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 Rd.
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

Con Zymaris
auugn@auug.org.au

The latest IDC figures on new server operating system installs in North America are out. I personally don't put much value on these figures, but they are bandied about by the trade press as if gospel. It is therefore worthwhile analysing the figures, to be versed in them, for when next they arise as a topic of conversation. The latest figures show that Windows is tied with Unix/Linux for first place (by volume) of the server market. Windows is on 41%, Linux 27% and Unix 14%. Linux is still the fastest growing OS. Unix is by far the biggest slice by value.

As you probably know, what these figures do not show, are the vast majority of Linux and free (BSD) Unix installs which will never appear on the blip--screens over at IDC. I imagine that for every copy of boxed Linux sold/shipped, there were 20 installs off cable-modem, cheap/free CDs, Pocketbooks and loaners from friends. Same would go for the BSDs through anon-ftp and CVSup. Real tip-of-iceberg stuff.

I guess the other thing that the figures don't show is this: besides the Microsoft Windows clan of operating systems, all other major OS types (including the BSD-based MacOS X) are derived or evolved from the genus *Unix*. This is indeed a major affirmation of what Thompson and Ritchie started building over 30 years ago. What this also means is that the Australian Unix User's Group (and by association AUUGN) has more potential to contribute to the advancement of computing systems in the future, than it has in the past. So, let's get to it.

Cheers,

Con

Thanks to our Sponsor:

Contribution Deadlines for AUUGN in 2001

Volume 22 • Number 2 – June 2001: **May 17th, 2001**

Volume 22 • Number 3 – September 2001: **August 17th, 2001**

Volume 22 • Number 4 – December 2001: **November 17th, 2001**

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

President's Column

David Purdue
David.Purdue@auug.org.au

I have read in a couple of newspapers lately that Mayor Siebold Hartkamp (of a Dutch town that the paper I have here neglected to mention) has offered a job interview to OnTheFly.

For those who do not instantly recognise the name, OnTheFly is the author of the Anna Kournikova virus, who was recently arrested. Apparently Mayor Hartkamp has no particular job in mind, but thinks that OnTheFly would be a good expert on computer security.

This is not the first time I have heard of hackers or crackers being offered jobs as computer security experts. It seems that there are organisations that agree with the Mayor that "former poachers make the best game wardens."

So I thought to myself, "These people are successful executives and politicians, they can't be wrong, can they? I will apply this thinking in my activities."

I recently had reason to travel to one of the southern islands of the Philippines. As you may have seen on the news, this is a rather dangerous place at the moment, especially for Westerners. Since I have no wish to be killed, I thought I had better get a bodyguard. And, applying the "former poachers make the best game wardens" (or FPMTBGW) principle, I hired Rocko "Bone Crusher" McThugsby*.

Rocko has killed 12 people in his career, 7 professionally, 4 in bar room brawls, and one totally by accident ("and I can tell you, no one was more embarrassed than I was," he says of the incident). Obviously, Rocko would be able to use his expertise in killing to stop anyone killing me.

It was just after we landed in Manila that the first unforeseen consequence of the FPMTBGW principle became apparent. Fortunately my cheque cleared in time to avoid what Rocko referred to as "the undesirable outcomes of a failed contract re-negotiation."

When we entered the jungles "Bone Crusher" let slip that he had only ever killed anyone by grabbing them around the throat and breaking their neck. When I asked him how you "accidentally" grab someone by the throat and break their neck, he became a bit shifty, and changed the subject.

Mind you, this is where the FPMTBGW principle came into its own - in my whole time in the Philippines not one person tried to grab me by the throat and break my neck. However, when people started shooting at me, Rocko was all at sea.

Despite all this I survived the trip, and on my return I read that a good way to validate your network security is to hire someone to perform a penetration test - apparently the "fat five" will charge you a fortune for attempting to hack in to your system.

