AUUGN

The Journal of AUUG Inc.

Volume 21 • Number 3
September 2000

Features:

News:

Regulars:

ISSN 1035-7521

Print post approved by Australia Post - PP2391500002

AUUG Membership and General Correspondence

The AUUG Secretary
PO Box 366
Kensington NSW 2033
Telephone: 02 8824 9511
or 1800 625 655 (Toll-Free)
Facsimile: 02 8824 9522
Email: auug@auug.org.au

AUUG Management Committee

Email: auugctee@auug.org.au

President:
David Purdue
David.Purdue@auug.org.au
iPlanet e-commerce solutions
The Tea House
Level 1, 28 Clarendon Street
South Melbourne VIC 3205

Vice-President:
Malcolm Caldwell
Malcolm.Caldwell@auug.org.au
Northern Territory University
Casuarina Campus
Darwin NT 0909

Secretary:
Michael Paddon
Michael.Paddon@auug.org.au
eSec Ltd
245 Racecourse Road
Flemington VIC 3031

Treasurer:
Luigi Cantoni
Luigi.Cantoni@auug.org.au
STM
PO Box 51
North Perth WA 6906

Committee Members:
Sarah Bolderoff
Sarah.Bolderoff@auug.org.au
University of South Australia
School of Computer and Information Science
Room F2-65,
Mawson Lakes Campus SA 5095

Alan Cowie
Alan.Cowie@auug.org.au

Greg Lehey
Greg.Lehey@auug.org.au
Linuxcare Inc.
PO Box 460
Echunga SA 5153

Peter Gray
Peter.Gray@auug.org.au
Information Technology Services
University of Wollongong
Wollongong NSW 2522

David Newall
David.Newall@auug.org.au
Tellurian Pty Ltd.
272 Prospect Road
Prospect SA 5082

AUUG Business Manager

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

Editorial

Günther Feuereisen
gunther@gfh.com.au

It had to happen eventually. It is with (some) sadness that I write my last Editorial. This will be my last issue of AUUGN as Editor.

Increasingly over the last 12 months, I have struggled with putting AUUGN together, due to more and more external commitments, and finally I realised earlier this year, that I just didn't have the time anymore. It was time to concede that I couldn't keep this up.

But, standing in the wings, willing to take the helm was Con Zymaris, our "Open Source Lucky Dip" Sub-Editor, who will be taking over from the next issue.

I'd like to say thanks to all of my Sub-Editors (Past and Present) for all their help. I've been at the helm since late 1996, and I've had the chance to meet and work with a great group of people.

To all of you who contributed, and dropped me a note with thoughts and ideas, thank you!

Bye, and best wishes ..

günther

❖

Thanks to our Sponsors:

AUREMA

TELLURIAN

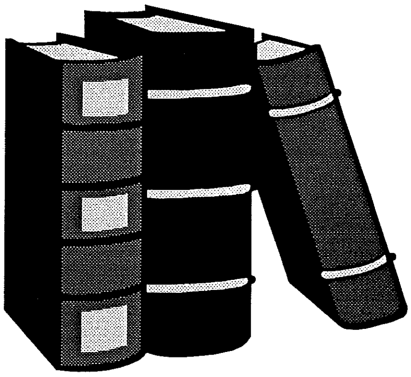
Book Reviews Sub-Editor wanted:

Our resident Book Reviews Sub-Editor, Mark Neely, will be stepping down from his current position due to other commitments.

If you are interested in coordinating the Book Reviews for AUUGN, then this is a wonderful opportunity to join the AUUGN Editorial team.

Drop us a note at:

auugn@auug.org.au



Contribution Deadlines for AUUGN in 2000/2001

Volume 21 • Number 4 – November 2000:	**October 17th, 2000**
Volume 22 • Number 1 – March 2001:	**February 17th, 2001**
Volume 22 • Number 2 – June 2001:	**May 17th, 2001**
Volume 22 • Number 3 – August 2001:	**July 17th, 2001**

AUUGN Editorial Committee

The AUUGN Editorial Committee can be reached by sending email to:
auugn@auug.org.au

Or to the following address:
AUUGN Editor
PO Box 366
Kensington NSW 2033

Editor:
Günther Feuereisen

Sub-Editors:
Frank Crawford
Mark Neely
Jerry Vochteloo
Con Zymaris

Public Relations and Marketing:
Elizabeth Carroll

AUUGN Submission Guidelines

Submission guidelines for AUUGN contributions can be obtained from the AUUG World Wide Web site at:
www.auug.org.au

Alternately, send email 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 please contact the AUUG Secretariat, or write to:
AUUG Inc.
Back Issues Department
PO Box 366
Kensington NSW 2033

Conference Proceedings

A limited number of copies of the Conference Proceedings from previous AUUG Conferences are still available. Contact the AUUG Secretariat for details.

Mailing Lists

Enquiries regarding the purchase of the AUUGN mailing list should be directed to the AUUG Secretariat.

Disclaimer

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

Copyright Information

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

President's Column

David Purdue
David.Purdue@auug.org.au

> **security** (si-'kyur-&-tE), noun, **1.** the quality or state of being secure: as (a) freedom from danger (b) freedom from fear or anxiety (c) freedom from the prospect of being laid off (job security) – *Merriam-Webster's College Dictionary*

I will be writing1 about security, but as this is a President's Column I will not be giving you a new procedure for locking hackers out of your system, but rather talking about ideas, approaches and responsibilities, with a few pertinent examples.

But I want to start by making a bold statement:

The Personal Computer was the worst thing to happen to computing.
In the good old days it took heaps of training before you could use a computer – chances were that you could not use one at all unless you had built it yourself. Computers were only used by those who understood them and were qualified to use them.

But Personal Computing means that any idiot can use computers. There are books devoted to the idiots who use computers. This means that "ease of use" becomes a priority, and this leads us to any number of pitfalls – but I shall return to this theme...

Let's look at an example, which I have borrowed from New Scientist.

The year is 2005, and Feed The World, Inc., release their latest genetically engineered grain. It will grow in any soil, it reseeds itself, and is resistant to 90% of known pests. We can grow it in the deserts of Africa and no one need starve again.

It also has one other feature – as we learn about the genes that resist the other 10% of pests, Feed The World, Inc. can modify the grain's DNA by releasing a virus into the crop.

How soon will it be before we discover that not only can Feed The World, Inc. modify the DNA, but natural viruses can as well? How long before malicious viruses from competing genetic labs are released? How long before we have protests in the streets (a lâ Montreal) and Greenpeace is breaking down the doors at Feed The World, Inc.?

The point is that we do not accept this behaviour in our food – so why do we accept it in our computer operating systems?

The answer is ease of use. If all I know about my computer is that there is a problem, and I am not a geek, then I want it fixed as easily as possible. So I point my browser at the Microsoft web site and automatically download and apply a patch. In fact I was offered an Office 2000 patch that way while I was preparing this column.

So we can see, and I think we have all experienced, that there is a trade off between convenience and security. Trivial example: it is easier to log in when you have a null password.
We are also seeing that there is a trade-off between features and security. New features are more marketable than bug fixes, but new features also imply a larger code base and hence a harder job of establishing and maintaining security. But the market demands more features – as a marketer, I must keep up!

Viruses only exist because programmable devices communicate. In the beginning that communication was the exchange of floppy disks. Now it is the instantaneous exchange of email via the Internet.

.
Guess what! Increasingly we find that more to have features implies making a device programmable.

Take the next generation of mobile phones. They will provide more and more generic communications functions. To speed time to market, and to ensure new features can be added in the field, they are programmable.

Could we see a "Melissa" or "I LOVE YOU" for mobiles – one that arrives on your phone then instantly sends itself via SMS message or email to everyone in your address book? The phone makers say that this is an unlikely scenario, but our experience is that if an attack *can* happen it *will* happen.

What about the humble Palm Pilot? I bought one recently and it is a great tool – but every time I talk to a fellow Palm owner they say, "Hey, let me beam you this great piece of software!" Surely this is a mechanism for virus propagation.

As an aside – the IS department of one multinational sent a message to all employees along the lines, "There is a virus that will arrive in a message with the subject, 'I love you.' If you see such a message please assume that nobody loves you and delete it immediately."

Scott McNealy stood up at Java One this year and pointed out to the assembled masses that "Melissa" and "I LOVE YOU" are not Internet viruses, they are Microsoft Outlook viruses. Fair enough.

But he went on to assert that if only we all used Java we would be immune from such viruses.

1 This column is a transcript of the footnote talk given by David Purdue at AUUG2K.

Well, I'm sorry, Scott, but it's not that simple – the Java Virtual Machine is a programmable device with network access, and so the possibility exists for a virus to be written. And maybe the Java security model means that this virus can not harm your data, but it could certainly cause a denial of service attack, and in these days of electronic commerce that can cost you just as much.

One final note on viruses – Microsoft released an Outlook patch that stops "Melissa" and :I LOVE YOU" cold. It blocks attachments of certain types (.bat, .exe, .vbs – it blocks based on extension rather than content), it stops programs accessing the address book and blocks scripting. According to a report in *Network World*, the user community hated the patch because it removed functionality and removed convenience!

Let's turn our attention to the universal Internet security panacea – the firewall. If we look carefully, we will see that they don't always do what we think they do.

On Sunday (at the AUUG2K tutorials) I sat down with someone who had hooked his laptop to the ANU network and had come up against the ANU firewall policy. The policy prevented the download of software (executables, even gzip files) and apparently images (GIF, JPEG) could only be downloaded during library open hours – go figure!2

However the firewall did allow SSH through – good, a nice secure protocol for secure access to secure systems. But if you use the SSH magic properly you can encapsulate other protocols, and one thing you can encapsulate is PPPoE. Now all bets are off: PPP over SSH from laptop to friendly machine outside the firewall, a bit of routing magic and you can run any protocol you like from the laptop to the Internet with no filtering.

So – it is obvious that the ANU firewall operators are kittens, they are weak. I am the Bastard Operator From Hell, and all my firewall allows is valid HTML passed over HTTP – because for some strange reason the staff and students insist on using the web. They think it some sort of research tool.

A few days later, my logs show this conversation:

Client browser (student linux box) to web server (somewhere on the net):

```
GET login%3A%20
```

Server to client:

```
Content-Type:text/html
<HTML>
<HEAD>
<TITLE>A Hack</TITLE>
</HEAD>
<BODY>
<P>root</P>
</BODY>
</HTML>
```

Client to server:

```
GET Password%3A%20
```

Server to client:

```
Content-Type:text/html
<HTML>
<HEAD>
<TITLE>A Hack</TITLE>
</HEAD>
<BODY>
<P>3blindmice</P>
</BODY>
</HTML>
```

Client to server:

```
GET Welcome%20to%20my%20 machine.%0a
```

This looks something like the client offering the HTTP server a remote login.

How could this happen?

Well, just point your browser to http://www.disgruntled-employee.org and we will send you the software, which will connect back to http://hack.disgruntled-employee.org. Sure the firewall maintainer can make this very difficult – he can block cookies and ensure the http connection is dropped after each request, so there is no state preserved between transactions. But this means that legitimate users of the web are going to have worse performance and are not going to be able to reach sites they may want to look at.

Is this too far fetched? Well the powers that be want to make it even easier.

A new protocol called SOAP (Simple Object Access Protocol) is in the W3C standards track. SOAP is a system independent remote procedure call mechanism that represents objects as XML and passes them back and forth over HTTP. As the nice folks at Microsoft say, "Currently developers struggle to make their distributed applications work across the Internet when firewalls get in the way... Since SOAP relied on HTTP as the transport mechanism, and most firewalls allow HTTP to pass through, you'll have no problem invoking SOAP endpoints from either side of a firewall."

One final illustration – where do we place or misplace trust? I am one of about two million

[2] One of the ANU network engineers who attended this talk pointed out that the "firewall" was just a router that ensured HTTP traffic went through a proxy, and that the "image during library hours" policy was motivated by the desire to reduce network costs by avoiding large downloads that incur volume charges. I contend that any set of devices that attempts to enforce a network policy is a firewall, that this firewall was enforcing a financial rather than a security policy, and that it failed to do so.

people who run SETI at home. This is a piece of software you download that uses the idle time on your CPU to analyse signals from space looking for E.T. phoning home. The combined might of all those CPU's mean that the project effectively has access to a 14 TeraFLOP supercomputer – that is 14 million million operations per second. But am I really searching for E.T.? To preserve the scientific integrity of the project you can only run an official binary that you download from Berkeley, and they do not give access to source. So, as far as I know, I could be cracking RSA keys for the NSA. But hey, man, these guys are from Berkeley, they wouldn't do that to us!

So where does that leave us? The conclusion is that there are no security absolutes. All we have is risk mitigation – if I want to do this thing I must accept that these other things may happen. If I want to be part of society, I must accept that not every member of society is a nice as I am.

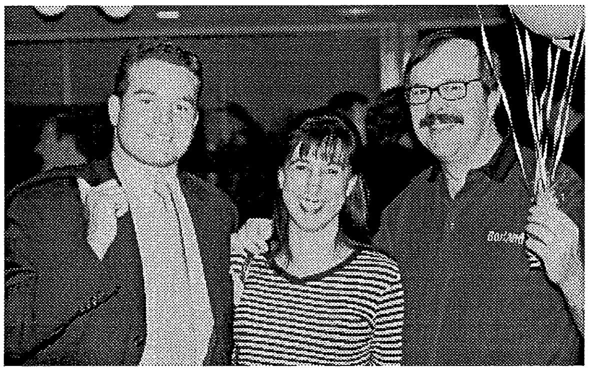
As IT professionals we have a lot of responsibility – we must educate our users on benefits vs. risk in what they do, and we must ensure that our applications allow users to make sensible and informed benefit vs. risk decisions. I recognise that this is a tremendous challenge.

