

These rules were approved by the AUUG Inc. Management Committee on 14/12/1994

1. NOTICE OF ELECTION

The Returning Officer shall cause notice of election to be sent by post to all financial members no later than March 15 each year.

2. FORM OF NOTICE

The notice of election shall include:

- a list of all positions to be elected, namely:
 - President
 - Vice President
 - Secretary
 - Treasurer
 - Ordinary Committee Members (5)
 - Returning Officer
 - Assistant Returning Officer
- a nomination form; the date by which nominations must be received (in accordance with clause 21(2) of the Constitution, this date is 14 April);
- the means by which the nomination form may be lodged;
- description of the format for a policy statement.

3. POLICY STATEMENT

A person nominated for election may include with the nomination a policy statement of up to 200 words. This word limit shall not include sections of the statement stating in point form the nominee's name, personal details and positions held on, or by appointment of, the AUUG Management Committee and chapters. Policy statements exceeding the word limit shall be truncated at the word limit when included in the ballot information. The Returning Officer may edit policy statements to improve readability, such edits being limited to spelling, punctuation and capitalisation corrections and spacing modifications. Use of the UNIX `wc` program shall be accepted as an accurate way to count words.

4. RECEIPT OF NOMINATIONS

In accordance with clause 21(2) of the Constitution, nominations shall be received by the Secretary up until April 14. A nomination shall be deemed to have been received by the due date if one of the following is satisfied: it is delivered by post to AUUG Inc.'s Post Box, the AUUG Secretariat's Post Box or the AUUG Secretariat's street address no later than 2 business days after April 14 and is postmarked no later than 12 midday on April 14; it is delivered by hand to the Secretary or the AUUG Inc. Secretariat no later than 5 pm on April 14; it is transmitted by facsimile to the Secretary or the AUUG Inc. Secretariat no later than 5 pm on April 14.

5. REQUIREMENT FOR A BALLOT AND DUE DATE

In accordance with clause 21(5), no later than May 1, the Secretary shall advise the Returning Officer of all valid nominations received; and if a ballot is required, shall advise the Returning Officer of a date no later than May 15 for the ballot for all contested election. In accordance with clause 42(3), the due date for return of ballots shall be 4 weeks after the date advised above.

6. FORM OF BALLOT PAPER

The ballot paper shall contain: details of all positions for which the number of nominations exactly equals the number of positions to be filled; for each position for which a ballot is required, the names of all persons seeking election to that position, except those already elected to a higher position, with a square immediately to the left, for the elector to place a voting preference; instructions on how to complete the ballot paper; instructions on how to return the ballot paper; a brief description of how the ballot is to be counted. The ballot paper shall not contain any identification of existing office-bearers. The ballot paper shall be accompanied by a copy of all policy statements submitted by all persons nominated, including any persons elected unopposed. These policy statements may be truncated or modified as outlined in 3.

7. METHOD OF VOTING

Voting for each position shall be by optional preferential vote. The number 1 must be placed against the candidate of the elector's first preference, and a number other than 1 against any or all of the other candidates. Preferences shall be determined by the numbers placed against other candidates, which must be strictly monotone ascending to count as preferences. A vote shall be informal if: it does not have the number 1 against exactly one candidate.

8. SECRECY OF BALLOT

The ballot paper shall be accompanied by two envelopes, which may be used by the elector to ensure secrecy. On completion of the ballot paper, the paper may be placed inside the smaller envelope. This envelope is then placed inside a second envelope. The elector must then sign and date the outer envelope, making the following declaration:

I, <insert full name here>, member number <insert member number here>, declare that I am entitled to vote in this election on behalf of the voting member whose membership number is shown above, and no previous ballot has been cast on behalf of this voting member in this election.

9. RETURNING BALLOT

To be considered to have been returned by the due date, the ballot paper together with declaration as above must be returned by one of the following means: it is delivered by post to AUUG Inc.'s Post Box, the AUUG Secretariat's Post Box or the AUUG Secretariat's street address no later than 2 business days after the due date and is postmarked no later than 12 midday on the due date; it is delivered by hand to the Returning Officer or the AUUG Inc. Secretariat no later than 5 pm on the due date.

10. METHOD OF COUNTING

Where there is an election for a single position, the votes shall be counted by the preferential method. Where there is more than one position to be filled, the votes shall be counted by the modified preferential Hare Clark system described in Schedule 1.

11. METHOD OF ELECTION

A person may be elected to only one position. Elections shall be counted in the order of positions described in 2(a). When counting ballots, any person previously elected shall be deemed withdrawn from that election, and all ballot papers shall be implicitly renumbered as though that person was not included.

12. NOTIFICATION OF RESULT

In accordance with clause 42(7) of the Constitution, the Returning Officer shall advise the Secretary in writing of the result no later than fourteen days after the due date. The Returning Officer shall advise all candidates

for election of the result no later than fourteen days after the due date. The Returning Officer shall advise the AUUGN Editor in writing of the result no later than fourteen days after the due date. The AUUGN Editor shall include the results in the first issue of AUUGN published after receiving the results from the Returning Officer.

13. PUBLICATION OF THESE RULES

The Returning Officer shall advise the AUUGN Editor of the current rules, and the AUUGN Editor shall cause the current rules to be published in the first issue of AUUGN published on or after 1 January each year. Where no issue of AUUGN has been posted by February 28 in any calendar year, the Returning Officer shall cause the current rules to be distributed with the notice of election.

14. OCCASIONAL VARIATION FROM THESE RULES

Subject to the Constitution, the Management Committee may authorise occasional variations from these rules. Such variations shall be advised in writing to all members at the next stage in the election process in which information is distributed to members.

15. EXECUTION

Where these rules require the Returning Officer to carry out an action, it shall be valid for the Returning Officer to delegate execution to the Secretariat from time to time employed by the Management Committee.

16. RETENTION OF BALLOT PAPERS

The Secretary shall retain that ballot papers and member declarations (as specified in 8) until the AUUG AGM of the calendar year following the year of the election, unless a general meeting of AUUG directs the Secretary to hold them for a longer period.

Schedule 1

Each ballot paper shall initially have a value of one. The value of each ballot paper shall be allotted to the candidate against whose name appears the lowest number on the paper among those candidates not elected or eliminated. If there is no such candidate (i.e. the ballot paper is exhausted) the ballot paper shall be set aside. A quota shall be calculated by dividing the number of formal votes by one more than the number of positions remaining to be elected, and rounding up to the next whole number. If any candidate is allotted a total value greater than the quota, that candidate shall be declared elected, and the ballot papers allotted to that candidate shall be assigned a new value by multiplying their previous value by the excess of the candidate's vote above the quota divided by the candidate's total vote. This new value shall be truncated (rounded down) to 5 decimal places. Ballot

papers that subsequently have a value of zero shall be set aside. Steps 2 and 3 shall then be repeated. If no candidate is allotted a total value greater than the quota, the candidate who is allotted the lowest total value among those candidates not elected or eliminated shall be eliminated. Steps 2 and 3 shall then be repeated. Where two or more candidates declared elected at the same stage of counting according to Step 4 have an equality of votes, and it is necessary to determine which is deemed elected first, or a candidate is required to be eliminated under Step 5, and two or more candidates have an equally low vote, the Returning Officer shall return to the immediately preceding stage of counting and in the case of candidates elected, deem first elected the candidate with the highest vote at the immediately preceding stage, and in the case where a candidate is to be eliminated, eliminate the candidate with the lowest vote at the immediately preceding stage. Where an equality of votes still exists at the immediately preceding stage, the Returning Officer shall continue proceeding to preceding stages until a result can be determined. In the event that candidates have maintained an equality of votes throughout the entire counting process, the Returning Officer shall determine which candidate is to be determined first elected or to be eliminated by lot in the presence of the Assistant Returning Officer. ♦

AUUG MANAGEMENT COMMITTEE:

Summary of Minutes of Meeting, 11th December 1994

Location: AUUG Business Office, North Sydney.

Present: Phil McCrea, Glenn Huxtable, Frank Crawford, Chris Maltby, Lucy Chubb, Stephen Boucher, Peter Wishart.

Apologies: Michael Paddon, Rick Stevenson.

Guests: Catrina Dwyer.

1. Matters Arising from the Minutes

Brenda is coordinating OSR articles. The first one should be in Feb. edition. It should include a diary of events.