Well, I know the big consultancies would not waste their clients money, so I got in touch with Rocko again to check that my personal defences were all in order. That was six weeks ago, and my neck is feeling much better now, thank you.

My point, if it is not clear, is that the knowledge of how to crack a vulnerable system is very different to the skills required to stop a system from being vulnerable. Also, failing to crack a system is not nearly as good an assurance of system security as an analysis of that system's vulnerabilities.

AUUG Symposia

The AUUG One Day Symposium programme is going from strength to strength, and we are planning to hold four symposia this year.

The UNIX Systems Administration Symposium will be held in Brisbane in early May, and the third Australian Open Source Symposium will be held in Canberra in late May. Later in the year we will have the return of the Security Symposium, and the introduction of the Network Technologies symposium.

If you have any comments about the programme, good or bad, please send them to <talk@auug.org.au>. If you would like to volunteer to help with organisation, or would perhaps like to sponsor one of the symposia, then please write to <auugexec@auug.org.au>.

* Rocko "Bone Crusher" McThugsby is a figment of the President's imagination.** Any resemblance to any real person living or dead is unintentional and purely coincidental.

** So there to those who thought the President didn't have one.

/var/spool/mail/auugn

Editor: <auugn@auug.org.au>

What follows are some of the AUUG-related email exchanges which have crossed your editor's desk in recent times. If you want to contribute to the list, mail *Majordomo@tip.net.au* with:

```
subscribe talk Your Name <your@email.com.au>
```

To: talk@tip.net.au
From: Scott Colwell <scolwell@uu.net>
Subject: Supportive of AUUG taking a public position on CPRM

Hi all,

I'm supportive of AUUG taking a position based on technical issues.

If the committe decide to also take a position on ethical grounds, they should ensure that the arguments expressed clearly distinguish between the two positions.

Scott Colwell

- -

Date: Sat, 6 Jan 2001 13:09:21 +1100 (EST)
From: David J N Begley <d.begley@uws.edu.au>
To: <talk@tip.net.au>
Subject:AUUG's position on the proposed CPRM extensions
to the ATA standard

Earlier today, owner-auug-announce@auug.org.au wrote:

> *The AUUG executive committee has recently been approached by SAGE-AU with information about planned objections to a proposed new extension to the ATA standard (previously known as IDE) for disk drives. CPRM (Content Protection for Recordable Media) is an encryption scheme similar in concept to the CSS scheme used on DVDs.*

Most people would already be aware of the "added features" that DVDs include (Macrovision, CSS, &c.); it looks like Hollywood is really on the warpath because at the same time that they're pushing to get CPRM included in storage devices:

"4C retreats in Copy Protection storm"
http://www.theregister.co.uk/content/2/15797.html

they're also pushing for similar restrictions in digital television (in the U.S. at least):

"Wary of a Video Napster, Hollywood Plots a TV Crackdown"
http://www.inside.com/jcs/Story?article_id=19517&pod_id=11

Like any creator, I understand the need for and respect the IP system (patents, trademarks, copyright) - however, this global ceding of control to Los Angeles, California is really irking me. :-(

In this year when Australia is supposed to be celebrating the centenary of its federation, it's seeming more and more pointless being an independent country when your rights, freedoms and laws are being controlled by a few cities in a country on the other side of the planet. :-(

Then there's CSS2, audio DVDs - and after digital phones and television, we haven't even begun the migration towards digital radio yet! Sheesh...