Am I saying we should all be security experts? No, but we do need to be security aware and we need to know our own limitations. If you need help, ask for it.

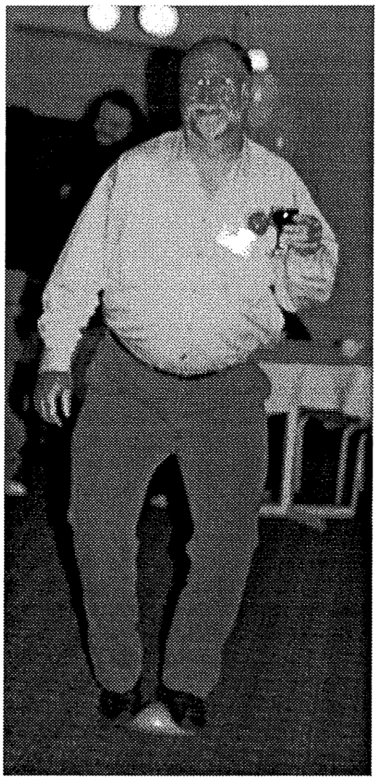
❖

Images from AUUG2K

Photos: David Purdue
David.Purde@auug.org.au
Captions: Elizabeth Carroll
busmgr@auug.org.au

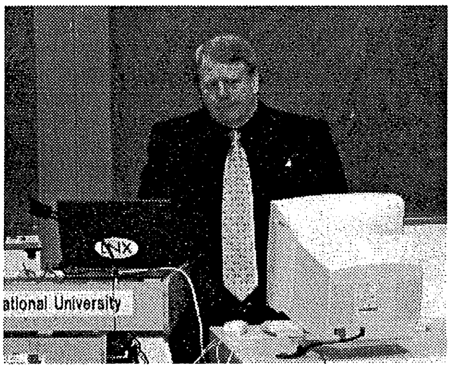


Liz Carroll with our sponsors from Borland - Cocktail evening

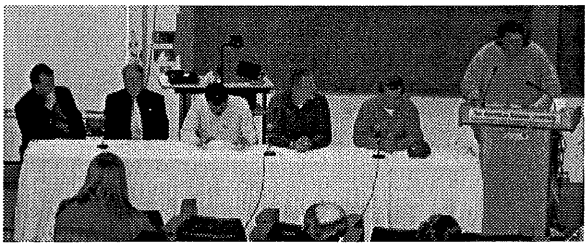


Greg Rose demonstrates how to balance a glass of wine while standing on a balloon?!!!

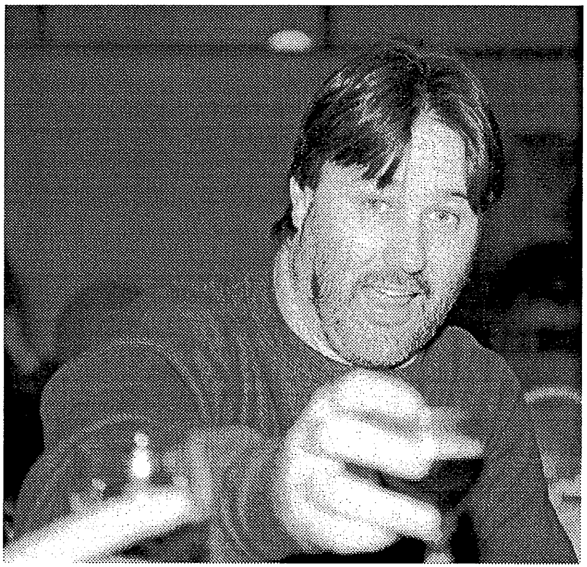
Adrian and Susie Close show off the latest in hair accessories!



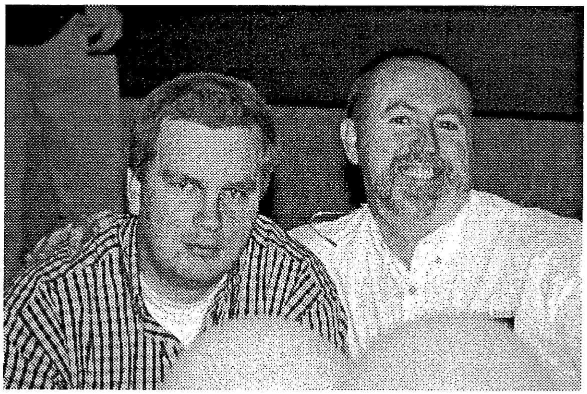
John Terpstra, Turbolinux – delivers his keynote presentation



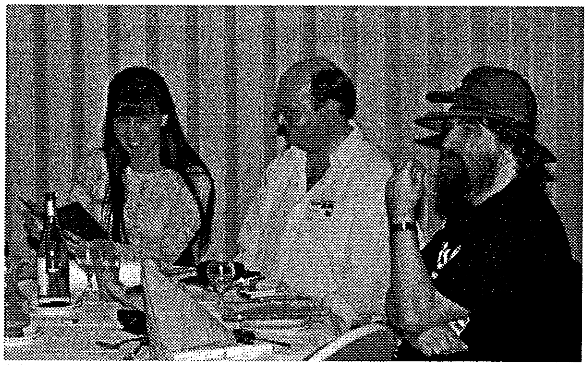
"The Panel" - Anthony Rumble with the Linux Panel



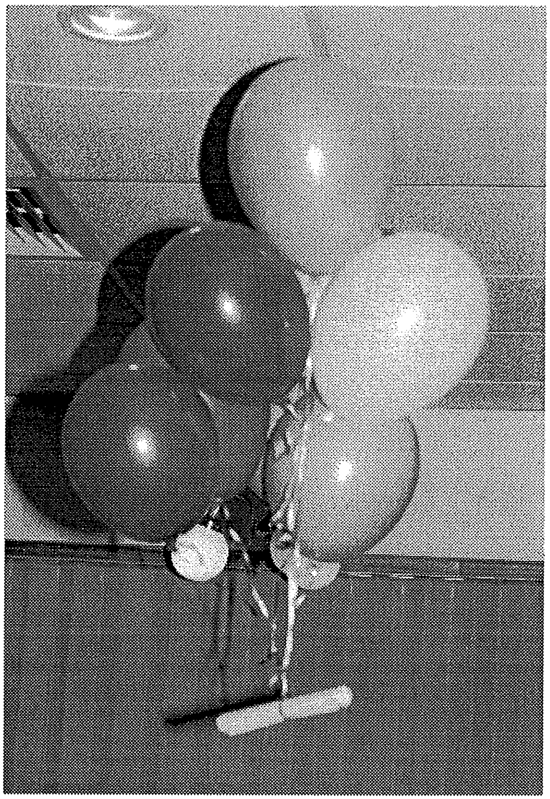
Michael Paddon - gets into the Conference spirit



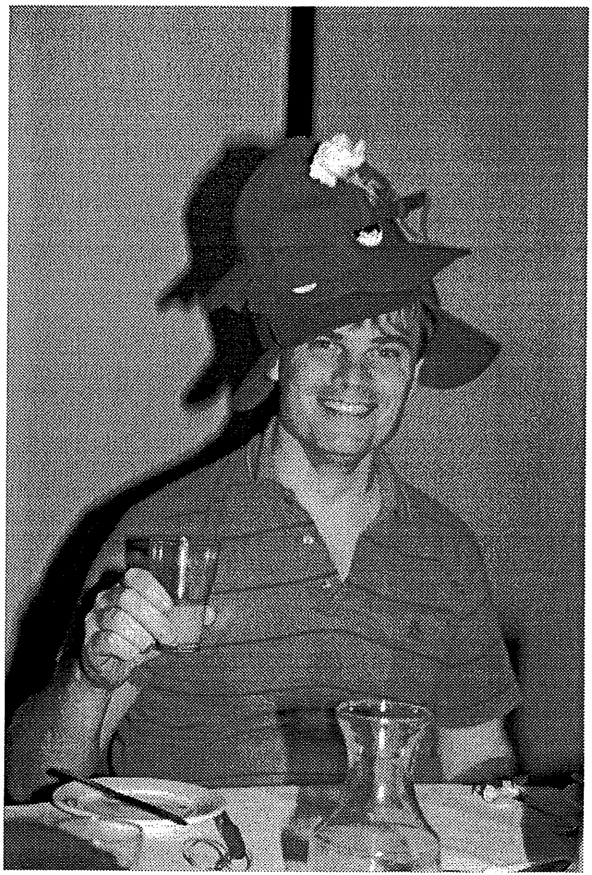
Mark White and Andrew McRae
unwind at the Conference Dinner



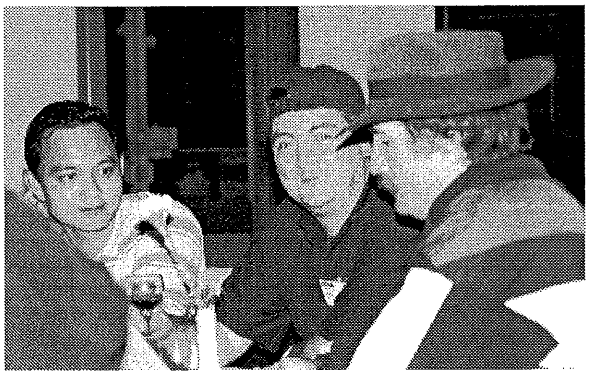
Liz Carroll, Lawrie Brown and Michael Lightfoot check
out the wine offerings at the Conference Dinner



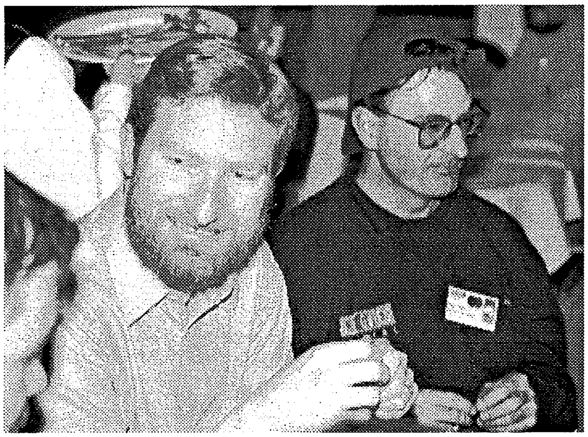
Another packet wings it's way across
the Conference Dinner



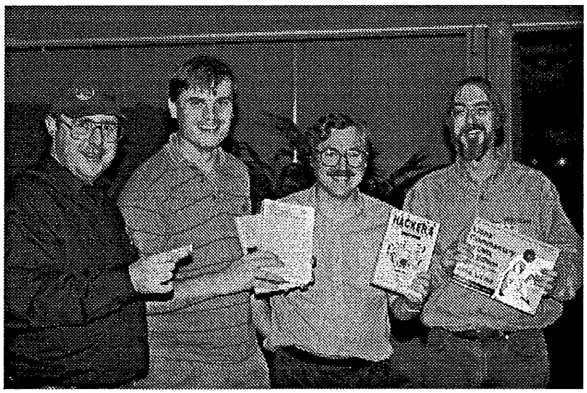
Andrew Tridgell, with one too many Red Hats



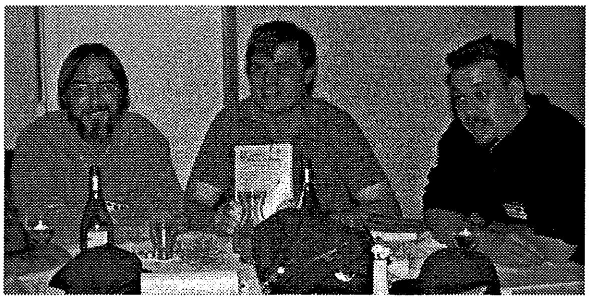
Euan Pryde - the Birthday Boy – celebrates at the Conference Dinner



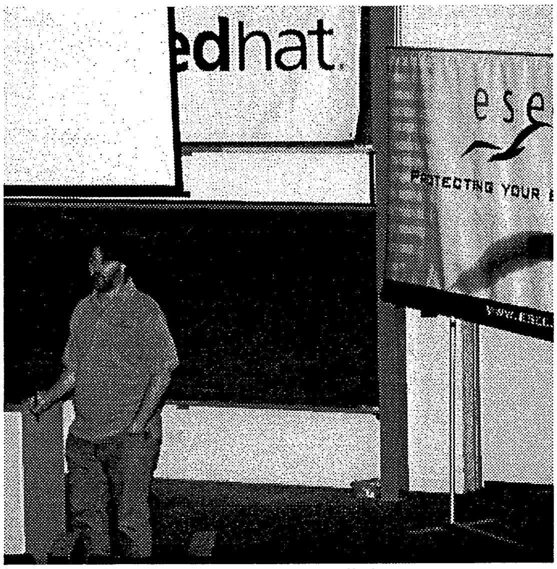
Peter Gray shows us how a "Red Hat" red hat should be worn.



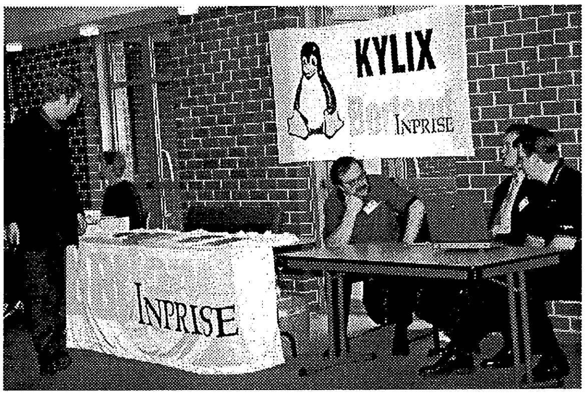
David Purdue (Auctioneer for the night!) - with Linuxcare, the new owners of the 'Unix Sacred Objects'



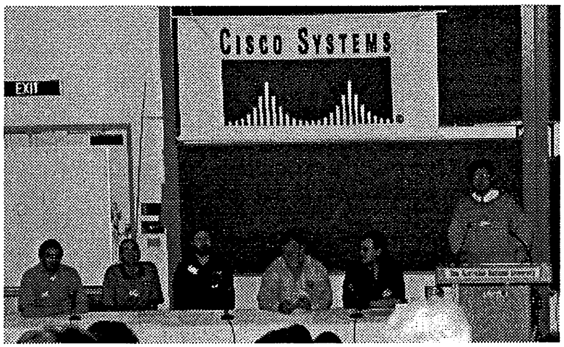
Our Linuxcare team check out their purchase!