Motion: That Phil Anderson of OzWare Developments be appointed editor of AUUGN, effective in Jan. 1995. Moved: FC/SB. CARRIED.

Motion: That Jagoda Crawford be thanked for her tireless efforts as AUUGN editor. Moved: CM/LC. CARRIED BY ACCLAMATION.

2. Treasurers Report

ACMS has now paid all monies from AUUG94. We probably made \$20-30K from the conference/exhibition (FC to confirm).

We should consider reducing % from tutorials given to OS speakers who get paid their airfares etc.

As at 5/12/94 we had \$102K (cheque acc) + \$25K (cash mgt) + \$37K (int. bearing) = \$164K.

AARNET invoices still need to be paid. AARNET renewals have all been rolled over to Dec. and have now been sent out. There are around 180.

3. Secretary's Report

Perhaps membership renewal notices should show names of all known contacts for corporate members. This might help speed-up renewals.

Discussion on membership cards: they should have an AUUG e-mail address. This e-mail address should have "auug" in the domain name (e.g., auug@auug.org.au). We will investigate credit card style membership cards.

4. AUUG Web Page

There was discussion about the Canberra chapter setting up an Internet system which could support an AUUG Web Page. Canberra chapter would be happy to do this. It was requesting support for disk space from AUUG. It was decided to support this proposal and use the Canberra chapter facilities for the Web page. We should also consider moving the current e-mail aliases (at munnari) to this facility.

5. Brent Chapman Tour

The event should be organised as an AUUG national event. Brent could also do some in-house workshops etc., for large companies. Also approach connect.com.au, OZSERT(?) etc to see if they are interested.

Projected costs: \$300/member, \$450/non-member. Target May 1995. Five slots: Syd, Mel, Bris, Can, Per. Plus one other.

6. Other Issues

6.1. Summer Conferences

All states except Vic and SA have now confirmed a date. Vic has just lost its venue (Clunies-Ross House is moving to Qld), and so date is in doubt.

Lachie can make some dates and notable speakers to go into diaries (plus OSR and Australian).

6.2. Insurance

We have received a quote from GIO for \$5,000,000 public liability insurance for AUUG activities.

Motion: That AUUG change its public liability insurance to GIO. Moved: FC/CM. CARRIED.

7. Next Meetings

The next meetings will be on Monday February 27th 1995 and Friday April 28th 1995, both at the AUUG offices in North Sydney.

Peter Wishart, AUUG Inc. - Secretary ♦

Updated AUUG Regional Contacts, 1994 -1995

Adelaide

Michael Wagner
mhw@syserv.com.au
tel: (08) 212-2800
fax: (08) 231-0321

Brisbane

Greg Birnie
greg@lna.oz.au
tel: (07) 340-2111
fax: (07) 340-2100

Canberra

John Barlow
john.barlow@anu.edu.au
tel: (06) 249-2930
fax: (06) 249-0747

Darwin

Phil Maker
pjm@cs.ntu.edu.au
tel: (089) 466-666
fax: (089) 270-612

Hobart

Steven Bittinger
steven.bittinger@its.utas.edu.au
tel: (002) 207-406
fax: (002) 207-488

Melbourne

David Taylor
davet@vaxc.cc.monash.edu.au
tel: (03) 857 5660

Perth

Glenn Huxtable
glenn@cs.uwa.edu.au
tel: (09) 380-2878
fax: (09) 380-1089

Sydney

Julian Dryden
julian@dwt.csiro.au
tel: (02) 809-9345
fax: (02) 809-9476



CALL FOR PARTICIPATION

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

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

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

This year it is possible that a number of the current committee will not stand again. We need some new faces and ideas on the committee.

Please consider whether you can donate your time and energy to helping make AUUG an even more successful organisation.

The management committee holds office from 1st July to 30th June of the next year. They meet formally for about one day once every two months, in a place convenient to most committee members (usually in Sydney). Reasonable travel costs for the meeting are reimbursed. There is frequent use of e-mail to communicate issues and discuss ideas. While e-mail

access is not required it does help with discussions between meetings. If you do not have e-mail access, do not be deterred from nominating for the committee, I am sure we could work out some effective communication (perhaps by FAX or a guest e-mail account somewhere). As well as attending meetings, committee members should be prepared to take on occasional special project activities (e.g. technical interface between AUUG and AARNET, membership drives, interface with local chapters).

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

Nominations for the election this year close on the 14th of April 1995. A call for nominations and nomination form are included in this issue of AUUGN.

Participation in the election process, either as a nominee, or as an elector, is your way of influencing the directions of AUUG. Please exercise your rights to ensure that the management committee remains representative of the interests of AUUG members. Please do not be deterred if you do not have ready access to the required AUUG members to sign your nomination form. Just contact your local chapter committee or contact me directly (contact details below) and we will organise the necessary signatures for your nomination form.

We look forward to 1995/96 being another year of success for AUUG.

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

Peter Wishart
Secretary - AUUG Inc.

Phone: (06) 261 2997 (W)

FAX: (06) 261 3806

(06) 247 2996 (H)

E-mail: peter.wishart@auug.org.au❖

AUUG Inc. 1995 Annual Elections: CALL FOR NOMINATIONS

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

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

Nominations must be made in writing and must be signed by the nominee and three (3) financial members of AUUG Inc and indicate which position(s) are sought. A sample nomination form is attached.

Nominees must be financial members of AUUG Inc and may nominate for any or all of the above positions. While any member may be nominated to more than one position, no person can be elected to more than one position. Election to positions is determined in the order shown above.

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

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

Nominations must be received by the Secretary by 14th April 1995. They may be lodged by one of the following methods:

(1) by post to:

The Secretary
AUUG Inc.
PO Box 366
Kensington NSW 2033

(must be received no later than 2 business days after April 14th and be postmarked no later than 12 midday on April 14th 1995)

(2) by hand to:

The Secretary (Peter Wishart) OR the AUUG Inc Secretariat no later than 5 pm on April 14th 1995.

(3) by FAX to:

The Secretary (FAX: (06) 261 3806, marked Attn: Peter Wishart) OR the AUUG Inc. Secretariat (FAX: (02) 332 4066) no later than 5 pm on April 14th 1995.❖



Nominee's Consent:

I,

Nominee's Name	AUUG Member No.

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

Signed (1)

Date

Each person may be elected to at most one position, and the ballot for positions shall be determined in the order shown on this nomination form.

We,

Name	AUUG Member No.
(1)	
and	
(2)	
and	
(3)	

being current financial members of AUUG Inc. do hereby nominate

Name:

for the following position(s): *(tick one or more)*

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

Signed (1)

Date

Signed (2)

Date

Signed (3)

Date

~/archive> !AUUGN

AUUGN Vol IV No I

Formatting C

Tim Long

Basser Department of Computer Science
 University of Sydney
 (timl:basservax)

1. Introduction

Every C programmer has strong views on idiom, style and formatting. Unfortunately these views are as idiosyncratic as they are inflexible. In C many semantically distinct constructs have only minor syntactic differences. For human beings formatting is often the only reasonable method of distinguishing them.

2. Object and type declarations

To establish some terminology we present the following example:

```
static unsigned int      stab_segs, stab_size = 1109, ref_counts[MAX_N];
<----base type---->    <--item-> <--item->          <-----item----->
<----first part---->   <-----second part----->
<-----declaration----->
```

A declaration usually has two parts. The first part, which we will call the base type, is a list of storage class specifiers, basic type specifiers and adjectival modifiers of basic types. Some examples of storage classes are static and register. The term storage class has lost much of its original intuitive meaning. For instance the modifier typedef is considered a storage class, but it clearly has nothing to do with storage. Examples of basic types are int, float and enum. Examples of adjectives are long and short.

The second part is a comma separated list of items to be declared and their initialisations. Each of these items includes an identifier, possibly surrounded by *, () or []. Any item may be followed by an = and an initial value.

2.1. Formatting simple declarations

Only one item should be declared per declaration: there should be no comma separated lists. For example:

```
char    *p, c;           /* WRONG */

char    *p;             /* RIGHT */
char    c;              /* RIGHT */
```

The reasons for this are

- (a) all but the first identifier in the WRONG case are hidden and often missed in a quick glance;
- (b) the mixture of types (pointer to character and character in the above example) can cause confusion;
- (c) it is harder to add a comment or initialisation to an item in the WRONG case.