Hrmph

- -

Date: Mon, 8 Jan 2001 12:27:16 +1100 (EST)
From: David J N Begley <d.begley@uws.edu.au>
To: <talk@auug.org.au>
Subject: Re: Darwin (Mac OS X)

Earlier today, Adam Donnison wrote:

> *Given the derivation of the name AUUG, and the current discussion perhaps its time we formulated an answer to the question what is Unix?*

Perhaps this is as good a starting point as any:

http://www.unix-systems.org/what_is_unix.html

Cheers..

dave

- -

Date: Mon, 8 Jan 2001 11:51:35 +1100 (EST)
From: Ben Elliston <bje@redhat.com>
Subject: Re: Darwin (Mac OS X)

davidp wrote:

> *This was a bit of a furphy, because the POSIX compliance was achieved with an add-on that no windows program would seriously use - it was SO SLOW.*
>
> *Perhaps Bill meant it was UNIX in terms of market segment rather than API? ;-)*

Correct me if I'm mistaken here, but I'm always amused at how each new version of Windows NT smells more and more like Unix. Let's see -- there's Kerberos, Telnet logins (which very generously give you one concurrent login session per license!), Windows Terminal Server and an increasing number of familiar Unix shell utilities. Unfortunately, the masses are led to believe that all of this is "innovation".

Cheers, Ben

Annual Election of Officers and General Committee Members

Call For Nominations

Get involved!

AUUG is 26 years old this year. Across this span, we have built a proud history of sharing knowledge, providing member services and, most importantly, creating a community of like minded professionals. Every year, however, brings fresh challenges and new opportunities. As a result, AUUG is in a constant process of evolution; a process of which every member in our association is a part.

The role of AUUG's Officers and General Committee Members is to manage, plan and execute, according to the will of the general membership. This stewardship is not passive, nor is it always easy. However, serving the AUUG community is also immensely rewarding because, simply, our goals matter and we can make a difference.

What should AUUG be doing next year? How can we serve our members and our community better? What great ideas are out there, just waiting for their chance to be tried out? How do we better promte our knowledge and philosophies?

Do you know the answers to some of these questions? Are you the sort of person who knows how to get things done? Or do you know someone like this?

AUUG needs people with fire and clue. Help make AUUG the kind of association you want it to be -- nominate the best people for election to our Management Committee. If you would like to know more about serving on the Management Committee, email the current committee at:

auugexec@auug.org.au

You need to be nominated by three voting members of AUUG (that is, either Individual Members or Institutional Members), and you must be an Individual Member yourself.

What? You can't find three members to nominate you? Send in your nomination form anyway - we'll find someone to sign it!

In order to nominate a member for the Committee, please copy and fill out the following official nomination form, and send it to the AUUG Secretary. All nominations must be received by:

14th April 2001

You can send in nominations by fax or mail:

Fax: (02) 8824 9522

AUUG Inc.
PO Box 366
Kensington NSW 2033
Australia

Nominees are encouraged to include a policy statement of up to two hundred words. This statement will be circulated to members with election materials, and is intended to assist them in making voting decisions.
The Secretary reserves the right to truncate statements at two hundred words (as measured by "wc") in order to minimise election expenses.

AUUG Inc. 2001 Annual Election Nomination Form

We,

(1) Name: AUUG Member #: and

(2) Name: AUUG Member #: and

(3) Name: AUUG Member #:

being current financial members of AUUG Inc do hereby nominate:

_____

for the following position(s):

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

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

Signed (1) Date:

Signed (2) Date:

Signed (3) Date

I, Name: _____ AUUG Member #: _____

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

Signed: _____ Date: _____

Public Notices

Upcoming Conferences

March 26-29, 2001
O'Reilly Conference on Enterprise Java
Westin Hotel -- Santa Clara, California

March 26-28
3rd USENIX Symposium on Internet Technologies & Systems
San Francisco, CA

March 30-31
Linux 2.5 Kernel Developers Summit
San Jose, CA

April 23-24
Java Virtual Machine Research and Technology Symposium
Monterey, CA

June 25-30
USENIX Annual Technical Conference
Boston, MA

July 23-27, 2001
O'Reilly Open Source Convention
in San Diego, California

July 30 - Aug. 3
The Systems and Network Administration Conference
Dallas, TX

August 13-17
10th USENIX Security Symposium
Washington, D.C.