Michael Paddon, eSec, enlightens us on "The Art of Keeping Secrets"



The Linux Penguin oversees our sponsors from Borland Inprise



'The Panel' - Part II - headed up by David Purdue

❖

Upcoming AUUG Events

Security Symposium

The AUUG Security Symposium will be held in Melbourne on:

3 November 2000

The purpose of this event is to exchange ideas on the improvement of the security for the systems and networks we manage.

✧ ✧ ✧

AOSS2

The second Australian Open Source Symposium will be held in Adelaide on:

25 November 2000

The purpose of this event is to bring together the Australian Open Source community on an annual basis.

✧ ✧ ✧

AUUG2001

Our annual conference will be held next year in Sydney, back in its traditional September timeslot:

23-28 September 2001

✧ ✧ ✧

Sponsorship Opportunities:

If you are interested in sponsoring any of these events, please contact the AUUG Business Manager, Elizabeth Carroll on:

Telephone: 02 8824 9511
or 1800 625 655 (Toll-Free)

or by email: `busmgr@auug.org.au`

❖

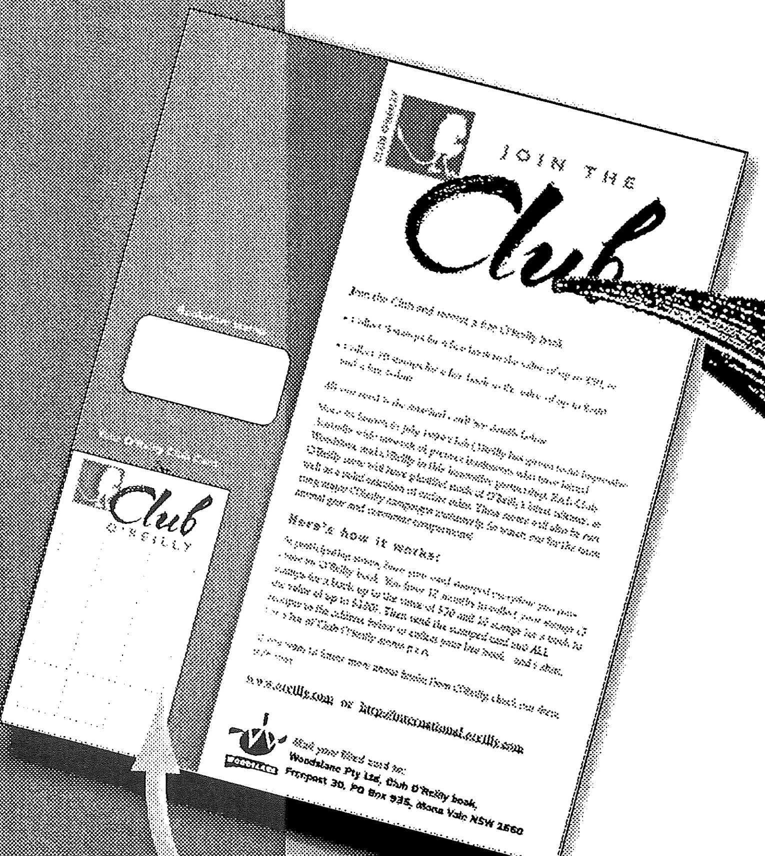
CLUB O'REILLY

JOIN THE *Club*

JOIN CLUB O'REILLY. BUY 5 O'REILLY BOOKS, ANI GET A FREE BOOK TO THE VALUE OF UP TO $50.

OR BUY 10 BOOKS FOR A FREE BOOK VALUED AT UP TO $100.

If your card is missing, please e-ma
Woodslane (sabinepa@woodslane.com) t
receive your Club O'Reilly member card


JOIN THE
Club
Here's how it works:
www.oreilly.com or http://international.oreilly.com
Mail your filled card to:
Woodslane Pty Ltd, Club O'Reilly book,
Freepost 30, PO Box 935, Mona Vale NSW 2660
Club
O'REILLY

Just collect 10 stamps here and keep your receipts

WOODSLANE

Ring Woodslane customer
service foy your nearest
Club O'Reilly store

1800 803 443

O'Reilly publications are distributed throughout Australia and New Zealand by Woodslane.

O'REILLY

5 points for running an Installfest

Sarah Bolderoff
sara@cs.unisa.edu.au
&
Richard Russell
rrussell@deh.sa.gov.au

{ Editor's Note: recently LinuxSA held a Linux Installfest, where local LUG members got together to help new Linux users install and configure Linux. Here follows the definitive guide to running an Installfest in your local area. }

1. Organisation

- Plan ahead
- Consider legal issues and insurance
- Have a queuing/registration system for installees
- Have a method of allocating IP addresses and network information
- Provide a means of identifying installers/helpers eg. T-Shirts
- Have a way of matching installers to installees, some installations may require special skills or knowledge.
- Make sure that the helpers know what is going on
- Have a whiteboard or three -- an area where any problems and people needing help are listed, an area where helpers can list their special skills
- Large numbers of labels, and a well-known (to helpers) labeling system
- SECURITY! A way of ensuring people leave with what they brought, nothing more and nothing less.
- Figure out in advance, a way of dealing with swarms of people

2. Venue

- Lots of space
- Power outlets
- Tables, benches
- Parking facilities
- Space to store equipment
- Convenient location for lugging equipment
- A quiet area for people giving talks

3. Time

- Carefully consider the timing of the event to ensure optimum attendance numbers
- Allow enough time for installations

This can be an issue -- you really need to have a policy on what will and won't be done... it is quite possible to spend three days setting up a Linux system and teaching newbies how to use it... you really want to get new systems to a state where they can connect to the Internet, and then tell people to subscribe to the mailing list. Otherwise people stay forever...

- Plan the date in advance so that you have enough time to advertise the event and venue, allow 2 weeks.
- Make sure t-shirts, posters, website, and whatever other publicity material is ready... Contact the press at least a week before.

4. People

- Helpers, installers, techies
- Presenters and people giving talks
- Non techies, someone to person the registration desk
- Installees
- A security person/team

You need to have a roster of sorts, and instructions on how to do these jobs... instructions are particularly important, because you don't want to have to explain things ten times... it's also important to rotate people around a bit so they don't get bored...

5. Equipment

- Power boards, extension cables, you can't have too many... label them all though!
- Hubs, network cables
- Whiteboards, whiteboard markers, normal markers, pens and paper
- Duct tape, masking tape, string
- Coffee, tea, sugar, milk, mugs, teaspoons
- Demo computers
- Floppies, blank and boot floppies
- Linux/BSD CD's
- Labels
- Clue stick (for delegating "CLUE")

  whack

It helps if you have an idea of who is attending. You want a healthy installer:instalee ratio. The more installers, the better.

The other thing is that installers need to be aware that it's OK not to know something, as long as they seek the answer from someone who knows, or from the web, and that it is important for knowledgeable folk to make themselves available whenever possible, rather than sitting there watching an entire RedHat installation...

Michael Davies has released the source code for his on-line Installfest registration web page. The web interface allows installees and installers to pre-register on-line. The requirements are PHP, Postgres and Apache.

Get the code:
`http://users.senet.com.au/~michaeld`

It's licensed under the GPL, so if you use it, please keep the GPL intact. Or else...

❖

Revamping the BSD multiprocessor code

Greg Lehey
grog@lemis.com

[Editor's Note: This is an excerpt of an article which was originally in ``Daemon News'', http://www.daemonnews.org/200008/dadvocate.html *Our thanks to Greg for permission to reproduce this article in AUUGN.]*

This time last year Mindcraft published benchmarks showing that Microsoft NT could outperform Linux in some very specific areas. You may also have noted that nobody in the BSD camp got up and said ``we can do better''. We were pretty sure it would still not have been as good as Microsoft. In this article I'll explain the background and what the FreeBSD project is doing about it.

THE SMP PROBLEM

UNIX was written for single processor machines, and many of the design choices are not only suboptimal for SMP, they're just plain ugly. In particular the synchronization mechanisms don't work well with more than one processor. Briefly:

- The process context, including the upper half of device drivers, doesn't need to protect itself. The kernel is non-preemptive: as long as a process is executing in the kernel, no other process can execute in the kernel. If another process, even with higher priority, becomes runnable while a process is executing kernel code, it will have to wait until the active process leaves the kernel or sleeps.

- Processes protect themselves against the interrupt context, primarily the bottom half of device drivers, by masking interrupts. The original PDP-11 UNIX used the hardware priority levels (numbered 4 to 7), and even today you'll find function calls like `spl4()` and `spl7()` in System V code. BSD changed the names to more descriptive terms like `splbio()`, `splnet()` and `splhigh()`, and also replaced the fixed priorities by interrupt masks in processors which support the concept, but the principle remains the same. It's not always easy to solve the question of which interrupts need to be masked in which context, and one of the interesting observations at this meeting was that as time goes on, the interrupt masks are getting ``blacker'': each `spl()` is masking off more and more bits in the interrupt mask register. This is not good for performance.

- Processes synchronize with each other using the `sleep()` or `tsleep()` calls. Traditional UNIX, including System V, uses `sleep()`, but BSD prefers `tsleep()`, which provides nice strings which `ps(1)` displays to show what the process is waiting for. FreeBSD no longer has a `sleep()` call, while BSD/OS has both, but `sleep()` is deprecated. `tsleep()` is used both for voluntary process synchronization (e.g. send a request to another process and wait until it is finished), and for involuntary synchronization (e.g. wait for a shared resource to become available).

  Processes sleep on a specific address. In many cases, the address in itself has no meaning, and it's probably easier to think of it as a number. When a process sleeps, it is put on a sleep queue. The `wakeup()` function takes the sleep address, walks through the sleep queue, and wakes every process which is sleeping on this address. This can cause massive problems even on single processor machines; UNIX was never really intended to have hundreds of processes waiting on the same resource, and a number of Apache performance problems center around this behaviour. As a partial solution, FreeBSD also has an additional function, `wakeup_one()`, which only wakes the first process it finds on a specific wait queue.

There are a number of reasons why this concept is not a good solution for SMP. Firstly, the simplistic assumption ``nothing else can be executing in the kernel while I am'' falls flat. FreeBSD currently hasn't implemented a solution for this. Instead, we found a way of enforcing this illogical state, the Big Giant Lock (BGL). Any process entering the kernel must first obtain the BGL; if a process executing on another processor has the lock, then the current processor spins (it sits in a tight loop waiting for the lock to become available); it can't even schedule another process to run, because that requires entering the kernel. This method works surprisingly well for compute bound processes, but for a large number of applications, including database and networking, it can give rise to performances which are only a fraction of what the hardware is capable of. This is the background to the success of the Mindcraft benchmark: at the time, Linux was also using this kind of synchronization.

The other issue is with masking interrupts. This is also quite a problem for SMP machines, since it requires masking the interrupts on all processors, which requires an expensive synchronization.

SOLVING THE PROBLEM

There's no quick and easy solution to this synchronization problem. Sun Microsystems has probably spent more effort on a good SMP implementation than anybody else, but it has taken them the best part of 10 years to do so, and only now is their Solaris 2 operating system showing the benefits.

The Linux people started working on improving their SMP support shortly after the Mindcraft results became known, and they have made significant progress. By comparison, in the FreeBSD camp, we have done almost nothing. NetBSD and OpenBSD haven't even released any SMP support at all. Why?

For some time, I have had a theory that the open source model works well for small projects, but it is not optimal for really big undertakings. Even before the Mindcraft incident I had decided that getting good SMP support for BSD would be a proof of this theory. Well, we're on the way to better support now, but the way it happened is rather unexpected.

BSDi TO THE RESCUE

A few months ago, Berkeley Software Design, Inc. (BSDi) and Walnut Creek CDROM merged. At the time of the merger, we had been told that FreeBSD and BSDi's proprietary operating system, BSD/OS, would be merged. It didn't take long for BSDi to announce that this wasn't going to happen, and there was some dissatisfaction as a result. BSDi did agree, however, to let the FreeBSD project merge some BSD/OS code into FreeBSD. In mid-May, BSDi made a snapshot of their development source tree available to the FreeBSD developers.

On the 15th and 16th June we had a meeting of BSDi and FreeBSD developers at Yahoo!'s facility in Sunnyvale CA. Chuck Patterson, BSDi's lead SMP developer, spent Thursday presenting how BSDi implemented SMP in BSD/OS 5.0 (as of yet unreleased). Chuck also briefly explained BSD/OS 4.x's SMP implementation. On Friday we discussed how to incorporate the structures into FreeBSD.

The BSD/OS 4.x SMP implementation is mainly comprised of a giant lock, but with a twist. Whenever a processor tries to acquire the giant lock it can either succeed or fail. If it succeeds, then it's business as usual. However, if the acquisition fails, the processor does not spin on the giant lock (in other words, it doesn't just keep looping until the lock becomes free). Instead, it acquires another lock, the scheduler lock or schedlock, which protects scheduler-related portions of the kernel, and schedules another runnable process, if any. This technique works extremely well for heavy work loads that have less than one CPU worth of system (kernel processing) load. It is very simple, and it achieves good throughput for these workloads.