All base types, items and initialisations within a group of declarations should be vertically aligned. For example:

```
char *tape_name = "/dev/rht0" /* WRONG */
unsigned long offset; /* WRONG */
int state = st_idle; /* WRONG */

char          *tape_name      = "/dev/rht0"; /* RIGHT */
unsigned long  offset;        /* RIGHT */
int           state          = st_idle;      /* RIGHT */
```

We can now consider a declaration to have three parts.

- (a) The base type, which is never omitted.
- (b) The item being declared, which may be omitted.
- (c) The initialisation, which will probably be omitted.

It is this three part nature which dominates the layout of simple declarations.

2.2. Complex type definitions

The definition of complex types such as structs, unions and enums should be isolated and typedefed. The definition of a complex type in C is a side effect of its appearance in the base type part of a declaration. To make this clearer, consider the following declarations:

```
enum states          state;
struct point         where;
```

Clearly the enum states and the struct point are base types and state

and where are items. Now consider this (badly formatted) example.

```
enum states (st_idle, st_active)      state;
struct point (int x; int y;)         where;
```

This is equivalent to the first example except that definitions are bound to the identifiers states and point. Notice that the definition of the members of the complex type is part of the base type. Finally it should be noted that it is not necessary to bind the complex type definition to an identifier, as the following example shows:

```
enum (st_idle, st_active)            state;
struct (int x; int y)                where;
```

2.3. Formatting complex type definitions

The complex type declarations in the previous section were in poor style: a new type name should be created for each complex type generated. There are two ways of doing this. This example demonstrates one:

```
typedef enum                          /* RIGHT */
{
    st_idle,
    st_active,
}
states;

typedef struct                          /* RIGHT */
{
    int    x;
    int    y;
}
range;
```

Much of the above formatting will be explained later. The main point is that the enum and struct are not bound to any identifier. A new type name is created to refer to the types as a whole. The declaration of the objects state and where becomes:

```
states      state;      /* RIGHT */
point       where;     /* RIGHT */
```

Unfortunately this method cannot always be used. When a struct or union references itself (in the form of a pointer) the type of the pointer can not be named because its declaration is not complete. In this situation the following variation can be used.

```
typedef struct struct_node    node;
struct struct_node
{
    int    node_value;
    node   *node_link;
};
```

This binds the definition of the structure to the identifier struct_node in order to achieve a forward reference. But the following declaration is also valid (and preferable):

```
typedef struct node    node; /* RIGHT */
struct node
{
    int    node_value;
    node   *node_link;
};
```

Notice that this binds the definition of a structure and a new type to two identifiers, both of which are called node. These identifiers come from logically distinct symbol tables. The structure binding is irrelevant and serves only as a mechanism for the forward definition of the type.

Formatting the member list of a complex type is straightforward. The on curly brace should be placed on a new line directly under the base type. The elements of the member list are indented one tab stop, and the formatting rules are applied recursively. The off curly brace is aligned with its matching one. In the second variation this is followed by the semicolon. But if a type name is being defined, the name is placed on a new line indented one tab stop from the off brace, followed by the semicolon.

There are several justifications for this layout.

- (a) The conceptually independent acts of type definition and storage allocation are separated.
- (b) The indenting and positioning of brackets serves to surround the memberlist declaration with white space, separating it from peripheral activity and placing it where it can be seen and modified. The same arguments apply here as for simple declarations.
- (c) The use of a typedef makes the programmer's intention clear.
- (d) Subsequent declarations become clean and narrow enough for the author to be consistent with vertical alignment.

The following is a trimmed example of large structure declaration. The source fragments comes from an include file. Near the top of this file is found the following block of typedefs:


```

/*
 *      Forward declarations of general purpose data types.
 */
typedef struct cfrag    cfrag;
typedef struct cnode    cnode;
typedef struct ident    ident;
typedef struct xnode    xnode;
typedef union data      data;

```

Although not all of these forward references were necessary all structures and unions were given them in this case for consistency.

The following structure definition was found further down the file along with all the other complex type definitions.

```

struct xnode
{
    union
    {
        xnode    *xu_xnd;
        ident    *xu_id;
    }
    x_left;
    union
    {
        xnode    *xu_xnd;
        cnode    *xu_cnd;
    }
    x_right;
    xnode    *x_type;
    xnodes   x_what;
    data     x_value;
    short    x_flags;
};

```

Typical declarations involving this and related types look something like:

```

register xnode    *x;
register ident    *id;
place            where;

```

3. Function definitions

```

char    *
strcpy(s1, s2)
char    *s1;
char    *s2;
{

```

The above function definition has a useful characteristic. Although the function returns a non int object, its name appears at the start of a line. This both improves readability and lends itself to automated searching methods. The alternative

```
char    *strcpy(s1, s2)
char    *s1;
char    *s2;
{
```

is readable but does not allow an easy distinction between invocations and the definition in an editor search. In general the same rules apply to a function definition as a simple type except that a new line is taken immediately before the identifier.

The leading bracket of the formal parameter list should be placed immediately after the function name. The formal parameters themselves should be placed on the same line with a space after each comma. The closing bracket should be placed hard against the last formal parameter (or the opening bracket if there are no formals). For example:

```
main(argc, argv, env)
main()
```

Declaration of the formal parameters follows, hard against the left margin and obeying the rules of simple declarations.

4. Formatting blocks

Blocks have two parts, surrounded by curly braces. These parts are

- (a) declarations local to this block;
- (b) executable statements.

Where the block is the body of a function the opening curly brace is placed on a line of its own, hard against the left margin. Each time a sub-block is opened the opening curly brace is indented one further tab stop from the level of the enclosing block. The brace always appears on a line of its own. For example:

```
while (i < n)
{
    dothis();
    dothat();
}
```

This positioning of the opening curly bracket is important to

- (a) visually separate the body of the block from surrounding peripheral activity;
- (b) act as a pointer to any flow control construct controlling the block;
- (c) allow a similar visual clue to any controlling expression.

Placing the opening curly brace on the end of the previous line both embeds any controlling expression in blocks of text and leads to special cases when blocks are opened to gain local variables.

The local declarations are started on a new line indented one tab stop from the initial brace. Formatting is as described above. One blank line should be left between the local declarations and the executable statements. If there are no declarations the code should start on a new line immediately after the on brace. For example:

```

{
    char    *p;
    int     i;

    p = "this is a demo";
    {
        i = 0;
        return i;
    }
}

```

The occasional blank line between executable statements is acceptable but should not be over-indulged. The significance of such blank lines is easily lost. Often a block comment is more appropriate (see "Comments").

5. Formatting executable statements

Statements are placed on new lines indented one tab stop from the level of the on and off braces of their surrounding block. It is unacceptable to have more than one statement on one line.

```

i = 0; j = 10;          /* WRONG */
return;                /* WRONG */

i = 0;                 /* RIGHT */
j = 0;                 /* RIGHT */
return;                /* RIGHT */

```

Placing many statements on one line banishes all but the first to oblivion. Although it may be argued that some statements are logically related this is not sufficient justification for the devaluation of statements tacked onto the end of another.

6. Formatting expressions

When an expression forms a complete statement, it should, like any other statement, occupy one or more lines of its own and be indented to the current level. Binary operators should be surrounded by spaces. Unary operators should be placed hard against their operand.

```
* p ++;          /* WRONG */
i=i*10+c-'0';    /* WRONG */

*p++;           /* RIGHT */
i = i * 10 + c - '0'; /* RIGHT */
```

The ternary operators ? and : should also be surrounded by spaces.

When a sub-expression is enclosed in brackets, the first symbol of the sub-expression should be placed hard against the opening bracket. The closing bracket should be placed immediately after the last character of the sub-expression.

```
a = b * ( c - d ); /* WRONG */
a = b * (c - d);   /* RIGHT */
```

Note that the symbols ->, ., and [] which build up primaries (factors) are not considered binary operators in this context. They should not be surrounded by spaces. For example:

```
addr = addrs[ (d >> 3) & 037 ]; /* WRONG */
addr -> csr = 0;                 /* WRONG */

addr = addrs[(d >> 3) & 037]; /* RIGHT */
addr->csr = 0;                 /* RIGHT */
```

The round brackets which surround the arguments of a function call attract no spaces.

```
puts ( "hi\n" ); /* WRONG */
puts("hi\n");   /* RIGHT */
```

Commas, whether used as operators or separators, should be placed hard against the previous symbol and followed by a space.

```
write(2,"whoops\n",7); /* WRONG */
write(2, "whoops\n", 7); /* RIGHT */
```

White space in expressions is useful as much by its lack as its presence. For instance placing spaces in the inside edges of brackets merely spreads out the expression and loses the suggestion of binding. Excessive white space causes inflation and promotes devaluation.