September 17-20, 2001
The O'Reilly Peer-to-Peer Conference
Omni Shoreham Hotel, Washington, DC

November 6-10
5th Annual Linux Showcase and Conference
Oakland, CA

December 2-7
15th Systems Administration Conference (LISA 2001)
San Diego, CA

My Home Network (March 2001)

Author: Frank Crawford
<frank@crawford.emu.id.au>

One of the nice things about having a home network is the number of things that the rest of my family can do with it, and all without the arguments over who is using the computer today. Of course it doesn't stop arguments over who wants the "best" computer.

However, at present I am amused by the sight of my two children "talking" to each other via a system located somewhere in the USA! What is also interesting is that there is nothing new in computing, while they think that "Instant Messaging" via MSN Messenger is the latest and greatest, I was doing the same thing with "talk" over a decade ago (and not going via the USA as well :-)).

While there are a lot of chat programs available, if I want to communicate with my children, it is necessary to be "compatible with Microsoft" (a traitor, aren't I ...). We'll after a bit of study, it turns out that this isn't true, as normal, Microsoft has picked up AOL's Instant Messenger protocol and modified it a bit. After a bit of searching on "freshmeat.net" a few choices turned up. The two best were "GAIM" and "EveryBuddy". In this case I picked EveryBuddy, as it had better support for proxying, although I feel GAIM looked a more mature product.

Given this, I can join in the conversation with my children, or more importantly, check on them from my work (a Unix installation). I even used EveryBuddy to get them to do some other testing for me from a "remote installation" (at least remote from where I was).

Talking about instant messaging, the ability to send messages "instantly" to my mobile phone has been invaluable to me. While many people wander around with a pager for receiving messages and a mobile to respond, the use of SMS (Short Messaging Service) virtually merges the two into a single object. While most people look at sending SMS messages from mobile to mobile, my main use is for computers to send me messages about certain events (even reminders to pick up children :-)).

The requirements to send SMS messages are many and varied, and to some extent depend on who you want to send messages to. The simplest method is to mail to some site that will send out the SMS message for you. Optus has such an address, which is reported to work for all mobiles, but I'm not sure it is for general use. RedRock (see below) also have such a service, but it costs money. It also is probably the least effective for time critical messages, as you have to contend with the vagaries of both mail and the phone system.

The next most common method is using the same software as for normal pagers. Long ago paging companies standardised on a communication protocol for use over a modem to submit pages. This is called variously, the Telocator Alpha-numeric Protocol (TAP) or IXO or PET, all of these are basically the same. This is supported by a number of network programs, in particular "hylafax", "qpage" and "beepage".

Hylafax is a fairly large package that primarily handles the sending and receiving of faxes, and includes the ability to send pages using the IXO protocol. While I won't go into the setup for hylafax, the additional items you need to set up for paging are a pagermap file ("etc/pagermap") which lists names and numbers to contact, and for each number to dial (i.e. the service not the mobile) an info file with details about the service. All this is explained in the hylafax documentation.

The only general number I have is for Telstra's Mobilenet, for which I have the following in my etc/pagermap file:

```
# Pager mappings
root    018018767/0419123456
frank   018018767/0419789012
jc      018018767/0419123456
```

This file defines an alias (e.g. "root", "frank" or "jc"), the number to dial for the service provider (O18018767) and the mobile number to send the message to (or other PIN for say a real pager, and BTW these aren't real phone numbers). In addition to this you would need the following:

```
&pagerMaxMsgLength:228
&pagerPassword:"mnmail"
&pagingProtocol:"ixo"
```

in the file "info/6118018767", as well as other possible definitions for faxing.

These definitions are fairly obvious, and include the maximum message length, the paging protocol ("ixo") and an authentication password ("mnmail" - n.b. this is the real password, as is the access number given above, apparently Telstra have just a single password for everyone).