The meeting concentrated on the BSD/OS 5.0 SMP implementation, which is more complex:

- The BGL remains, but becomes increasingly meaningless. In particular, it is not always necessary to obtain it in order to enter the kernel. The main reason for its existence is to provide a default synchronization mechanism for system components which haven't been converted yet.

- Instead the system protects shared data structures with mutexes. These mutexes replace calls to `tsleep()` when waiting on shared resources (the involuntary process synchronization mentioned above). In contrast to traditional UNIX, mutexes will be used much more frequently in order to protect data structures which were previously implicitly protected by the non-preemptive nature of the kernel. This mechanism replaces calls to `tsleep()` for involuntary context switches.

  Compared with the use of `tsleep()`, mutexes have a number of advantages:

  - Each mutex has its own wait (sleep) queue. When a process releases a mutex, it automatically schedules the next process waiting on the queue. This is more efficient than searching a possibly very long, linear sleep queue. It also avoids the flooding when multiple processes get scheduled, and most of them have to go back to sleep again.

  - Mutexes can be a combination of spin and sleep mutexes: for a resource which may be held only for a very short period of time, even the overhead of sleeping and rescheduling may be higher than waiting in a tight loop. A spin/sleep lock might first wait in a tight loop for 2 microseconds and then sleep if the lock is still not available at that time. This is an issue which Sun has investigated in great detail with Solaris. BSDi has not pursued this yet, though the BSD/OS threading primitives make this an easy extension to add. It's possibly an area for us to investigate once the system is up and limping again.

- Interrupt lockouts (spl()s) go away completely. Instead, interrupt functions use mutexes for synchronization. This means that an interrupt function must be capable of blocking, which is currently impossible. In order to block, the function must have a ``process" context (a stack and a process structure). In particular, this could include kernel threads.

  BSD/OS on Intel currently uses light-weight interrupt threads to process interrupts, while on SPARC it uses normal (``heavyweight") processes. Chuck indicated that the decision to implement light-weight threads initially was probably the wrong one, since it gave rise to a large number of problems, and although the heavyweight process model would give lousy performance, it would probably make it easier to develop the kernel while the light-weight processes were being debugged. There is also the possibility of building a kernel with one or the other support, so that in case of problems during development it would be possible to revert to the heavy-weight processes while searching for the bug.

THE FREEBSD WAY

On the Friday we discussed how to implement this code in FreeBSD. There are a number of things we need to do. During the meeting we didn't get beyond deciding the first couple of things:

- First remove the BGL (currently a spinlock) and replace it with two, maybe three mutexes, one for the scheduler (schedlock), and a blocking mutex for the kernel in place of the BGL. BSD/OS also has an ipending lock for posting interrupts. At the time, we thought it might be a good idea to implement it as well.
- In addition, implement the heavy-weight interrupt processes. These would remain in place while the light-weight threads were being debugged.

PROGRESS WITH FREEBSD SMPNG

Since that meeting, we have made significant progress. As this article went to press in mid-August, We now have implemented these first two steps on Intel single processor machines, and they run stably. Strangely, we didn't find the expected performance decrease; despite a number of debugging tools in the kernel, performance drop was only about 1% instead of the up to 50% we had been fearing.

We have also made progress on Intel SMP machines, but there is still a lot to do before we can run stably with more than one processor.

WHAT ABOUT NETBSD AND OPENBSD?

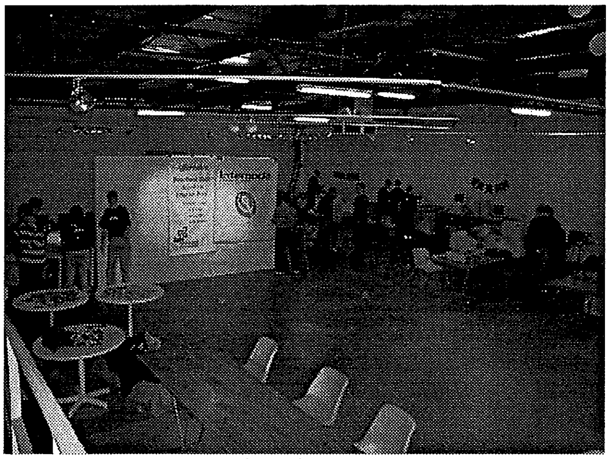
I'm not aware of the state of negotiations between BSDi and the NetBSD and OpenBSD communities. The people I've spoken to at BSDi sounded very interested in supplying the code to NetBSD and OpenBSD as well, and hopefully they'll be able to come to an agreement on how to use the code.

FURTHER READING

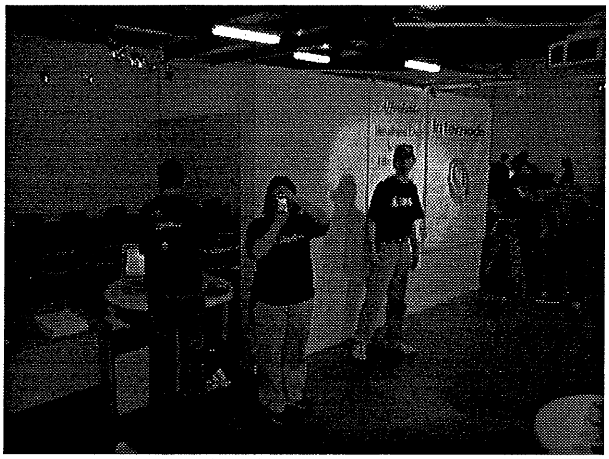
Jason Evans, the project manager, has a web page at `http://people.FreeBSD.org/~jasone/smp/` which tracks the progress of the project. It also contains pointers to a number of facilities, including the source code of the current development.

❖

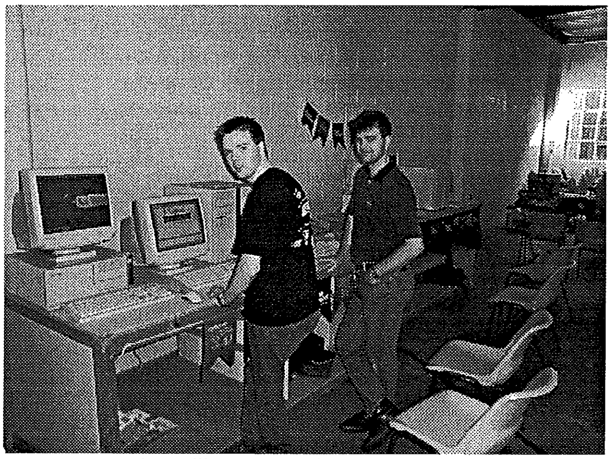
Images from the LinuxSA Installfest



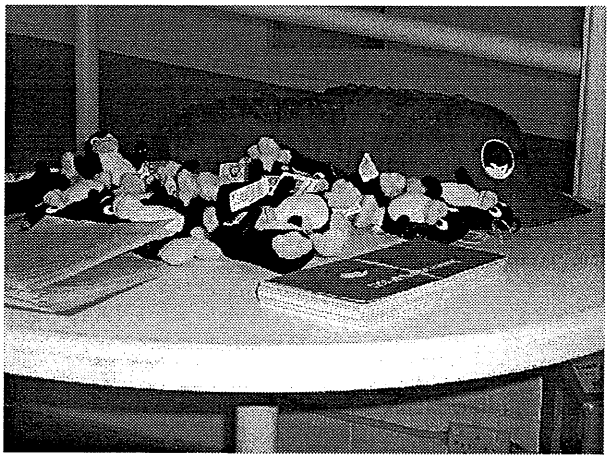
Before the event



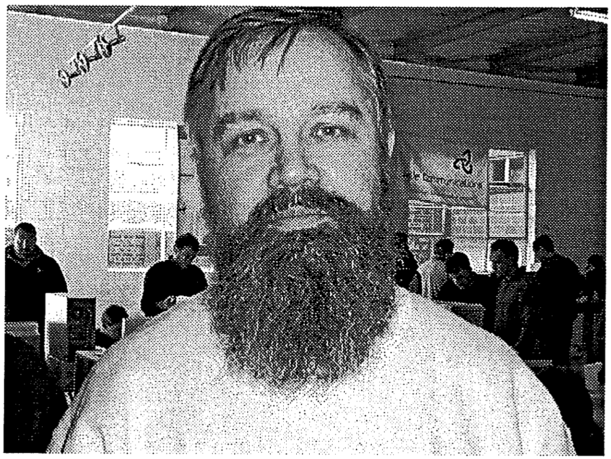
Just foolin' around



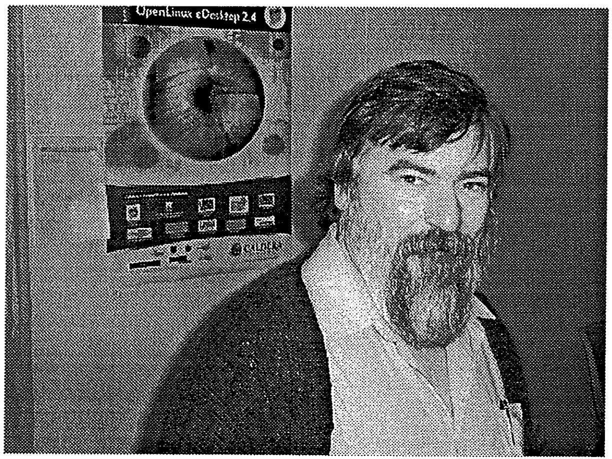
Installing some demo systems



Penguins sleeping before their big day



Greg Lehey

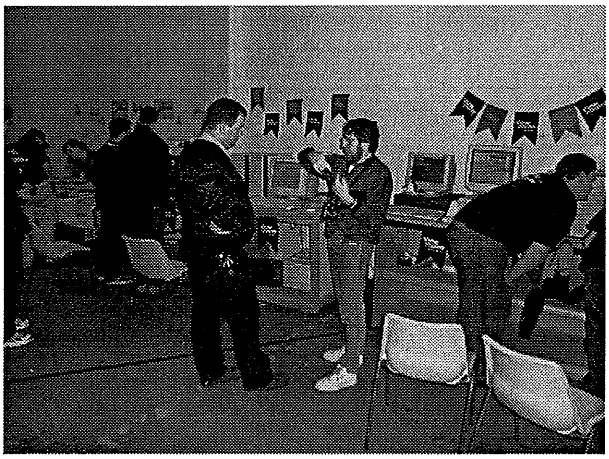


Kevin Macuinus

Doors open for 30 minutes

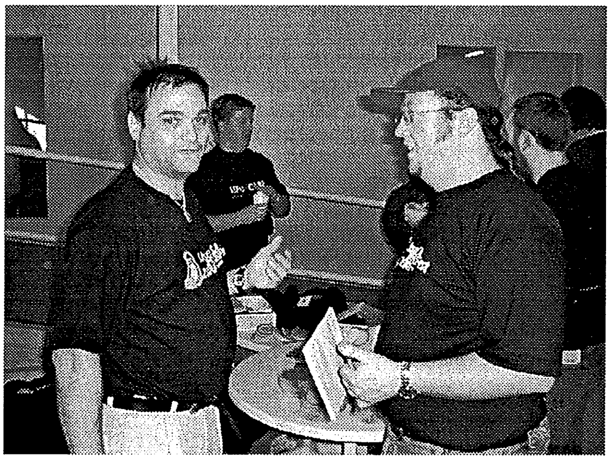
Richard Russell

Dan Shearer giving a seminar



Richard Sharpe explaining Linux

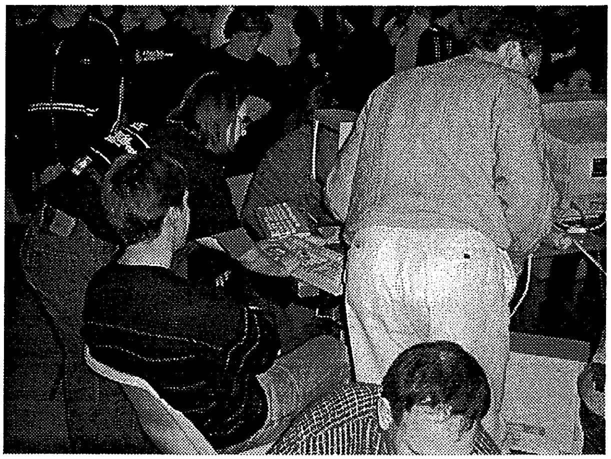
Running out of space; set up on the floor!