Occasionally expressions become too large to fit on a single line. Breaking at an arbitrary column is distasteful and often unreadable. Rewriting the expression as two, possibly using a temporary, may destroy its conceptual integrity and efficiency. The solution is to reformat the expression over several lines. Consider the following:

```
fprintf
(
    stderr,
    "%s: Could not open %s for reading. %s\n",
    my_name,
    tape_name,
    errno > sys_nerr ? "" : sys_errlist[errno]
)
```

This demonstrates the formatting of the most common cause of long lines, the function call with many arguments. Notice the position of the opening and closing brackets. The actual parameters are aligned vertically one tab stop in from the current level. Each actual parameter occupies a line of its own.

```
if
(
    (id->id_type == NULL)
    ||
    (
        (id->id_type->x_what == xt_arrayof)
        &&
        (item->x_left->x_what == xt_arrayof)
        &&
        (id->id_type->x_subtype == item->x_left->x_subtype)
        &&
        (id->id_type->x_flags & XIS_DIMLESS)
    )
)
```

Here we see another common line length transgressor put in its place. Notice the placement of binary operators and brackets on lines of their own.

The basic message in the above examples is don't be afraid of using more lines to make the expression clear.

7. Formatting flow control constructs

In order to give visual distinction between flow control constructs (such as for and while) and function calls, a small variation in formatting is introduced. A space is used to separate a flow control keyword from any controlling expression. For example:

```
if (p != NULL)
{
    dothis();
    dothat();
}
return p;
```

The space separates the keyword in order to emphasise the flow control dominating the following statement or block.

In view of the above the formatting of for and while statements is straightforward:

```
for (p = root; p != NULL; p = p->next)
    process(p->data);

while ((c = getchar()) != EOF)
    putchar(c);
```

When formatting if statements several alternatives are possible. The simple if statement is again straightforward:

```
if ((fid = open(name, O_READ)) == SYSERROR)
    perror(name);
```

In a simple if-else combination the else keyword should be placed on a line of its own at the same indentation as the if:

```
if (c == '\\')
{
    ...
}
else
{
    ...
}
```

Although these are the only variations of if statements distinguished in the language the author feels that it is often desirable to consider an if-else chain as a flow control construct in its own right. In this case the following layout is acceptable:

```

if (c == '\\')
{
    ...
}
else if (c == '"')
{
    ...
}
else if (c == '\')
{
    ...
}
else
{
    ...
}

```

The formatting of switch statements is simple:

```

switch (pid = fork())
{
    ...
}

```

However the placement of case labels and labels in general often gives trouble. The keyword case should be placed on a line of its own at the same indent level as the controlling switch keyword. A space should separate the word case from the constant expression which is immediately followed by the colon. A blank line should be left above a case label if program flow does not fall through it. For example:

```

switch (pid = fork())
{
case SYSERROR:
    fprintf(stderr, "%s: Could not fork.\n", my_name);
    exit(1);

case 0:
    ...
}

```

Ordinary labels and defaults follow the same rules.

Placing executable statements on the same line as a label (of any sort) is unacceptable since

- (a) the statement is visually hidden by the label;
- (b) it is impossible to be consistent with indenting, there will always be some constant expression too long.

The formatting of do statements is difficult. The intuitive method

is:

```
do
{
    ...
}
while (...);
```

However the duality of the while keyword often leads to confusion, especially if the preceding block is large. To avoid this an arbitrary convention is adopted (as in the case of flow control keywords and function calls). The while keyword should be indented one tab stop from the level of the closing brace:

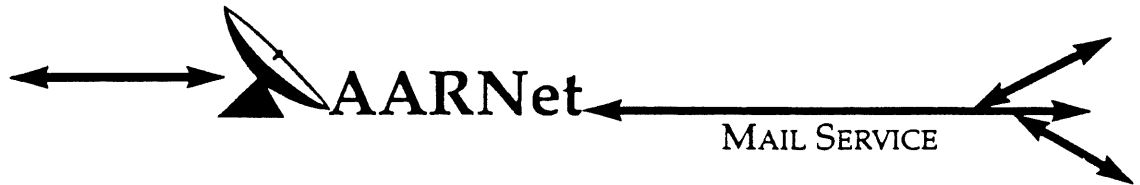
```
do
{
    ...
}
    while (...);
```

8. Comments

Much of this document has concerned itself with formatting aimed at improving readability. The tacit assumption is that readable code is easier to understand than unreadable code. Comments do not improve readability but attempt to directly aid understanding and maintenance.

Comments embedded in code tend to create a dense mass of text. Comments which begin and end on the same line, intermixed with code, should be avoided. It is better to use a few large comments than many smaller ones distributed through the text.

```
/*
 *   This demonstrates the layout of a "block comment". One
 *   comment such as this at the head of a hundred line
 *   function is often more useful than hundreds of two or
 *   three words.
 */
main(argc, argv)
int    argc;
char   *argv[];
{
```

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.



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 Australian Vice-Chancellors' Committee (AVCC) to maintain on my behalf an electronic mail delivery record in the Australian Academic and Research Network (AARNet) allow my organisation to send and receive electronic mail carried across AARNet.

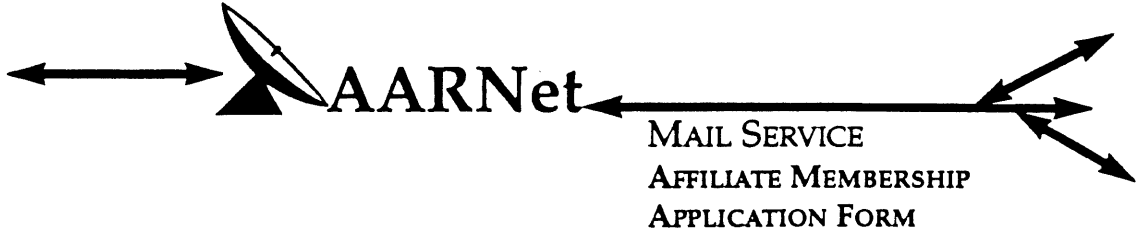
I understand that the AVCC may consult the recorded logs of my organisation's usage of AARNet facilities for 1990, and determine that I am ineligible for registration under terms of the agreement between AVCC and AUUG Inc. I understand that AUUG Inc will invoice my organisation for this service for the 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 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 owned by an AUUG member. Corporation host owners may only claim the member payment if the corporation is an Institutional member of AUUG Inc. My cheque payment of either \$250 or \$600 as appropriate is enclosed with this application.



PLEASE PRINT CLEARLY!

Date: _____

Name of Organisation/Owner: _____

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

Name: _____ Position: _____

on behalf of the organisation named above.

Address: _____

_____ Postcode: _____

Administrative Contact: _____ Title: _____

E-Mail: _____ Phone: () _____

Fax: () _____

Technical Contact: _____ Title: _____

E-Mail: _____ Phone: () _____

Fax: () _____

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

Domain Names Requested: _____

Gateway Addresses: _____

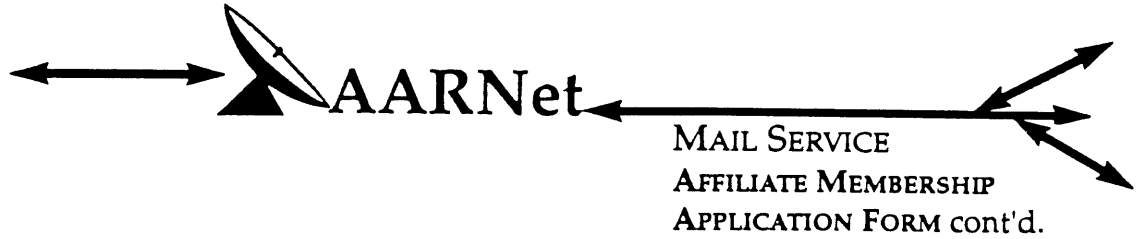
Expected Link Protocol: UUCP SL/IP MHSnet Other: _____

* * * *

Send this page only to:

AUUG Incorporated
PO Box 366
Kensington NSW 2033

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



Note A. Mail Delivery Information

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

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

Example Mail Directory Information request:

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

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

Note B. Getting Connected

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

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

AUUGN Volume 15 Index

Compiled by Stephen Prince
<sp@clcs.com.au>

Papers (by title, with abstracts)

Chubb, Lucy, "Articles From the Australian Newspaper," AUUGN, vol. 15, no. 4, pp. 85-94, AUUG Inc., Sydney, NSW, August 1994.