One interesting thing, most Australian paging companies require a password or some other form of authentication, while most USA companies don't. This means that many simple scripts and programs you find, do not have the provision for sending a password. It is easy to hack in, contact me directly if you want details.

The cost for sending such a page is the same as a normal phone call, but it is restricted to only sending to Mobilenet numbers. Optus also have a service called Optus Fleet Page, but requires pre-registration and individual authentication (contact Optus directly if you require details).

Further, a number of other companies also support sending SMS messages, and other pager messages, in particular Link Paging and Hutchinson/Orange, but again you need pre-registration. They do however,

allow you to send to mobiles for a number of different telcos. Probably the best independent service for this is RedRock.com.au and their messagenet service. This allows messages to be sent by email or IXO. See http://www.messagenet.com.au for more details.

Other software that I've found designed to specifically send pages are "qpage" (see http://www.qpage.org), which is probably the standard in the Unix environment, and "beepage" (http://www.beepage.org). Both of these are easy to set up and have different advantages and disadvantages.

While systems that send pages via a modem are fine, in our networked world we need to be able to send pages from any system, and require a network server to handle the paging for you. In fact, there is a standard protocol (SNPP, port 444/tcp) and an RFC (RFC 1861) defining just that. This is supported by both hylafax and qpage (beepage has its own, non-standard network protocol) and both have clients to implement it. There are also a number of shareware and commercial clients available for Microsoft systems, and even a Perl module (part of libnet). It also appears that in the USA there are some companies offering SNPP servers for their clients, unfortunately I haven't found any in Australia. On the other hand, SNPP is an excellent facility to support within an organisation.

Of course today, the Internet means the Web, and there is nothing else (:-)), and the various telcos know that. SMS is a product that the telcos have started to market and to make it available to the general public, Web interfaces for SMS messaging have been developed. Of course, this is fine for people, but not necessarily for automation. Luckily enough, Australia is a bit behind here and a number of overseas developers have already addressed the issue. Again a quick search of freshmeat.net turned up a number of possibilities, from quick perl hacks to a well developed system, called "smssend".

Rather than just connecting to a single service, smssend provides a parser and a scripting language to allow you to navigate a web site and send your message. It comes with a fair few scripts and is reasonably easy to develop new ones. It even has features to automatically download new scripts from a remote site. You can pick all this up from http://zekiller.skytech.org/smssend_menu_en.html.

While SMS Web pages seem fairly common overseas, particularly Europe, they are new within Australia. There are only two that I know of: http://www.info2you.com.au, run by Optus, and http://www.freeSMS.com.au, run by RedRock. Info2you currently can only connect to Optus and Vodafone mobiles (but there are plans to connect to the others), while freeSMS can connect to most carriers, is free, but includes advertising in the message. Both require a mobile phone number to register.

To generate a script for smssend you need to be able to trace the flow of traffic, including a fair few details of HTTP (e.g. the POST data). Just looking at the screen isn't sufficient for anything more than a single form. This is where the power of your own home network comes in. There are a number of tools that will trace network traffic for you, allowing you to capture what is going on below the surface. In my case I used a product called "ethereal" to capture my connection, and later to analyse it. It has a nice feature where it will show you a "TCP session", rather than forcing you to try and put them together yourself. You could also use something like "tcpdump" but that is a bit harder to trace the packet data.

Anyway, once you have traced the data you need for your communication, it is fairly easy to create a script for use with smssend. I've already done this for info2you, and will shortly do it for freeSMS (I only found them while writing this article). Both of these scripts I'll send back to the maintainer for smssend for inclusion in future releases. However, to get you started, I've included the info2you script below. To use it you would run the command:

```
smssend info2you 0401234567 mypwd
0401765432 "Hello there"
```