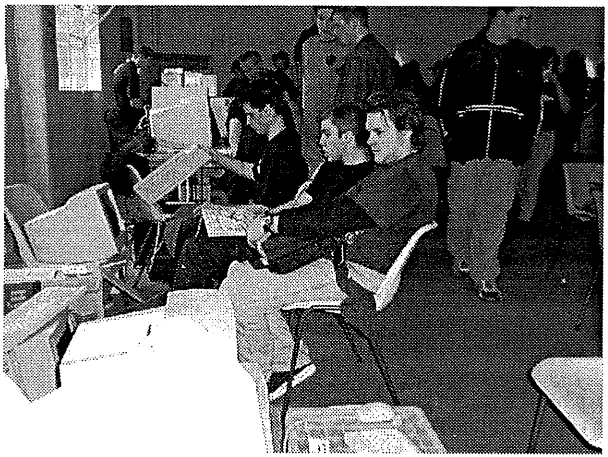
David Newall and Richard Russell

David Newall looking chuffed at
the event's success

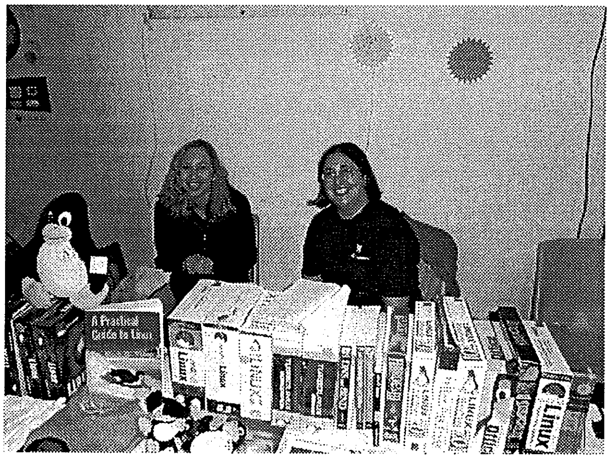
Doors open for 60 minutes



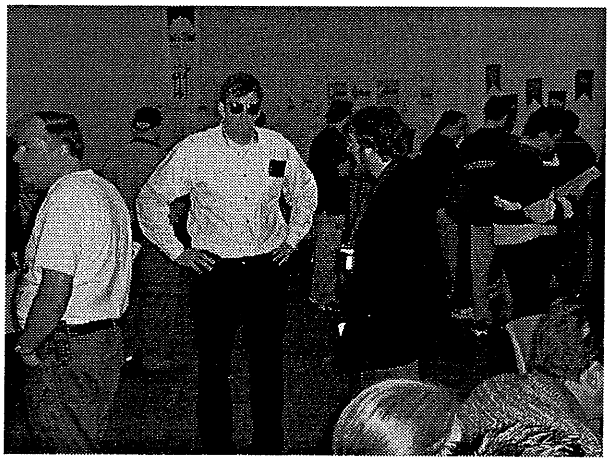
"Can I help?"



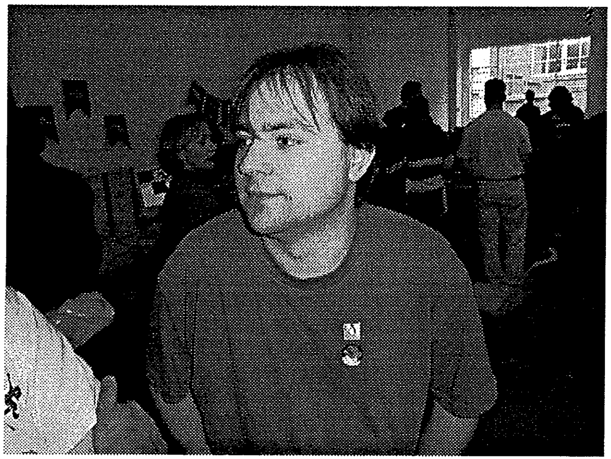
No more bench space, but plenty of boxes :-)



NetCraft's booth

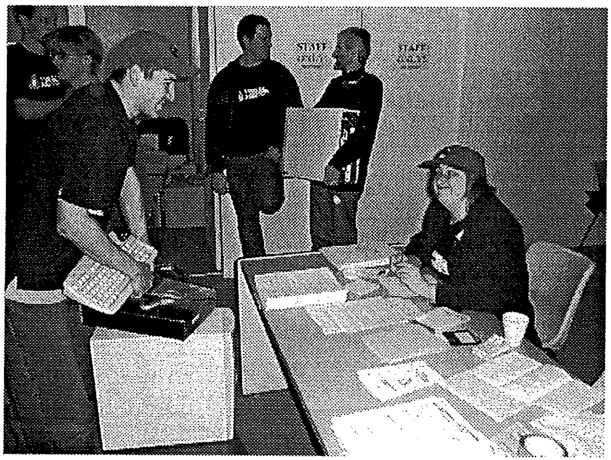


Dave Bennett (Cisco), Dan Shearer (LinuxCare) and Richard Sharpe (Ethereal)



Geoffrey Bennett (owner of the Toshiba Windows Refund cheque)

"FreeBSD? Looks like Linux. Can I have a CD?"



Checkin desk

Equipment Waiting Room - These boxes don't have Linux on them yet

❖

Why write for AUUGN?

AUUGN is looking for articles, so why should you write one?

- It is good experience.
- It looks good on your CV – many jobs these days call for "good communication skills."
- It is your moral duty to share your experience with other AUUG members.

o You could get paid for it.

AUUG is launching a refereed article section in AUUGN. Articles submitted will be reviewed (anonymously) by our esteemed panel, and if your article is accepted for publication, you will be paid an honorarium of $200.

If you would like to submit an article for review, please contact David Purdue at David.Purdue@auug.org.au

AOSS 2
Call for Participation

The second Australian Open Source Symposium (AOSS 2) will be held in Adelaide on Saturday November 25, 2000. The purpose of this event is to bring together the Australian Open Source community on an annual basis.

AOSS is run by developers, for developers. Our goals are to promote the sharing of information and experience, give the community a place to interact, and nurture and harness synergies between Open Source projects.

Just as Open Source is a little different, so is AOSS. While we welcome formal papers, we are actively encouraging informal (but well prepared) presentations that are both timely and interesting. We know that Open Source changes fast, and that developers would rather write code than papers.

The first AOSS event was a resounding success. If you are an Open Source Developer, get involved and make the next one be even better.

Particular topics we are looking for:

- Open Source ideology and/or economics.
- "Work in progress" for an ongoing project.
- "Life in the trenches" experiences from a project (successful or not).
- "Cool ideas" for those who want to start a new project.

TIMETABLE:

Abstracts (around 100 words) are due Monday, 25 September 2000.

PRESENTERS WILL RECEIVE FREE REGISTRATION.

Please email submissions to aoss@esec.com.au

AOSS 2 is proudly supported by AUUG Inc, ISOC-AU, and SAGE-AU.

AUUG Website: http://www.auug.org.au
Phone: 1800 625 655 or +61 2 8824 9511

Fragments from the Usenix Security Symposium

The Anonymous Delegate

Denver, Colorado, USA
August 16-17th 2000

When I arrived in Denver it was unusually quiet. But that was not to last.

The night preceding the conference saw the bar packed to the gills with all manner of strange and dangerous people. Weird concoctions were guzzled, tall tales told, wild plans hatched. Marcus Ranum held forth on the virtues of the Harley Davidson as the pinnacle of the motorcycle art, and the perfection of the chopper form. He was not seen again.

The wireless LAN covering the bar ensured that the gentle light of laptop screens illuminated the scene with an otherworldly ambiance. It was impossible to visually distinguish lemonade from margaritas...

The next morning the main gig was opened by Dr Blaine Burnham, who reviewed all the things we used to know about security but seem to have forgotten. He also reviewed the dress code at DEFCON. These are strange connections, indeed. In any case, he rightly pointed out that if we'd stop periodically reinventing the wheel, we might actually make progress forward and build some secure systems. He especially said good things about orange book, so it might be time to blow the dust off your copy and give it a reread.

Then the dastardly plan became clear. There were two tracks; invited talks and refereed papers. Alas! We actually had to exercise both brain and free will, and all before the hangover has faded. So if I saw things that others didn't, then they must have been in the other room. Hell, who am I kidding? I'd been seeing things that other people couldn't since the eleventh glass last night.

Dave Dittrich gave a taxonomy of distributed denial of service attacks, including a blow by blow description of the discovery of the early Trinoo, TFN and stracheldhrat populations. Amazingly, traces of these agents were picked up months before that late 1999, early 2000 large scale attacks, but the whole shebang was kept under wraps. Full disclosure, not! The sense of the talk was the best is yet to come... DDoS is here to stay and is evolving stealth technology rapidly.

Duncan Campbell's presentation was about Echelon. He seems to have spent a great deal of his recent life tracking down details about the satellite communications interception stations that you find in places like NZ, Britain, Australia and the US. A whole lot is known about Echelon nowadays, thanks to the book "Secret Power" (which apparently still can't be ordered from amazon.com). However, there were a bunch of interception facilities that Duncan showed photos of whose purpose is still unknown. Cool.

The last session of the day saw Mark Chen doing a quick tour of PKI technology, and then explaining how it can all fall over in the real world. Mark seems to be one of a growing chorus of security experts who aren't exactly falling over themselves recommending the wholesale adoption of PKI systems. Apparently, not everything the CA vendors promise comes true, and some of them are even fibbing!

I snuck out of the last part of Mark's talk to catch John Scott Robin's analysis of the Pentium architecture's capability to support a secure virtual machine monitor. Guess what the answer was? Ah well, maybe next time, Intel.

Day 1 was a wrap. All I needed to do was to survive the reception and retire early. Quelle chance?, as they say. Theo de Raadt and the motley OpenBSD band set up in the bar, strategically placed to ensure optimum visibility to the waiting staff and minimum delivery time for fine beverages. As we discovered, fine beverages does not include beer in Denver. Given previous excesses with tequila, however, caution was the better part of valour. So I stayed away from the hard liquor and performed a sequential search for an acceptable ale.

Neat things are happening in the OpenBSD world. Encrypted file systems, cool. Encrypted virtual memory, paranoid and cool. Kick ass IPSEC, with multi hundred megabit throughput.

I also learned that order N algorithms are to be avoided.

The next morning I hit the refereed papers. Intrusion detection was the name of the game, starting with Calvin Ko's explaining how to use software wrappers to detect and counter system intrusion. These wrappers are a layer inserted into the kernel, so that you can audit what is going on, detect attack profiles and take appropriate countermeasures. Like all the IDS-in-the-kernel people, the claim was that the performance hits are insignificant.

Yin Zhang was next with a talk on detecting backdoors and stepping stones. This was done by passively watching traffic going past on the network, and picking up on the signature traffic generated by an attacker's presence. When these techniques were run against real traffic traces from LBNL and UCB, they were effective at identifying real instances with only a few false positives.

Anil Somayaji finished up the intrusion detection block with a description of an ingenious intrusion countermeasure. By tracing kernel calls, he builds up a profile of a given program working correctly. When it exceeds the parameters of that profile, delays are added to each system call. The bigger the deviation, the greater the delays, until the process effectively freezes. The trick seemed to be to get a good starting profile (how do you know you aren't compromised already), but the system obviously fails soft if it isn't a perfect profile.

After lunch, Robert Stone described a method for backtracing DoS packet floods. The obvious approach of querying each router in the path, in turn, apparently doesn't work well, because not all routers have sufficient debugging facilities to be useful. Instead, he suggests creating an overlay network, where interesting packets are sent to special tracking routers, connected to edge routers via tunnels. Note which tunnel the packet came down, and bingo! you know the ingress point.

Yongguang Zhang started from the observation that the use of IPSEC makes it impossible to do things like bandwidth reservation, traffic shaping, proxying, etc. An answer, he proposed to apply two different cryptographic transforms... one to the header and one to the body. Routers sharing keys could therefore peek inside the header to pursue routoid goals, while the payload remained safe from all but the intended recipient. I have to say that I was left with the feeling that in the future we are going to have to decide whether we want security or fancy router parlour tricks, and that the two may be mutually exclusive.

Matthew Smart finished the session describing a brilliant means of slowing the attackers down. One of the things that kiddies often do is run nmap (and friends) on networks, not only to scan for machines and ports but to identify operating systems so that they know which exploits to run. Matthew has built a bridge that tweaks the traffic flowing through it to remove the unique idiosyncrasies that different systems exhibit; he calls this a "fingerprint scrubber".

I switched back to the invited talk stream to hear what the justifiably famous Mudge had to say about antisniff. Now this is truly inspired technology, and if you are not across it go to the L0pht website right now (www.l0pht.com) and find out more. Go on, I'll wait...

OK, back again? Devilishly clever, eh?. I guess you could circumvent antisniff by cutting the TX lead on the ethernet cable, or equivalently performing surgery on pr_ouput() on a captured system. But then people are probably going to notice you're there in other ways: "Hey, the quake server isn't responding!", or "Who is that guy with a laptop plugged into the wall?" In general, therefore, antisniff is truly useful and should be in every sysadmin's toolbox.

I was really taken by Mudge's logic when he was describing what he called the "war college" approach. The sense of it was, we study how to attack our own systems, so that when the enemy attacks we know how to turn them back. This is at odds with the other high profile viewpoint being floated at the conference, being that you don't need to attack systems to learn how to make them secure, with the corollary that full disclosure is bad. That, my friends, is what we call misguided.

The symposium was capped off by a bewildering array of around a dozen five minute work in progress talks, which I shan't try to summarise. You had to be there.

Next year, I'm working up a multiple personality disorder so I can attend all the streams at once. I really hate it though, when you run into yourself in the bar, buy yourself a drink, and then skip out when it's your shout.

If you are kicking yourself for missing the primo security gathering of 2000, then don't despair. The written papers were excellent, and I'm sure you could talk Usenix into selling you a set (www.usenix.org).

✣

Dear AUUG Members,

Over some Japanese beer it was concluded that AUUG members would benefit from a discussion list. The purpose for this list is to provide a means for AUUG members to communicate, ask questions, network, discuss random geek stuff and the finer points of beer.

To join the list, send a message to talk-request@auug.org.au with the subject of "subscribe talk"

Sarah Bolderoff
Sarah.Bolderoff@auug..org.au

AMERICAN
BOOK STORE

***10% DISCOUNT
TO AUUG MEMBERS
ON OUR COMPLETE RANGE
OF COMPUTER, BUSINESS
AND GENERAL BOOKS***

**173 Elizabeth St, Brisbane Queensland 4000
Ph: (07) 3229 4677 Fax: (07) 3221 2171 Qld Country Freecall: 1800 177 395
american_bookstore@compuserve.com**

Name: ____________________ Date: __________

Address: ____________________

____________________ Post Code: __________

Phone Number: ____________________

Payment Method: □ Cheque □ Money Order □ Amex □ Bankcard
□ Diners □ Mastercard □ Visa

Card Number: ____________________

Expiry Date: __________ Signature: ____________________

This is a: □ Special Order □ Mail Order □ Book on Hold

QUANTITY TITLE PRICE

SUBTOTAL $ ....