Michael Paddon -- The Bandwidth Crisis / Roger Fraumann -- Business is War! / Phil McCrea -- Will we see an "Open NT Foundation"? / Brenda Parsons -- System Administration: The Emerging Profession / Margaret Hassell -- CASE Tools / Michael Werner -- Distributed Computing for the Masses / Paul-Michael Agapow -- Passing on C++ / Paul-Michael Agapow -- Brave New Technologies / Michael Paddon -- Take Unix Home to Meet the Family /

Paddon, Michael, "Articles From the Australian Newspaper," AUUGN, vol. 15, no. 3, pp. 71-82, AUUG Inc., Sydney, NSW, June 1994.

Michael Paddon -- What Makes a System Open? / William Eisman -- Client-Server Technology Empowers the User and Provides Corporations with a Competitive Advantage / Glenn Kowack -- EUnet: An Infrastructure-Building Alternative / David Tory -- Future Trends in Computing: The Distributed Computing Environment / Phillip McCrea -- I have seen a glimpse of the future... and the future is good! / Michael Paddon -- Will NT Take the Future By Storm? / Phil McCrea -- A Leopard Really Can Change its Spots / Adrian Booth -- Will COSE create easier-to-run systems? / Frank Crawford -- The Internet - A modern Social Experiment / Michael Paddon -- Open Systems? Or Unix?

Henning, Michael, "C++ - Is it really better C?," AUUGN, vol. 15, no. 4, pp. 54-56, AUUG Inc., Sydney, NSW, August 1994. Email: michi@citr.uq.oz.au

Barton, Phil, "Considerations and Issues when Implementing a Document Management System - Case Study: Prospect Electricity," AUUGN, vol. 15, no. 4, pp. 63-74, AUUG Inc., Sydney, NSW, August 1994.

The purpose of this paper is to examine the components of a Document Management System and address some of the issues involved in the implementation of a Document Management System (DMS). Prospect Electricity will be used as a case study to share some of the experiences gained in what is, and continues to be, a large and complex project.

Lance, Kate, "Crafting a Code of Ethical Conduct," AUUGN, vol. 15, no. 4, pp. 25-35, AUUG Inc., Sydney, NSW, August 1994.
Email: clance@cs.newcastle.edu.au

<unknown>,, "A Dataflow Shell or A Workbench for a Tools-Based Mail System," AUUGN, vol. 15, no. 1, pp. 58-64, AUUG Inc., Sydney, NSW, February 1994.

At the school of CS&E-UNSW we have replaced the traditional mail handler (sendmail) with a more flexible and more understandable tools-based mail system. Any tools-based system requires a workbench to join the tools together, just as the "shell" does for traditional UNIX tools. This paper descusses a new command language interpreter which was developed to integrate the mail tools. It is different from the traditional shell in that it focuses on the flow of data rather than the flow of control.

Booth, Adrian, "Delving into the UNIX kernel," AUUGN, vol. 15, no. 3, pp. 65-68, AUUG Inc., Sydney, NSW, June 1994. Email: abcc@dialix.oz.au

Why would anyone care about what happens "under the hood", inside the UNIX kernel? This paper tries to answer this question, and the briefly describe ways in which the operation of the kernel can be examined.

Booth, Adrian, "The Electronic Interviews," AUUGN, vol. 15, no. 2, pp. 45-47, AUUG Inc., Sydney, NSW, April 1994.

This, the second "Electronic Interview", is with Piers Lauder of Sydney University, who has been involved with UNIX from its earliest days in Australia. Piers was the programme chair for AUUG93, and also a director of both Message Handling Systems (MHS) and BSDI (Australia).

McGrath, Tony, "Firewalls and Proxy Servers - Access and Security on the Internet," AUUGN, vol. 15, no. 3, pp. 54-64, AUUG Inc., Sydney, NSW, June 1994.

One of the more prevalent activities on the Internet in recent times has been the creation of firewall systems to restrict access between systems. A number of different facilities have been created to help network administrators to create the level of security they need and also allow limited access to services for trusted users. One of these systems is SOCKS, which comprises an authentication daemon and a limited API that allows BSD socket programs to be created that can access services through the SOCKS daemon. One of the major restrictions of the SOCKS package is the implicit assumption that only a single firewall system exists between the client system and the server system. An assumption that is no longer valid given the proliferation of firewall systems. This paper will address the issues that arise when multiple levels of firewalls are established on the network, but it is still required to permit client applications to access services that exist on systems that have multiple firewalls between them. A limited implementation of this facility, called PROXY, is also described.

Purdue, David, "Getting Performance Data from UNIX," AUUGN, vol. 15, no. 5, pp. 102-109, AUUG Inc., Sydney, NSW, October 1994.
Email: davidp@knowledge.com.au

Unix has a reputation, especially among mainframe fanatics, of not allowing easy access to performance related information, such as how much of certain resources (eg: cpu or memory) are being used. In fact there are a number of standard UNIX commands that will report this information. This paper

examines some of these commands, and briefly discusses two tools, Patrol and SAS/CPE, that aid in making sense of their results.

Carve, Charles, "**I Couldn't Survive without my Internet Connection**," AUUGN, vol. 15, no. 3, pp. 43-53, AUUG Inc., Sydney, NSW, June 1994.
Email: cmc@sydney.unidata.oz.au

This paper is a case study in the technical and financial benefits gained by a software company by implementing an Internet connection for electronic mail. Following the connection of the Sydney subsidiary and the US head office to the Internet, many benefits have been gained in the areas of technical support, dissemination of information, distribution of software patches and documentation. Facsimile charges and telephone usage has dropped giving financial and time saving benefits. Unidata Australasia has been able to take advantage of the abundant software available through Internet, in particular the GNU packages from the Free Software Foundation (FSF), and information and resources available through the USENET news system. This paper is aimed at people who don't yet have an Internet connection, summarizing the benefits to justify its installation. Experienced Internet users may also benefit by finding new uses for this powerful resource.

Vassos, Tom, "**The Internet -- On-Ramp To The Information Superhighway**," AUUGN, vol. 15, no. 5, pp. 62-64, AUUG Inc., Sydney, NSW, October 1994.
Email: vassos@vnet.ibm.com

Crawford, Frank, "**The Internet - How to Connect in Australia: Service Provider Choices and Costs**," AUUGN, vol. 15, no. 6, pp. 56-70, AUUG Inc., Sydney, NSW, December 1994.
Email: Frank.Crawford@ansto.gov.au

Kilgour, Alistair C., "**Letter to Dr I. Johnstone**," AUUGN, vol. 15, no. 4, pp. 52-53, AUUG Inc., Sydney, NSW, August 1994.
**** Reprint from AUUGN v1n4 ****

Elz, Robert, "**Letter to Ian Johnstone**," AUUGN, vol. 15, no. 3, p. 42, AUUG Inc., Sydney, NSW, June 1994.
**** Reprint from AUUGN v1n2 ****

Elz, Robert, "**Letter to W.N. Joy**," AUUGN, vol. 15, no. 6, pp. 52-55, AUUG Inc., Sydney, NSW, December 1994.
**** Reprint from AUUGN v2n6 ****

Robert explains to Bill what he has been doing to VMUNIX since he gave him the tape with his latest system at the end of June.

Crawford, Frank, "**Managing Mailing Lists with Majordomo**," AUUGN, vol. 15, no. 4, pp. 57-62, AUUG Inc., Sydney, NSW, August 1994.
Email: frank@atom.ansto.gov.au

Mailing lists are becoming an increasingly important way of distributing information over the Internet and the related networks, eg: ACSnet. One reason for this is that, unlike network news, it is very focused medium, with all people on the list having an interest in the topic. This paper will discuss the requirements for a mailing list, the administration of a simple

list and the administration of a complex list. Tools such as Majordomo, which help with complex mailing lists, will be discussed.