to send a message from your mobile account (0401234567) to a friends phone (0401765432). You would be billed for the call, but as a side effect, it would register as coming from your phone, and they could reply back to you. You can send to multiple people with the info2you site (this isn't true for all SMS sites), just separate the numbers by commas (','). The script for smssend is in "info2you.sms" and is:

```
## Info2You provider file - www.info2you.com.au
# Version 1.00
# by Frank Crawford <Frank.Crawford@ac3.com.au>
# Error codes:
#       1 -> Bad Login/Password
#       2 -> Invalid number
#       3 -> Error sending message

NbParams 4
%Login : Your login from Info2You (your Optus GSM
number)
%Password Hidden : Your password
%Tel : GSM number of recipient
%Message Size=160 Convert : Your message

# Logging into Info2You
GetURL http://www.info2you.com.au
GO

PostURL http://www.info2you.com.au/cgi-
bin/sms_service/html_sms_services.cgi
Referer http://www.info2you.com.au
PostData xyz_msisdn=\%Login%&password=\%Password%
Search Please check your details and login again
ErrorMsg 1 Bad login and/or password
GO

GetURL http://www.info2you.com.au/cgi-
bin/send_sms/send_sms.cgi
Referer http://www.info2you.com.au/cgi-
bin/sms_service/html_sms_services.cgi
GO

# Sending message
PostURL http://www.info2you.com.au/cgi-
bin/send_sms/send_sms_action.cgi
Referer http://www.info2you.com.au/cgi-
bin/send_sms/send_sms.cgi
PostData
mobile_phone=\%Tel%&max_count=200%0A&message=\%Me-
ssage%&counter=0
Search has been successfully sent
PrintMsg Message successfully sent
ElseSearch The mobile number you have entered is
invalid.
ErrorMsg 2 Invalid number
Else
ErrorMsg 3 Error sending message
GO

GetURL http://www.info2you.com.au/cgi-
bin/logout/logout.cgi
```

```
Referer http://www.info2you.com.au/cgi-
bin/send_sms/send_sms_action.cgi
GO
```

Of course this does depend on the format of the web site which could be changed at will by Optus, so use at your own risk. If it does fail, let me know and I'll look into it.

Just as an aside, almost all the products I've mentioned have additional interfaces available, from web interfaces for qpage, etc, to Gnome GUI's for smssend. This means that you can setup something for any environment for your family to use (but then we are back to user interfaces and not computer interfaces).

Finally, to send messages, you could hook a mobile phone directly to your PC. There are a number of packages available to do this, although I'd suspect the cost of dedicating a mobile to your PC is out for most home networks. Of course, if you have a business, you may well look at it, I think it makes you much more independent.

So, what do we have here, a mishmash of different communication methods. If you want to be able to send messages to your mobile, free of charge, you can either use a modem and TAP to send to a Telstra Mobilenet phone, or the info2you web site to send to Optus or Vodafone, or be willing to see ads in your message and get access to all of them. Which one suits you depends on your equipment, your provider and who you want to send to.

Have fun.

National Linux Installfest: 2001

AUUGN would like to draw your attention to the preparations presently underway for this years National Installfest. The Installfests have become a major part of the yearly Linux event calendar, often attracting several hundred participants; many of them totally new to the ways of Unix. As such, we feel that events like this, which get national IT media coverage, are a great way to spread the word about the qualities and ethos of the Unix platform. If you have the skills, the enthusiasm or both, please sign up to support your State's LUG.

Details

from: Sarah Bolderoff <sarah@cs.unisa.edu.au>

The date is the 25th of August 2001.

The mailing list for national scale organisation: installfest-org@auug.org.au It's a good idea for local user groups to have their own mailing list for local organisation.

The web site will be at installfest.linux.org.au

AUUG is willing to provide support for user groups that wish to be involved in the installfest but aren't incorporated and don't have insurance.