LESS 10% DISCOUNT $ ....

POST & PACK $ ....

TOTAL $ ....

POSTAGE AND HANDLING FEES: 1 BOOK $6.00 2-4 BOOKS $7.00
BOOKS OVER $70.00 WE WILL SEND CERTIFIED - PLEASE ADD ANOTHER $1.50 OR WAIVE CERTIFIED DELIVERY.

FOR SPECIAL ORDERS, PLEASE ENCLOSE $10.00 PER BOOK AS A DEPOSIT.

My Home Network

Frank Crawford
frank@crawford.emu.id.au

This column is late, being written just after the start of Daylight Savings. In fact, this year's Daylight Savings is early, being moved forward for the Olympics. The early start to each day (including waking up in the dark) is a major problem with the change to the time, but there is also one reason I like it, it gives me a reason to set the times on all the clocks in the house.

Of course there is one problem with setting all the clocks, what do you use as a standard? Most people I know just believe what is on their watch, others phone up Telecom, others believe their radio. In my case, I believe the time on my computer.

If you wander around any office you will find that the computers are set to random times, most of which cluster around the correct one, but, in fact computers and the Internet form one of the most accurate time systems widely available. Most people know of atomic clocks, and many know that there are ways to set the time across the Internet, but most don't think of the applications for their home.

In reality, anywhere there are two or more computers it is trivial to synchronise the time between them, and further, if you have some external connection, you can synchronise with it. While there are at least three different time synchronisation protocols, and most can be run either as a single shot or as a daemon, I find the best, at least for the Unix world is the Network Time Protocol (NTP).

NTP is a protocol that allows continuous synchronisation between both a server-client form or in a peer arrangement. The daemon that implements the protocol is call `xntpd`, and comes with a number of monitoring and control programs. Even more importantly, when `xntpd` is running it gradually shifts the system time to bring it in line, ensuring that there are no steps backwards or dramatic steps forward. Certainly, this requires assistance from the kernel, but the `adjtime` system call is fairly standard these days.

Along with `xntpd` is the program `ntpdate`, which is a one-shot program, generally used at boot time to the initial time. `ntpdate` can either shift the time in one hit (best used at boot-time) or through the `adjtime` system call.

The standard distribution of `xntp` for Red Hat (`xntp3-5.93-14` at the time of writing this) is simple to set up and includes configuration options to run `ntpdate` prior to starting `xntpd`. The only information you require is the hostname or IP address of a suitable time server. While you can try and connect to a top level server, in most cases it would be better to connect to one specifically set up by your ISP (often with the name `ntp` or `time`).

`xntpd` is controlled by a configuration file, which is normally found in `/etc/ntp.conf`, and looks something like:

```
#
# Undisciplined Local Clock. This is a fake driver intended for backup
# and when no outside source of synchronized time is available. The
# default stratum is usually 3, but in this case we elect to use stratum
# 0. Since the server line does not have the prefer keyword, this driver
# is never used for synchronization, unless no other
# synchronization source is available. In case the local host is
# controlled by some external source, such as an external oscillator or
# another protocol, the prefer keyword would cause the local host to
# disregard all other synchronization sources, unless the kernel
# modifications are in use and declare an unsynchronized condition.
#
server 127.127.1.0   # local clock
fudge  127.127.1.0 stratum 15

#
# Drift file.  Put this in a directory which the daemon can write to.
# No symbolic links allowed, either, since the daemon updates the file
# by creating a temporary in the same directory and then rename()'ing
# it to the file.
#
driftfile /etc/ntp/drift
#multicastclient                 # listen on default 224.0.1.1
broadcastdelay      0.008
```

```
#
# Authentication delay.  If you use, or plan to use someday, the
# authentication facility you should make the programs in the auth_stuff
# directory and figure out what this number should be on your machine.
#
authenticate no

#
# Keys file.  If you want to diddle your server at run time, make a
# keys file (mode 600 for sure) and define the key number to be
# used for making requests.
# PLEASE DO NOT USE THE DEFAULT VALUES HERE. Pick your own, or remote
# systems might be able to reset your clock at will.
#
keys          /etc/ntp/keys
trustedkey    65535
requestkey    65535
controlkey    65535

server ntp.crawford.emu.id.au     prefer
```

If you look at this file, you will see that there are a lot of possible options, most of which aren't important for a home network. The two important lines here are the "server" lines. One of them sets the preferred server to ntp.crawford.emu.id.au, i.e. my local time server, and the second, as the comment say, is a fake driver, to follow the local clock if no connection is possible. The additional "fudge" line set the local clock to a low “reliability” (NTP clock start at 1 and count up for each level below that).

For my server system, instead of setting the time to ntp.crawford.emu.id.au, I would set it to the NTP server of my ISP, in which case, when the connection is down, the local clock is important.

Along with /etc/ntp.conf, Red Hat has a configuration file, /etc/ntp/step-tickers, which contain the names or IP addresses of hosts to use to set the initial time at boot.

Okay, so now your Unix and Linux hosts are running fine, keeping time, and generally ticking along, what about those other poor machines you have in your home network. Don't despair, help is at hand. In fact, the simplest is to run NTP on those as well. xntpd has been ported to Microsoft Windows NT, it runs as a service and works well. In fact it works so well that Microsoft have incorporated a cutdown version of NTP, called SNTP (Simple NTP) in Windows 2000. Ignoring the difficulty of getting through Microsoft's documentation, it synchronises well against a full NTP version.

To complete the set, Apple now ship NTP as standard in MacOS 9, and again you can happily synchronise with your Unix system.

Of course if you are stuck with some old system that doesn't easily have NTP available, doesn't mean you have to go "unsynched". NetBIOS has long had the ability to synchronise with a time server, using the command:

```
net time \\server /set
```

where server is your time server.

You can put this in a batch file to be executed at startup and then every time you boot, your clock will be right. (BTW Windows 2000 systems in a domain automatically synchronise with the master server for the domain.)

To enable your Unix box to act as a server, you need to run Samba (and who doesn't these days :-)) and add the line:

```
time server = yes
```

to you /etc/smb.conf configuration file.

Of course, if you have an old Macintosh, this wouldn't work, but don't worry, there is a solution for you too. Some years ago, the University of Melbourne wrote a program called tardis for MacOS, which allowed it to synchronise using a proprietary protocol. They also wrote a server which works with Netatalk, the Linux package which supports AppleTalk. Unfortunately, the server program, timelord, has a few byte ordering problems on Intel platforms, so you will need to pick up a few patches which are available on the Netatalk home page, or from me (as I wrote them originally - see I do some other things occasionally :-)).

So given all these tools, it is easy to keep all your computers running with the correct time, of course, if you are on Unix, you also need to make sure you have the correct timezone, a totally different problem I won't go into now.

For those of you who read this far each time, you will notice I still haven't written about security. I will some time, but only when I have sufficient detail, so keep reading, let me know what you think, and send in some interesting ideas.

❖

Security Symposium
Friday 3 November, Melbourne

Call for Participation

The AUUG Security Symposium will be held in Melbourne on Friday 3 November 2000.

The purpose of this event is to exchange ideas on the improvement of the security for the systems and networks we manage.

AUUG Inc invites proposals for papers relating to:

- Network Security
- Host Security
- Risk Assessment and Mitigation
- Intrusion Detection
- Distributed Security Solutions
- Authentication and Authorisation Methods

Speakers may select one of two presentation formats:

Technical presentation:
A 25-minute talk, with 5 minutes for questions.

Management presentation:
A 20-25 minute talk, with 5-10 minutes for questions (i.e. a total 30 minutes).

Panel sessions will also be time-tabled in the day and speakers should indicate their willingness to participate, and may like to suggest panel topics.

TIMETABLE:

Abstracts (around 100 words) are due Monday, 18 September 2000. Please note that formal papers will not be required, since there will be no proceedings for this event.

Presenters will receive free registration

All submissions to be sent via Email to: busmgr@auug.org.au
Or Faxed to AUUG at: +61 2 8824 9522

Further information can be obtained by calling AUUG on:
Phone: 1800 625 655 or +61 2 8824 9511

The Open Source Lucky Dip

Con Zymaris
conz@cyber.com.au

Welcome back.

I'm whipping up this edition's cocktail of code and comment in between preparing for the Melbourne IT 2000 trade show. I'm helping man a stand in the Linux Pavilion, which should be lot of fun. We were involved in something like this at last years event, and I can tell you that the Linux and Open Source arena was the hit of the show. We talked to over 5,000 people, many of them new to Linux Open Source and Unix.

The reaction from this event highlighted, in my mind, something of great importance; there's one thing that this industry needs; something that it needs to keep the 'buzz' alive; something to keep new talented practitioners joining the industry; something it needs to sell hardware and services; something that it needs which helps differentiate the technical computing arena from say the Car or Toaster industry. That something is an idea, technology or promise which, in a sense, overthrows most everything that preceded it. I've seen it happen on three occasions during the 21 or so years that I've been coding or using computers. Here's a whirlwind re-count.

The first time was around 1979, when, incredibly, someone managed to fit a whole computer into something that could sit on a desk! Something that an individual enthusiast can claim of: "it's mine! All mine!" That something, of course, was the 8-bit microcomputer, denoted by the likes of Apples, CP/M and MicroBees (all of which I used and admired.) These systems, I was sure, would one day change the World. And they did.

The second time this happened was around 1989. At the time, I had an account on a few systems at Melbourne Uni. On these, I'd discovered about this wonderful universe called the Internet. Simply amazing. Here, in it's as yet unrealised proto-form, was something I thought was the simple most important method of generating and spreading ideas developed since Gutenberg's press. I was sure, this Internet thing would one day change the World. And it did.

The last of the trifecta of 'disruptive' technologies or ideas, is, of course, Open Source. While I've been using Linux (and it's spiritual precursor, Minix) at the office for almost a decade, it's only been the last few years when I finally came to understand the power of its underlying meme. Here, yet again, is something, (that yes, has been in 'backroom' practice for decades,) is turning our industry on its head, right here, right now. And the industry loves Linux and Open Source for that very reason. We thrive on riding the bow-wakes of 'these' disruptive ideas, like excited buoys when a speed boat zooms past. Our industry feeds off that excitement. It's like throwing accelerant on a camp-fire. So, here's to the technical IT industry. Let there be many more 'buzz' inducing disruptive ideas to come. Oh, and I did mention that I think Linux and Open Source will one day change the World, didn't I? ;-)

✧ ✧ ✧

Let's now take a look at this edition's grab-bag of tools and apps.

✧ ✧ ✧

In a press release issued earlier today, Microsoft attacked Stallman's outlandish requests. "At Microsoft, we don't scream at people who say Windows instead of Microsoft/Windows..." -- <smirk>

✧ ✧ ✧

WXWINDOWS/GTK

For the cross-platform coders amongst you, `wxWindows`, a long available cross -platform GUI library, is now available with GTK widget support. According to the `wxWindows` team, this new offering has classes for all common GUI controls as well as a comprehensive set of helper classes for most common application tasks, ranging from networking to HTML display and image manipulation. There are also Python bindings available for the GTK and the MS Windows port, and documentation available for practically all classes.

License: BSD
Grab it at :

```
http://wesley.informatik.uni-
     freiburg.de/~wxxt/
```

✧ ✧ ✧

WEB2LDAP

For those experimenting with LDAP, check this out. `web2ldap.py` is a full-featured LDAP client written in Python designed, according to author Michael Stroeder, to run as a stand-alone Web gateway, as a CGI-BIN under the control of a WWW server, or as handler module under Apache with `mod_python`.

License: GPL

```
http://www.web2ldap.de/
```

✧ ✧ ✧

VISUALOS

Something cool to get your students into! Forget those drab Turing machine simulations ;-) VisualOS was developed as an educational visual simulator of an operating system for GNOME/GTK+. It represents a working operating system visually, allowing the user to select the different algorithms to use for each of the simulated subsystems: CPU, Memory and disk I/O. I'm sure if I had this way-back-when, I would have achieved greater marks for comp-sci. That's my story and I'm sticking to it...

License: GPL

`http://VisualOS.sourceforge.net/`

✧ ✧ ✧

TOMAHAWK

This is an interesting one. Tomahawk claims to be an Apache-based Web server with integrated Squid object cache capabilities running on an intuitive Web-based UI. It also claims dramatically increased server performance. Go figure ;-)

License: GPL

`http://www.elctech.com/`

✧ ✧ ✧

MOTION

For all you site-security mavens out there: motion uses a video4linux device for detecting movement. It makes snapshots of the movement which later will be converted to MPEG movies, making it usable as an observation or security system. It can send out email and SMS messages when detecting motion.

License: GPL

`http://motion.technolust.cx/`

✧ ✧ ✧

MCFEELY

There are probably dozens of job-control systems our there for Unix/Linux, but here's another to add to the list. McFeely, its author claims, makes it possible to run multiple programs, in a specified order, on multiple hosts. It was created to solve the problem of automatically managing users at an ISP where the users have resources like home directories on multiple machines.

License: GPL

`http://web.systhug.com/mcfeely/`

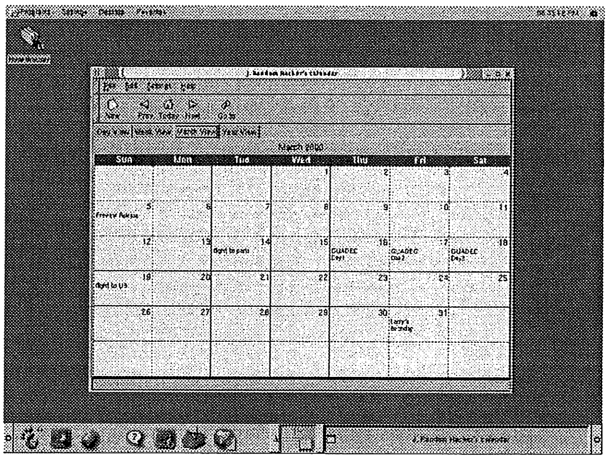
✧ ✧ ✧

HELIX CODE

If you want the latest and sexiest Linux/Unix desktop around, look no further than Gnome's Helix Code.