Bishop, Craig, "**No You Can't Have Root (How Software Vendors Can Help System Administrators)**," AUUGN, vol. 15, no. 5, pp. 77-85, AUUG Inc., Sydney, NSW, October 1994. Email: Craig.Bishop@BarwonWater.Vic.Gov.Au

As Unix matures more and more software vendors are entering the Unix Market. The potential for profits within this large, expanding market is very attractive. Combine that with the advantages of portability and reusability on multiple platforms, the desire to get products to market becomes very strong. This paper addresses the concerns of Unix System Administrators who have to install and integrate these software packages. These concerns have increased with the influx of software vendors into the Unix market. Where do they come from? What experience do they have? What do they know about the philosophy and history of Unix? If they follow the guidelines in this paper, I probably won't care.

McClelland, Geoff, "**Open Systems Architectural Deployment**," AUUGN, vol. 15, no. 5, pp. 86-101, AUUG Inc., Sydney, NSW, October 1994.

Longo, Joseph, "**Perkin-Elmer Supports Unix Or A Philosophical View of Unix in the Commercial Environment or Will Success Spoil Unix?**," AUUGN, vol. 15, no. 1, pp. 54-55, AUUG Inc., Sydney, NSW, February 1994.
**** Reprint from AUUGN v4n3 ****

McDonald, Bryan, "**Please Unsubscribe Me from this List...**," AUUGN, vol. 15, no. 4, p. 24, AUUG Inc., Sydney, NSW, August 1994. Email: bigmac@usernix.org

<unknown>,, "**Q&A: The Pentium FDIV Bug**," AUUGN, vol. 15, no. 6, p. 30, AUUG Inc., Sydney, NSW, December 1994.

Maltby, Chris, "**Review of the AIR 2.0 TCP Software Packages for Windows**," AUUGN, vol. 15, no. 4, pp. 41-42, AUUG Inc., Sydney, NSW, August 1994.
Email: chris@softway.sw.oz.au

Pascoe, Sam, "**Sam's Serial Listening Device (Mark 3)**," AUUGN, vol. 15, no. 1, pp. 56-57, AUUG Inc., Sydney, NSW, February 1994.
Email: samp@DIALix.oz.au

This article describes a (relatively) simple hardware tool which was designed to permit serial traffic passing between a user's terminal and a host system to be observed on a separate screen or printer.

Booth, Adrian, "**Securing an Internet Connection**," AUUGN, vol. 15, no. 6, pp. 71-73, AUUG Inc., Sydney, NSW, December 1994. Email: abcc@dialix.oz.au

This is a summary of the talk I gave at WAUG's November meeting. It discusses an Internet connection established by a large organisation, with particular emphasis on the security aspects of the connection.

Merrilees, Scott, "Stashbase," AUUGN, vol. 15, no. 1, pp. 65-67, AUUG Inc., Sydney, NSW, February 1994. Email: Sm@bhpeso.oz.au

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

Chubb, Peter, "Testing Tools under UNIX," AUUGN, vol. 15, no. 2, pp. 54-63, AUUG Inc., Sydney, NSW, April 1994.

Testing is divided into two parts: the hard part and the tedious part. The hard part is determining what to test; the tedious part is doing it. There are at least three packages I know about for helping with the tedious bit (one developed at Softway, one developed by the FSF, called DejaGnu, and some expensive ones). This talk will describe a tool developed at Softway to automate running tedious tests and compare it with DejaGnu.

<unknown>,, "Top Ten New Intel Slogans for the PENTIUM," AUUGN, vol. 15, no. 6, p. 48, AUUG Inc., Sydney, NSW, December 1994.

Crawford, Frank, "A Twisty Little Maze of Machine Descriptions or An Overview of GCC Porting," AUUGN, vol. 15, no. 2, pp. 64- 71, AUUG Inc., Sydney, NSW, April 1994. Email: frank@atom.ansto.gov.au

The Gnu C Compiler (gcc) from the Free Software Foundation (FSF) has taken over from the Portable C Compiler as the common C Compiler available across many different platforms. In many ways it is also following in the footsteps of UNIX, by being developed and extended by the user community and by being adopted for use on new systems. However, unlike UNIX, there are restrictions on the redistribution of gcc for commercial purposes, so its use is limited to either being a secondary compiler or on free (or at least very cheap) systems. Despite the wide spread use of gcc, very few people understand how it achieves its portability. This paper will outline what is required to port gcc to a new architecture, using, as an example, a recent port carried out by the author to the IBM System/370 architecture.

Booth, Adrian, "Unix Tricks & Traps," AUUGN, vol. 15, no. 2, pp. 91-92, AUUG Inc., Sydney, NSW, April 1994. Email: abcc@dialix.oz.au

Jackson, Janet, "Unix Tricks & Traps," AUUGN, vol. 15, no. 4, pp. 83-84, AUUG Inc., Sydney, NSW, August 1994. Email: jackson@cwr.uwa.edu.au

Graham Jenkins - Handling FTPmail

Brewer, Carl, "Unix Tricks & Traps," AUUGN, vol. 15, no. 6, p. 97, AUUG Inc., Sydney, NSW, December 1994. Email: carl@oversteer.library.uwa.edu.au

The Perils of pico (or, why you should use vi!).

Jackson, Janet, "Unix Tricks & Traps," AUUGN, vol. 15, no. 1, pp. 78-79, AUUG Inc., Sydney, NSW, February 1994. Email: jackson@cwr.uwa.edu.au

Working in two places at once & Perl to the rescue again.

Jackson, Janet, "Unix Tricks and Traps," AUUGN, vol. 15, no. 5, pp. 110-111, AUUG Inc., Sydney, NSW, October 1994. Email: janet@dialix.oz.au

Years after having the idea, I've finally gotten around to writing the following program, which I call listhosts. It is for sites that use NIS (Yellow Pages) netgroups to group their hosts into categories. Unlike Unix groups, netgroups may be nested, so the program has to recurse.

Jackson, Janet, "User Support Mailbox," AUUGN, vol. 15, no. 3, pp. 69-70, AUUG Inc., Sydney, NSW, June 1994. Email: jackson@cwr.uwa.edu.au

McKusick, M. Kirk, "What's New in the 4.4BSD User Code," AUUGN, vol. 15, no. 2, pp. 48-53, AUUG Inc., Sydney, NSW, April 1994.

Papers (by author)

Author/Title	Issue	Page	Month
<unknown>			
Q&A: The Pentium FDIV Bug	6	30	December
Top Ten New Intel Slogans for the PENTIUM	6	48	December
A Dataflow Shell or A Workbench for a Tools-Based Mail System	1	58-64	February
Barton, Phil			
Considerations and Issues when Implementing a Document Management System - Case Study: Prospect Electricity	4	63-74	August
Bishop, Craig			
No You Can't Have Root (How Software Vendors Can Help System Administrators)	5	77-85	October
Booth, Adrian			
The Electronic Interviews	2	45-47	April
Unix Tricks & Traps	2	91-92	April

Author/Title	Issue	Page	Month
Securing an Internet Connection	6	71-73	December
Delving into the UNIX kernel	3	65-68	June

Brewer, Carl

Unix Tricks & Traps	6	97	December
---------------------	---	----	----------

Carve, Charles

I Couldn't Survive without my Internet Connection	3	43-53	June
---	---	-------	------

Chubb, Lucy

Articles From the Australian Newspaper	4	85-94	August
--	---	-------	--------

Chubb, Peter

Testing Tools under UNIX	2	54-63	April
--------------------------	---	-------	-------

Crawford, Frank

A Twisty Little Maze of Machine Descriptions or An Overview of GCC Porting	2	64-71	April
Managing Mailing Lists with Majordomo	4	57-62	August
The Internet - How to Connect in Australia: Service Provider Choices and Costs	6	56-70	December

Elz, Robert

Letter to W.N. Joy	6	52-55	December
Letter to Ian Johnstone	3	42	June

Henning, Michael

C++ - Is it really better C?	4	54-56	August
------------------------------	---	-------	--------

Jackson, Janet

Unix Tricks & Traps	4	83-84	August
Unix Tricks & Traps	1	78-79	February
User Support Mailbox	3	69-70	June
Unix Tricks and Traps	5	110-11	October

Kilgour, Alistair

Letter to Dr I. Johnstone	4	52-53	August
---------------------------	---	-------	--------