License: GPL

`http://www.helixcode.com/`



Gnome Helix Code

❖

If you have any experiences using Linux that you would like to share with other AUUGN readers, drop us a line at:

auugn@auug.org.au

We'd love to hear from you!

DRAFT:

AUUG AGM Minutes

Thursday 29 June 2000

Committee present:

David Purdue	- DP
Michael Paddon	- MP
Mark White	- MW
Luigi Cantoni	- LC

Minutes taken by:

Elizabeth Carroll	- EC

1) Meeting opened at 1700 by DP

2) Apologies:

Stephen Boucher

3) Minutes of the previous AGM:

Motion for the minutes to be accepted: Greg Rose
Seconded: Anthony Rumble
Carried.

4) President's report: DP

We currently have a financially stable organisation, thanks to a profitable AUUG2K, generous sponsorship and a healthy bank balance. AUUG has tightened up many of its internal processes and expense, including bringing things like membership processing back in-house.

AUUG is weak in respect that we currently have low membership numbers. This means that we rely on the annual conference, as well as membership fees, to fund our activities.

We would like to see more delegates at events. AUUG2K had slightly lower attendance that hoped, however we knew that the unusual date of the conference would likely have that effect. We also suspect that some members couldn't attend due to the implementation of the GST.

In the coming financial year, AUUG will not be having an annual conference; the next is scheduled for September 2001, in Sydney. Therefore, we will be concentrating on delivering smaller events, such as the open source and security symposia. By returning more value, in more ways, to our members we intend to attract new people to AUUG.

We also need to call upon the membership to remind you that AUUG is a member driven organisation. You should not be asking, "What can AUUG do for you?", but rather "What can you do for AUUG?". We challenge all members to get more involved.

Motion to accept the President's report: Andrew McRae
Seconded: Don Griffiths
Carried.

5) Treasurer's report: LC

I am Luigi and I took office as treasurer in July of last year. I was unable to immediately take over my role as treasurer as the previous accounts and books were incomplete, and thus all the information I required was not available. I was given the cheque and deposit books shortly after and was thus able to control the flow of funds into and out of our accounts.

Since last year's conference, I have had full control (in my capacity as treasurer) of the accounts and how they are maintained. The accounts I will be presenting represent a complete and accurate record from that point.

In the middle of last year, the executive decided that AUUG should take over more of the roles previously done externally, such as memberships and conference management. This would both save costs and provide more accurate information and control in these areas. This conference has been the first for many years where we have managed all the organisation and logistics for ourselves.

The executive would appreciate your feedback to know how successful we have been. We would like to know if you feel this conference's registration, administration etc. have been an improvement over the past. We would also like to know if your membership handling has been better, more accurate and faster. These are the areas of administrative improvement we have concentrated on this year and by doing them in house we have also saved your membership money.

We began to implement these new ideas during the 1999 conference. This resulted in a significantly better than expected financial result for that event.

At that time a separate cheque account was maintained for conferences. I am unable to provide an audited report for either this account or for the general AUUG account as we do not possess complete records for that period. However, what I can say is that the accumulated funds from the last few conferences (including 1999) in the conference account was $50,943.87, and I believe that the surplus from the 1999 conference represents more than 50% of this amount. The conference account has now been closed and all funds transferred to the general AUUG account.

We now trade entirely from one account but maintain separate budgets and costing areas for conferences, tutorials etc.

To summarise this year's tutorial program and conference:

For the tutorial program:

- All tutors have been paid and there only remain printing costs of $1,273.56 and venue costs of $2,076.00 to be paid. There is $16,550 of income from attendees that is yet to be received.
- If all monies owed are received, the tutorial program should create a forward estimated surplus of $26,583.25. This is the first year that all tutors have been paid before the conference finishes.

For the conference program:

- All suppliers have been paid, excepting only printing and venue/food costs of about $22,000. There is about $8,250 of income from attendees that is yet to be received.
- If all monies owed are received, the conference should create a forward estimated surplus of about $34,000.
- The total number of attendees is approximately 150. This is slightly down from previous years but given the different time of year in which the conference is being held, and the fact that it is not being held in Sydney or Melbourne, I feel the numbers are still good.

Once again we are very keen to receive feedback. What are your thoughts on this type of venue and format? We have over a year before the next conference and we would like to give you the type of event you want.

General accounts:

- Membership funds have brought in $52,200. Since September, when I took over the accounts, the main costs have been:
  * AUUGN - about $11,500
  * General office costs - about $7,800
  * Wages and associated costs - about $37,000

Here it should be noted that a great deal of this area was actually spent on conference work and in reorganising everything to the new in-house way of recording and administration. In future we will be costing this more directly to the areas where it is spent.

* Executive meeting costs - about $8,600

- It must also be noted that during this period we also paid off previous liabilities of $16,500.

- Our accounting systems are now improved to the point where we are able to give a more complete financial position at the AGM.

- Currently we have $116,857 in the bank. We are owed about $31,550. Uncashed cheques and money we owe comes to about $40,000. Therefore, we have about $108,000 as funds to go forward with.

- This improved financial position and better controls should enable us to maintain our current fee structure over the next period.

- Chapters with their own funds have not been reported on here, and this is one area which we will be concentrating on during the next year.

In my opinion AUUG is definitely solvent and has sufficient funds to enable us to work towards providing a better service for you, the members.

The John Lions Fund now has over $30,000 in it and we are changing the investment strategy to make it self sustaining. Nevertheless this should not be seen as a reason for no longer adding to that fund. We still need to nurture new open systems talent amongst our students and a growing fund will help achieve that aim.

The following questions were put to LC by the membership:

A) Don Griffiths:
Q: In regards to the GST, will AUUG accept the GST or absorb it?
A: AUUG will not absorb the GST, although membership rates remain the same.

B) Lawrie Brown:
Q: What is floating out there finance wise?
A: Basically, AUUG looks after the chapters centrally, with the exception of ACT, QLD and VIC.

C) Greg Rose:
Q: Did we get a copy of the old members' list during the changeover?
A: Yes

D) Catherine Allen:
Q: If you can't audit, does it affect us with the ATO?
A: No, unless members request an audit. We need to have a full year's activity, therefore next year we will be in that position. Having the chapters managed under the standardised controls will be a great help.

DP stated that chapters can have AUUG centrally handle their funds, in that case we need a full record of transactions from them.

MP stated that the Exec feels that our accounts are well under control, and that there is absolutely no evidence present of improper activities anywhere within AUUG and its chapters.

Motion to accept the Treasurer's report: Frank Crawford
Seconded: Lawrie Brown
Carried.

6) Secretary's report

There was no Secretary's Report.

7) Other business

- General discussion re: AUUGN.

It has been noted, that from past surveys and experience, AUUGN is usually cited as being the best member benefit provided by AUUG.

Gunther Feuereisen, the editor, has found it frustrating over the past year, due to the fact that the contribution for content has fallen. This is reflected in the June edition, which only contains 32 pages.

As Gunther has stated, he wishes to produce a quality journal, but at the end of the day, he himself, can only write so much.

Gunther has indicated that he will stay on until the end of the year, as editor, at which time AUUG will need to find a new one.

The question, also arises, do the AUUG members still want to receive AUUGN in its current form, maybe a monthly A4 flyer would suffice?

The following issues were discussed by the membership:

- What about the web... it reduces cost?
  * Many people prefer hardcopy.
  * Printed journal carries prestige.
  * Web delivery has it's own costs.
  * Risk of losing historic aspect.

- How about producing an A4 sheet?
  * Easy to confuse with junk mail.

Up until now, copy from Login was free for AUUGN to reprint, however Usenix is beginning to charge other user groups for this right. AUUG can purchase reprint rights, or simply redistribute Login complete.

Catherine Allen stated that she does not want to read Login. If she did, she would have joined Usenix. She would rather see Australian content.

Suggestions from the membership as to possible ways of obtaining quality articles:

- Try the Computer Science Departments at the universities, possibly offer a free 1 year subscription.

- Pay contributors up to $200 an article, via a refereed process.
- A refereed process possibly makes AUUGN more attractive to academia.
- AUUGN is primarily a professional journal, not academic.

- Deadline dates should be emailed to auug-announce.

- Look towards the Linux market for articles, eg. through the various Linux User Groups.

The best member benefit is the other members of the user group. It provides the opportunity to exchange ideas etc. AUUGN is an extension of this. Although we have events, not all members are in a position to attend, therefore AUUGN is there for them. If a member is doing something interesting, it provides them the opportunity to let other members know about it.

DP called for volunteers to edit AUUGN, with key goals being to grow the amount of content.

- General discussion re: the LinuxSA installfest.

To be held on 15 July in Adelaide. This event is be sponsored by AUUG, and is an opportunity help kick start the SA Chapter. The public liability insurance for the event will be covered by AUUG.

- New committee.

The Executive Committee which comes into office on 1 July 2000, consists of:

President	- David Purdue
Vice President	- Malcolm Caldwell
Secretary	- Michael Paddon
Treasurer	- Luigi Cantoni
General Committee	- Alan Cowie
	- Peter Gray
	- David Newall

There are two vacancies for General Committee, therefore AUUG is looking for volunteers. Nominations were taken from the floor.

Volunteers nominated:
- Sarah Bolderoff
- Adrian Close
- Greg Lehey
- David Shaw

The Executive Committee will interview the nominees and second two individuals according to the procedures laid out in the AUUG constitution.

- Thank you to the auug2k programme chair.

Motion to thank Frank Crawford, the AUUG2K Programme Chair, for his efforts: Greg Rose
Seconded: Shane Matson
Carried.

- Membership.

Andrew McRae stated that in regard to membership, it is a very hard task to gain new members and asked the Committee what ideas they had in order to achieve this.

MP stated that the focus on generic chapter activities has become less effective over recent years, and was being supplanted by specialised symposia. The intention of this change is to attract new members.

General discussion:
- AUUG currently has over 600 members, from a peak of 1300. This peak was skewed by the large joint WWW conferences run in 95-96.
- The approach of running many small events has attracted many members to Usenix.

- What does AUUG stand for?
We can say that it is the, Australian Unix Users' Group. Our focus is open systems and open source. Unix is a large part of this, but not to the exclusion of all else.

8) Meeting closed by DP at 1805.

Chapter News:
Victoria

Enno Davids
president@vic.auug.org.au

Well its been a long time between drinks, at least as far as writing one of these columns is concerned for me. But changes are upon us and so it seems apt that I jot down a few notes to keep everyone informed.

First off, those of you who have been coming to our regular meetings or indeed those who haven't but were meaning to, should be aware that we're changing our venue. For some time we've been meeting at Asti's in Carlton but with the change of owners has come a change of direction for the restaurant and so we're casting about for a new venue. We tried the Carlton Curry House this week past and may yet try a few more places before settling on a new semi permanent venue. We have some alternatives and we'll be trying them in the upcoming months. To hear about these arrangements, its best to subscribe to the members-announce list at `vic.auug.org.au` and you can do this by sending email with subscribe in the Subject to the usual -request address, i.e.:

`members-announce-request@vic.auug.org.au`

Note that that's the `vic.auug.org.au` server and not `auug.org.au` though.

Meetings are still mostly on the third Wednesday of the month so expect an announcement at least immediately before the meeting and committee organisational skills allowing a week ahead of the event as well.

If any of you know good venues in or near the CBD that you feel we should be considering, now is the time to drop us a suggestion. As we're sampling a few alternatives, we can always try on or two more till we find the right fit.

Next, its worth noting that as we missed out on organising a Summer conference this year we're going to try to have one in November. So if you had a paper working up or indeed if you have something interesting you'd like to talk about, now is the time to start polishing your notes. In the style of AOSS-1 we're thinking of making the paper itself optional. This is mostly to reduce the burden on speakers of preparing, given the other gross calls on our time that all of us seem subject to these days. Given fewer papers, we expect little of no printed material for the delegates which in turn will mean a low cost of attendance. What papers & supplementary material we get will be published on the Web for delegates to download and print at their discretion. Dates are still being finalised but expect the big day to be in November as I noted.

Finally, just an advance warning that the pre-Christmas Go-Kart night is going to be on once again. Why am I noting this now? Well as a friend noted, there are only 16 weeks left to Christmas (as I write this). What a scary thought. Anyway, time to get out the driving gloves, dust off the helmet and try that zen thing to get into the Schumacher/Hakkinen/Montoya mindset. Either that or just have fun. Both seem to work. Dinner afterwards to clear out the petrol fumes as usual.