Author/Title	Issue	Page	Month
--------------	-------	------	-------

Lance, Kate

Crafting a Code of Ethical Conduct	4	25-35	August
------------------------------------	---	-------	--------

Longo, Joseph

Perkin-Elmer Supports Unix Or A Philosophical View of Unix in the Commercial Environment or Will Success Spoil Unix?	1	54-55	February
--	---	-------	----------

Maltby, Chris

Review of the AIR 2.0 TCP Software Packages for Windows	4	41-42	August
---	---	-------	--------

McClelland, Geoff

Open Systems Architectural Deployment	5	86-101	October
---------------------------------------	---	--------	---------

McDonald, Bryan

Please Unsubscribe Me from this List...	4	24	August
---	---	----	--------

McGrath, Tony

Firewalls and Proxy Servers - Access and Security on the Internet	3	54-64	June
---	---	-------	------

McKusick, M. Kirk

What's New in the 4.4BSD User Code	2	48-53	April
------------------------------------	---	-------	-------

Merrilees, Scott

Stashbase	1	65-67	February
-----------	---	-------	----------

Paddon, Michael

Articles From the Australian Newspaper	3	71-82	June
--	---	-------	------

Pascoe, Sam

Sam's Serial Listening Device (Mark 3)	1	56-57	June
--	---	-------	------

Purdue, David

Getting Performance Data from UNIX	5	102-09	October
------------------------------------	---	--------	---------

Author/Title	Issue	Page	Month
--------------	-------	------	-------

Vassos, Tom

The Internet -- On-Ramp To The Information Superhighway	5	62-64	October
---	---	-------	---------

Reports (by author)

Author/Title	Issue	Page	Month
--------------	-------	------	-------

<unknown>

1994 Membership Survey	5	27-28	October
------------------------	---	-------	---------

Booth, Adrian

WAUG Meeting Review	2	25	April
1994 Perth AUUG Summer Conference Overview	2	26-27	April
WAUG Meeting Reviews	4	23	August
WAUG Meeting Review	3	25	August

Brandsma, Mitchell

Linus Does Perth	6	14-15	December
------------------	---	-------	----------

D'Cruze, Patrick

A Summary of Linus Torvalds' Presentation at WAUG	6	13	December
---	---	----	----------

Dauids, Enno

AUUG-Vic Chapter Activities	4	21	August
AUUG-Vic Chapter Activities	3	23	June
AUUG-Vic Chapter Activities	5	54-55	October

Davie, Peter

AUUG Canberra Chapter	4	19	August
AUUG Canberra Chapter	3	21	June

Griffiths, Don

WAUG Meeting Review	6	39	December
---------------------	---	----	----------

Author/Title	Issue	Page	Month
--------------	-------	------	-------

Jackson, Janet

Impressions of the 1994 Perth AUUG Summer Technical Conference	2	28-29	April
From the Western Front	2	24	April
From the Western Front	4	22	August
From the Western Front	6	38	December
WAUG Meeting Review	1	36	February
From the Western Front	3	24	June
Impression of AUUG'94	5	29-32	October
From the Western Front	5	57	October

Jones, David

A Report on LISA '93	3	39-41	June
----------------------	---	-------	------

Lobb, Brenda

Conference Comment: UniForum NZ'94	3	26-27	June
------------------------------------	---	-------	------

Nicol, Stuart C.

AUUG Incorporated Financial Report for 1993/1994	5	12-26	October
--	---	-------	---------

Parsons, Brenda

AUUG NSW Chapter	4	19	August
AUUG NSW Chapter	3	21	June

Prince, Stephen

Update on AUUG Inc. - Victorian Chapter Activities	2	23	April
--	---	----	-------

Rose, Greg

Report on '94 Winter Usenix	1	39-41	February
-----------------------------	---	-------	----------

White, Mark

AUUG Queensland Chapter Update - July Meeting Report	4	18	August
AUUG Queensland Chapter Update	3	20	June

Book Reviews (by book authors)**Albitz, Paul and Cricket Liu***DNS and BIND in a Nutshell*

Reviewer: Lawrie Brown

Issue: 1

Andreasson, Torbjorn and Jan Skansholm*Getting Started with UNIX and X*

Reviewer: Lawrie Brown

Issue: 6

Angell, David and Brent Heslop*The Elements of E-mail Style*

Reviewer: Dave Horsfall

Issue: 6

Associates, O'Reilly &*The X Resource: Issue 9 - Proceedings 8th Annual X**Technical Conference*

Reviewer: J. Wright

Issue: 2

Bergman, Phyllis Eve and Sally A. Browning*UNIX System V Performance Management*

Reviewer: Mathew Lim

Issue: 1

Bloomer, John*Power Programming with RPC*

Reviewer: Ian Crakanthorp

Issue: 2

**Boykin, Joseph, David Kirschen, Alan Langerman,
and Susan Loverso***Programming Under Mach*

Reviewer: Peter Chubb

Issue: 6

Costales, Bryan*Sendmail*

Reviewer: Lucy Chubb

Issue: 1

Dyk, John A. Van*Network Administration UNIX SVR4.2**(Administration Series)*

Reviewer: Brenda Parsons

Issue: 1

Galvin, Silberschatz et. al.*Operating System Concepts - Fourth Edition*

Reviewer: David Hughes

Issue: 3

Garfinkel, Simson and Gene Spafford*Practical Unix Security*

Reviewer: Chris Dale

Issue: 1

Goodheart, Berny and James Cox*The Magic Garden Explained: The Internals of UNIX
System V Release 4*

Reviewer: Peter Chubb

Issue: 1

**Goodman, Michele Ann, Manoj Goyal, and
Robert Massoudi***Solaris Porting Guide*

Reviewer: David Purdue

Issue: 6

Heller, Dan and Paula M. Ferguson*Motif Programming Manual, Second Edition
(Volume Six A)*

Reviewer: David Hughes

Issue: 5

Inc., The Santa Cruz Operation*SCO Open Desktop/SCO Open Server User's Guide*

Reviewer: Lindsay Trewin

Issue: 4

Kerrigan, James F.*Migrating to Fortran 90*

Reviewer: Michael Werner

Issue: 3

Krol, Ed*The Whole Internet, User's Guide and Catalog,
Second Edition*

Reviewer: Phil McCrea

Issue: 5

Lunde, Ken*Understanding Japanese Information Processing*

Reviewer: Greg Doherty

Issue: 2

Mui, Linda and Eric Pearce*X Window System Administrator's Guide for X11 Release 4
and Release 5*

Reviewer: Gordon Turner

Issue: 5

Nye, Adrian*The X Resource: Issue 9 - Proceedings 8th Annual X
Technical Conference*

Reviewer: Michael Werner

Issue: 2

Ousterhout, John K*Tcl and the Tk Toolkit*

Reviewer: Greg Black

Issue: 5

Press, SCO*SCO Open Desktop/SCO Open Server System
Administrator's Guide*

Reviewer: Craig Macbride

Issue: 3

Reichard, Kevin and Eric F Johnson

UNIX in Plain English
Reviewer: Adrian Booth
Issue: 5

Schimmel, Curt

UNIX Systems for Modern Architectures: Symmetric Multiprocessing and Caching for Kernel Programmers
Reviewer: Peter Chubb
Issue: 6

Schwartz, Randal L.

Learning Perl
Reviewer: Jagoda Crawford
Issue: 1

Spuler, David A.