AUUG Corporate Members

as at 31 August 2000

Andersen Consulting
ANI Manufacturing Group
ANSTO
Aurema Pty Ltd
Australian Bureau of Statistics
Australian Geological Survey Organisation
Australian Industry Group
Australian National University
Australian Taxation Office
Australian Water Technologies P/L
BHP Information Technology
British Aerospace Australia
Bunnings Building Supplies
Bureau of Meteorology
C.I.S.R.A.
Camtech
Cape Grim B.A.P.S.
Central Queensland University
Centrelink
CITEC
Commercial Dynamics
Computer Science, Australian Defence Force Academy
Corinthian Industries (Holdings) Pty Ltd
Corporate Express Australia Limited
Crane Distribution Limited
CSC Australia Pty. Ltd.
CSC Financial Services Group
CSIRO Manufacturing Science and Technology
Curtin University of Technology
Cyberscience Corporation Pty. Ltd.
Cybersource Pty. Ltd.
Daimler Chrysler Australia - Pacific
Dawn Technologies
Deakin University
Department of Defence
Department of Land & Water Conservation
Education QLD
Energex
eSec Limited
Everything Linux
G.James Australia Pty. Ltd.
Great Barrier Reef Marine Park Authority
IP Australia
IT Services Centre, ADFA
Land Information Centre
Land Titles Office
Macquarie University
Mercantile Mutual Holdings
Motorola Australia Software Centre
Multibase WebAustralis Pty Limited
Namadgi Systems Pty Ltd
Nokia Australia
NSW Agriculture
NSW Public Works & Services, Information Services
Peter Harding & Associates Pty. Ltd.
Qantas Information Technology
Rinbina Pty. Ltd.
SCO
Security Mailing Services Pty Ltd
Snowy Mountains Authority
St. John of God Health Care Inc.
St. Vincent's Private Hospital
Stallion Technologies Pty. Ltd.
Standards Australia
State Library of Victoria
TAB Queensland Limited
Tellurian Pty. Ltd.
The Fulcrum Consulting Group
The University of Western Australia
Thiess Contractors Pty Ltd
Tower Technology Pty. Ltd.
University of New South Wales
University of Sydney
University of Technology, Sydney
Victoria University of Technology
Westrail
Workcover Queensland

❖

Unix Traps and Tricks

Jerry Vochteloo
jerry@socs.uts.edu.au

A while back I asked people what they wanted from this column, there seemed to be some interest in covering some of the basics again. I have written a little article on UNIX file permissions, I apologise if it is a little Linux ext2 centric.

If anyone else would like to contribute with a short article on anything that they found initially tricky or anything else please contribute. Those people that emailed me and said that they would contribute, that would be gratefully accepted. We need contributors.

Thanks

SHORT PRIMER ON UNIX FILE PERMISSIONS

All resources in UNIX are viewed as files. It is therefore not surprising that access permissions are file based. Every process in UNIX has associated with it a user id and a group id. These id's determine the access that each process has on each file. There are three main operations that can be performed on a file: read, write and execute. All files in UNIX also have an owner, and belong to a group. Rights to a file are specified by three sets of permissions. The first is the rights of the owner, the next set determines what rights the group members have, while the last set determines what rights all other users have.

```
unix> id
uid=10371(jerry) gid=10371(jerry)
unix> groups
jerry mungi accstaff
unix> ls -l
total 1130
-rw-------   1 jerry    jerry     19129 Oct 30 11:35 dead.letter
-rw-------   1 jerry    jerry     22190 Oct 12  1996 grub-ext2fs-floppy.gz
-rw-------   1 jerry    jerry     28400 Jul 30 09:03 in-mail
drw-r-x---   1 jerry    mungi      1024 Jul 30 09:02 mungi-src
-rw-------   1 jerry    jerry      8142 Jul 30 09:04 newgive.doc
-rw-r-----   1 jerry    mungi       322 Jul 30 08:02 proposal
```

In the above example, the user id of the process is 10371 for user `jerry`. User `jerry` also belongs to groups `jerry`, `mungi`, and `accstaff`. When the files in a directory are listed, the rights are represented as 9 characters following the letter indicating the file type. The first three letters indicate the rights that the owner `jerry` has on the file; `rw-` in the case of file `dead.letter`, corresponding to read and write permissions. The next three indicate the rights that the group has. In the case of the file proposal, which belongs to the group `mungi`, the permissions are `r--`. This means that all members of the group `mungi` have read permissions on the file. The last three letters indicate the rights that other users in the system have on the files. All of the files in the example have rights `---` in this field. This means that others users have no access to these files.

UNIX provides a protected procedure call through the use of set-user-id programs. When executed, these programs run with the user id of the program's owner, usually root (the superuser to whom no access rights checks are applied). An example of this on Linux is

```
unix> ls /usr/bin/passwd
-r-s--x--x   1 root     root        22312 Sep 26  1999 /usr/bin/passwd
```

In the above example, the execute bit on the passwd file is set to `s` instead of `x`. This means that if any user executes this file it will run with the user id of `root`. This allows users to modify the password database, which is an operation which they are normally prevented from doing.

The access permissions on files are reasonably obvious, how they interact with directories I will cover another time.

The UNIX protection model is simple and well known. It does, however, have a number of drawbacks and a couple of tricks that we can use.

The first problem is the granularity of protection. A file can only ever have one owner and belong to one group. This prevents more than one group having access to a file. Compounding this problem is the fact that only the system administrator is able to create groups in UNIX, restricting the users ability to tailor protection for their files. For example sharing a file with a person that shares no groups with you is difficult, the only options we have are: share with group, or share with all other users. The default, I tend to find, is that files that are meant to be shared are usually made world readable.

Note: Most modern UNIXes do provide full Access Control Lists (that is allow you to specify the level of access to a file for any user), Linux `ext2` seems to have source code hooks for it, but no implementation as yet.

The second problem is that set-user-id programs have been the cause of many security breaches in UNIX systems, Set-user-id programs that are used to perform system duties (such as adding a file to the printer queue) usually are set to be uid `root`, as `root` is the only user that is guaranteed to be able to access the caller's file. This is in gross violation of the principle of least privilege, in that a process that only needs to have the rights to access a user file and a printer spooler actually has access to all the files in the system.

The third problem is that there are no permission checks for `root` (at least on linux `ext2`, I don't have access to a non-linux box that I have `root` on). This means that if you have a file that is read-only to `root` then the permissions will not remind you that the file should not be writable. (which is why we should alias `rm` to `rm -i`)

A little useful trick to finish off. In UNIX Permissions are checked in the order user, group, other. As soon as permissions are found to be denied the search stops. This means that if you have

```
unix> ls testfile
-rw-----rw-  1 jerry    mungi       22 Oct 15  1996 testfile
```

permissions on a file, any members of the group `mungi` would NOT have permission on this file, while all other users would. This allows you deny access particular group of people.

I hope that this has given some insight to UNIX file permissions, 'till next time, I will talk about permissions on directories and what they mean

Dear AUUG members,

As an incentive to submit items/articles/photos/whatever to AUUGN, we are introducing AUUGN Donor Points.

You can use your AUUGN Donor Points to purchase random stuff provided from time to time by the AUUG Management Committee. Perhaps membership/conference/symposium discounts, T-shirts, software distributions, mugs, caps or other geeky paraphernalia.

To qualify for AUUGN donor points, you merely submit an article (and have it published) in AUUGN. The better the article the more AUUGN Donor Points...

The awarding of AUUGN Donor Points is entirely at the discretion of the AUUGN Editor.

The topic is open for discussion. We want to know what you think of this idea, so please join the new AUUG mailing list talk@auug.org.au and tell us what you want for your AUUGN points.

AUUG Chapter Meetings and Contact Details

CITY	LOCATION	OTHER
BRISBANE	Inn on the Park 507 Coronation Drive Toowong	For further information, contact the QAUUG Executive Committee via email (qauug-exec@auug.org.au). The techno-logically deprived can contact Rick Stevenson on (07) 5578-8933. To subscribe to the QAUUG announcements mailing list, please send an e-mail message to: <majordomo@auug.org.au> containing the message "subscribe qauug <e-mail address>" in the e-mail body.
CANBERRA	Australian National University	AUUG (Canberra) run (semi) regular monthly meetings held at 7:30pm in Cellar Bar/Fellows Garden at University House, Balmain Cres, ANU; on the second Tuesday of the month
HOBART	University of Tasmania	
MELBOURNE	Various. For updated information See: http://www.vic.auug.org.au/auugvic/av_meetings.html	The meetings alternate between Technical presentations in the odd numbered months and purely social occasions in the even numbered months. Some attempt is made to fit other AUUG activities into the schedule with minimum disruption.
PERTH	The Victoria League 276 Onslow Road Shenton Park	Meeting commences at 6.15pm
SYDNEY	The Wesley Centre Pitt Street Sydney 2000	

Up-to-date information is available by calling AUUG on 1800 625 655.

Application for Institutional Membership

Section A: MEMBER DETAILS

The primary contact holds the full member voting rights and two designated representatives will be given membership rates to AUUG activities including chapter activities. In addition to the primary and two representatives, additional representatives can be included at a rate of $88 each. **Please attach a separate sheet with details of all representatives to be included with your membership.**

NAME OF ORGANISATION: ______________________________

Primary Contact

Surname ____________________ First Name ____________________

Title: ____________________ Position ____________________

Address ____

Suburb ____________________ State ____________ Postcode ____________

Telephone: Business ____________________ Facsimile ____________________

Email ____________________ Local Chapter Preference ____________________

Section B: MEMBERSHIP INFORMATION.

Renewal/New Institutional Membership of AUUG *(including Primary and Two Representatives)*	□	$429.00
Surcharge for International Air Mail	□	$132.00
Additional Representatives Number	□	@ $88.00

Rates valid as at 1 March 2000. Memberships valid through to 30 June 2001 and include 10% GST.

Section C: PAYMENT

Cheques to be made payable to **AUUG Inc** *(Payment in Australian Dollars only)*

For all overseas applications, a bank draft drawn on an Australian bank is required. Please do not send purchase orders.

-OR-

□ Please debit my credit card for A$____________

□ Bankcard □ Visa □ Mastercard

Name on Card ____________________

Card Number ____________________

Expiry Date ____________________

Signature ____________________

Please mail completed form with payment to:

Reply Paid 66
AUUG Membership Secretary
PO Box 366
KENSINGTON NSW 2033

Or Fax to:

AUUG Inc
(02) 8824 9522

Section D: MAILING LISTS

AUUG mailing lists are sometimes made available to vendors. Please indicate whether you wish your name to be included on these lists:

□ Yes □ No

Section E: AGREEMENT

I/We agree that this membership will be subject to 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: ____________________

Title: ____________________

Date: ____________________

AUUG Secretariat Use

Chq: bank ____________ *bsb* ____________

A/C: ____________ *#* ____________

Date: ____________ *$* ____________

Initial: ____________ *Date Processed:* ____________

Membership#: ____________

AUUG Inc

UNIX® AND OPEN SYSTEMS USERS

Membership Application

AUUG Inc
PO Box 366, Kensington NSW 2033, Australia

Tel:	(02) 8824 9511
Free Call:	1 800 625 655
Fax:	(02) 8824 9522
email:	auug@auug.org.au

ACN A00 166 36N (incorporated in Victoria)

http://www.auug.org.au

AUUG Inc is the Australian UNIX and Open Systems User Group, providing users with relevant and practical information, services and education through co-operation among users.

AUUG OFFERS SOMETHING FOR YOU!!

AUUGN

Technical Newsletter

AUUG's quarterly publication, keeping you up to date with the world of UNIX and open systems.

Education

Tutorials
Workshops

Events.....Events......Events

- Annual Conference & Exhibition
- Overseas Speakers
- Local Conferences
- Roadshows
- Monthly Meetings

DISCOUNTS

to all AUUG events and education.

Reciprocal arrangements with overseas affiliates.

Discounts with various internet service providers, software, publications and more...!!

Connections

- Newsgroup aus.org.auug

Application for Individual or Student Membership

Section A: PERSONAL DETAILS

Surname ______________________ First Name ______________________
Title: ______________________ Position ______________________
Organisation ______________________
Address ______________________
Suburb ______________________ State ______________ Postcode __________
Telephone: Business ______________________ Private ______________________
Facsimile: ______________________ E-mail ______________________

Section B: MEMBERSHIP INFORMATION

Please indicate whether you require Student or Individual Membership by ticking the appropriate box.

RENEWAL/NEW INDIVIDUAL MEMBERSHIP

Renewal/New Membership of AUUG □ $110.00

RENEWAL/NEW STUDENT MEMBERSHIP

Renewal/New Membership of AUUG □ $27.50
(Please complete Section C)

SURCHARGE FOR INTERNATIONAL AIR MAIL □ $66.00

Rates valid as at 1 March 2000. Memberships valid through to 30 June 2001 and include 10% GST.

Section C: STUDENT MEMBER CERTIFICATION

For those applying for Student Membership, this section is required to be completed by a member of the academic staff.

I hereby certify that the applicant on this form is a full time student and that the following details are correct.

NAME OF STUDENT: ______________________
INSTITUTION: ______________________
STUDENT NUMBER: ______________________
SIGNED: ______________________
NAME: ______________________
TITLE: ______________________
DATE: ______________________

Section D: LOCAL CHAPTER PREFERENCE

By default your closest local chapter will receive a percentage of your membership fee in support of local activities. Should you choose to elect another chapter to be the recipient please specify here:

____

Section E: MAILING LISTS

AUUG mailing lists are sometimes made available to vendors. Please indicate whether you wish your name to be included on these lists:

□ Yes □ No

Section F: PAYMENT

Cheques to be made payable to **AUUG Inc**
(Payment in Australian Dollars only)

For all overseas applications, a bank draft drawn on an Australian bank is required. Please do not send purchase orders.

-OR-

□ Please debit my credit card for A$__________

□ Bankcard □ Visa □ Mastercard

Name on Card ______________________
Card Number ______________________
Expiry Date ______________________
Signature ______________________

Please mail completed form with payment to:

Reply Paid 66
AUUG Membership Secretary
PO Box 366
KENSINGTON NSW 2033
AUSTRALIA

Or Fax to:

AUUG Inc
(02) 8824 9522

Section G: AGREEMENT

I agree that this membership will be subject to 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.

Signed: ______________________
Date: ______________________

AUUG Secretariat Use

Chq: bank __________ *bsb* __________
A/C: __________ *#* __________
Date: __________ *$* __________
Initial: __________ *Date Processed:* __________
Membership#: __________