C++ and C Debugging, Testing and Reliability
Reviewer: Greg Rose
Issue: 4

Stevens, W. Richard

Advanced Programming in the UNIX Environment
Reviewer: David Hughes
Issue: 3

Stevens, W. Richard

TCP/IP Illustrated, Volume 1 - The Protocols
Reviewer: Michi Henning
Issue: 4

Stevens, W. Richard

TCP/IP Illustrated, Volume 1 - The Protocols
Reviewer: Danny Lee
Issue: 4

SunSoft

Solaris Application Developer's Guide
Reviewer: Adrian Booth
Issue: 2

SunSoft

Solaris Desktop Integration Guide
Reviewer: Catherine Allen
Issue: 4

Taylor, Dave

Teach Yourself UNIX in a Week
Reviewer: Jon Wright
Issue: 5

Todino, Grace, John Strang, and Jerry Peek

Learning the UNIX Operating System - 3rd Edition
Reviewer: Greg Black
Issue: 4

Todino, Tim O'Reilly, Grace

Managing UUCP and Usenet 10th Edition
Reviewer: Brenda Parsons
Issue: 2

AUUG Institutional Members

as at 17/02/95

- A. Goninan & Co. Limited
 AAll
 Aberfoyle Resource Limited
 ACAY Network Computing Pty.Ltd.
 Actrol Parts
 Adept Software
 Advanced Software Engineering
 Alcatel Australia
 Amalgamated Television Services
 Amdahl Australia Pty Ltd
 Amdahl Pacific Services
 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 Airlines Limited
 Australian Archives
 Australian Bureau of Statistics
 Australian Centre for Remote Sensing (ACRES)
 Australian Defence Industries Ltd.
 Australian Electoral Commission
 Australian Film Television and Radio School
 Australian Information Processing Centre
 Pty. Ltd.
 Australian Museum
 Australian National Audit Office
 Australian National University
 Australian Submarine Corporation
 Australian Taxation Office
 Australian Technology Resources (ACT) Pty. Ltd.
 Australian Technology Resources Pty. Ltd.
 Australian Technology Resources (WA) Pty. Ltd.
 Australian Tourist Commission
 Australian Wool Research & Promotion
 Organisation
 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, Buckridge & 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
 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.
 Computech Pty. Ltd.
 Computer De Tokyo Corporation
 Computer Law Corporation
 Computer Sciences of Australia Pty. Ltd.
 Computer Software Packages
 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
 CSIRO Division of Wool 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 Lands, Housing & Local
 Government
 Department of Planning & Development
 Department of Public Administration
 Department of State Services
 Department of the Treasury
 Dept. of Industrial Relations, Employment,
 Training & Further Education
 Dept. of Lands, Housing & Local Government
 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 Australia
 GEC Alsthom Information Technology
 GEC Marconi Systems Ltd.
 Geelong & District Water Board
 Genasys II Pty. Ltd.
 General Automation Pty. Ltd.
 GIO Australia
 Golden Casket Office
 Golden Circle Limited
 Great Barrier Reef Marine Park Authority
 Gribbles Pathology
 Haltek Pty. Ltd.
 Hamersley Iron
 Hannan Group Computer Services
 Hansen Corporation Pty. Ltd.
 Heath Insurance
 Hermes Precisa Australia Pty. Ltd.
 Hitachi Data Systems
 Hollywood Private Hospital
 Honeywell Australia Ltd.
 Honeywell Ltd.
 Hong Kong Jockey Club Systems (Australia)
 Pty. Ltd.
 I.P.S Radio & Space Services
 IBA Healthcare Pty. Ltd.
 IBM Australia Ltd.

AUUG Institutional Members

Iconix Pty. Ltd.
Ideas International Pty. Ltd.
Independent Systems Integrators
Informatel Online
Information Technology Consultants
Information Technology Services Branch,
Dept of Public Administration
Informed Technology
Insession Labs Pty. Ltd.
Insurance & Superannuation Commission
Integration Design Pty. Ltd.
Intelligent Network Development
International Imaging Systems
Internode Systems Pty. Ltd.
ISR Group Ltd.
ISSC Australia Ltd.
James Cook University of North Queensland
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 Technology Services
Memtec Limited
Mentor Technologies Pty. Ltd.
Mercedes-Benz (Australia)
Message Handling Systems
Metal Trades Industry Association
Minenco Pty. Ltd.
Mirjana Salinovic (Logica)
Mitsubishi Motors Australia Ltd.
Mitsui Computer Limited
Moldflow Pty. Ltd
Motorola Communications Australia
Motorola Computer Systems
MPA International Pty. Ltd.
MPA Systems
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
Northern Territory University
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.
Ochre Development
Office of State Revenue
Office of the Director of Public Prosecutions
ONA
Open Software Associates Ltd.
Open Technology Pty Ltd
Opentec Pty Ltd
OPSM
OSIX Pty. Ltd.
OzWare Developments Pty. Ltd.
Pacific Semiconductor Pty. Ltd.
Pacific Star Communications
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, Design
& Survey Branch
QLD Electricity Commission
Quality Bakers Pty. Ltd.
Quality By Design Pty. Ltd.
Redland Shire Council
Rehabilitation Tasmania
Renison Golffields Consolidated Ltd.
RGC Minerals Sands, Divisional Office
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. Catherine's School
St. Gregory's Armanan School
St. John of God Health Care System
St. Vincent's Private Hospital
Stallion Technologies Pty. Ltd.
Standards Australia
State Revenue Office
State Super (SSIMC)
Steelmark Eagle & Globe
Sterling Software
Storage Technology of Australia
Strategic Information Technologies Pty. Ltd.
Sunburst Regency Foods Pty. Ltd.
Sydney Electricity
Sydney Electricity - Level 6
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
Tattersall Sweep Consultation
Technical Software Services
TechNIX Consulting Group International
Telecom (Applied Research & Development)
Telecom Australia
Telecom Australia Corporate Customer
Telecom Network Engineering Computer
Support Services
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
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 Limited
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
University of Western Australia
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
Work Health Authority
Workstations Plus
XEDOC Software Development Pty. Ltd.
Zircon Systems Pty. Ltd.
Zurich Australian Insurance

Notification of Change

You can help us! If you have changed your mailing address, phone, title, or any other contact information, please keep us updated. Complete the following information and either fax it to the AUUG Membership Secretary on (02) 332-4066 or post it to:

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



(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):	PLEASE PRINT YOUR NEW CONTACT INFORM
Name/Contact: _____	Name/Contact: _____
Position/Title: _____	Position/Title: _____
Company: _____	Company: _____
Address: _____	Address: _____
Postcode _____	
Tel: BH _____ AH _____	Tel: BH _____
Fax: BH _____ AH _____	Fax: BH _____
email address: _____	email address: _____

AUUG Secretariat Use

Date: _____

Initial: _____

Date processed: _____

Membership # _____



MEMBERSHIP APPLICATION

INDIVIDUAL



UNIX® AND OPEN SYSTEMS USERS

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

**REPLY PAID 66, AUUG MEMBERSHIP SECRETARY,
P.O. BOX 366, KENSINGTON, NSW 2033, AUSTRALIA
Tel: +61 2 361-5994 or 1 800 625 655 • Fax: +61 2 332-4066**

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

NOTE: Please do not send purchase orders - perhaps your purchasing department will consider this form to be an invoice. Foreign applicants please send a bank draft drawn on an Australian bank.

INDIVIDUAL OR STUDENT MEMBERSHIP:

I, _____ do hereby apply for:

- Renewal/New membership of AUUG \$ 90.00
- Renewal/New Student membership \$ 25.00 (please complete certification portion)
- International air mail \$ 60.00

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

TOTAL REMITTED:

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

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: _____

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

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

Signature _____

Title _____

Date _____

Over for Institutional Membership

Please charge \$ _____ to my

Bankcard, Visa, Mastercard

Account number: _____

Expiry date: _____

Name on card: _____

Signature: _____

AUUG Secretariat Use

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

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

MEMBERSHIP APPLICATION

INSTITUTIONAL



To apply for AUUG membership, complete this form and return it with payment in Australian Dollars to:
**REPLY PAID 66, AUUG MEMBERSHIP SECRETARY,
 P.O. BOX 366, KENSINGTON, NSW 2033, AUSTRALIA**
 Tel: +61 2 361-5994 or 1 800 625 655 • Fax: +61 2 332-4066

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

NOTE: Please do not send purchase orders - perhaps your purchasing department will consider this form to be an invoice. Foreign applicants please send a bank draft drawn on an Australian bank.

We, _____
(Company Name)

do hereby apply for:

Renewal/New* Inst. membership of AUUG \$350.00
 International air mail \$120.00

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

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

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

Signed _____ Date _____
 Title _____

INSTITUTIONAL MEMBER DETAILS:

To be completed by institutional members only.

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

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

1st Rep. _____

Position/Title _____

Address _____

Bus. Tel: _____ Bus. Fax: _____

e-mail Address _____

Local Chapter Pref. _____

2nd Rep. _____

Position/Title _____

Address _____

Bus. Tel: _____ Bus. Fax: _____

e-mail address _____

Local Chapter Pref. _____

Please charge \$ _____ to my
 Bankcard, Visa, Mastercard
 Account number: _____
 Expiry date: _____
 Name on card: _____
 Signature: _____

AUUG Secretariat Use

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

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