For information on last years installfest go to: www.linux.org.au/installfest

The 5 points on running an installfest can be found at www.linux.org.au/installfest/5points/

I would like to put together a list of participating lugs, so people/lugs interested in joining in the installfest fun, can email me, sarah@cs.unisa.edu.au

AUUG Corporate Members

as at 1 March 2001

- Andersen Consulting
- Aurema Pty Ltd
- Australian Bureau of Statistics
- Australian Industry Group
- Australian Taxation Office
- Australian Water Technologies P/L
- BHP Information Technology
- British Aerospace Australia
- Bureau of Meteorology
- C.I.S.R.A.
- Cape Grim B.A.P.S
- Central Queensland University
- Central Sydney Area Health Service
- Centrelink
- CITEC
- Commercial Dynamics
- Commonwealth Steel Company
- Computer Science, Australian Defence Force Academy
- Computing Services, Dept Premier & Cabinet
- Corinthian Industries (Holdings) Pty Ltd
- Corporate Express Australia Limited
- Crane Distribution Limited
- CSC Australia Pty. Ltd.
- 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
- Energex
- eSec Limited
- Everything Linux
- Fulcrum Consulting Group
- G.James Australia Pty. Ltd.
- HIH Insurance
- HIH Winterthur
- IP Australia
- IT Services Centre, ADFA
- Land and Property Information, NSW
- LPINSW
- Macquarie University
- Mercantile Mutual Holdings
- Motorola Australia Software Centre
- Multibase WebAustralis Pty Limited
- Museum Victoria
- Namadgi Systems Pty Ltd
- Nokia Australia
- 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
- TAB Queensland Limited
- Tellurian Pty. Ltd.
- The University of Western Australia
- Thiess Contractors Pty Ltd
- Tower Technology Pty. Ltd.
- University of Melbourne
- University of New South Wales
- University of Sydney
- University of Technology, Sydney
- Victoria University of Technology
- Westrail
- Workcover Queensland

UNIX Systems Administration Symposium

Call for Participation

Continuing the AUUG Inc. series of one day symposia, we will be holding the inaugral UNIX Systems Administration Symposium in Brisbane on Friday, 4th May 2001. The event is being supported by SAGE-AU.
The goal of this symposium is to promote the sharing of information and experience among systems administrators. In line with the aims of AUUG Inc., we are concentrating on UNIX and UNIX-like operating systems, including Linux, *BSD, Solaris, HPUX and AIX.
As for other AUUG symposia, we do not require formal papers. Instead, we are calling for well prepared informal presentations that are both timely and interesting.
Particular topics we are looking for:

- Large Scale UNIX/UNIX in the glass house.
- UNIX as a desktop O/S.
- Distributed Systems Administration.
- Important tools and software.

Presentations should be 30 minutes long, including question time.

TIMETABLE:

- Abstracts (around 200 words) are due by Friday, 7th April 2001.
- Symposium held on Friday, 4th May 2001.

Presenters will recieve free registration.

Please email submissions to Sarah.Bolderoff@auug.org.au.

VENUE:

The symposium will be held at the Department of Primary Industry Conference Centre, 80 Ann Street, Brisbane.

The UNIX Systems Admininistration Symposium is proudly supported by AUUG Inc and SAGE-AU.

Call for Papers AUUG 2001 - Always On and Everywhere

The AUUG Annual Conference will be held in Sydney, Australia, 26, 27 and 28 September 2001.

The Conference will be preceded by three days of tutorials, to be held on 23, 24 and 25 September 2001.

The Programme Committee invites proposals for papers and tutorials relating to:

- Security in the Enterprise
- Applications made possible by Open Source
- Technical aspects of Computing.
- Networking in the Enterprise.
- Business Experience and Case Studies
- Open Source projects
- Business cases for Open Source
- Technical aspects of Unix, Linux, and BSD variants
- Open Systems or other operating systems
- Computer Security
- Performance Management and Measurement
- Networking, Internet (including the World Wide Web)

Presentations may be given as tutorials, technical papers, or management studies. Technical papers are designed for those who need in-depth knowledge, whereas management studies present case studies of real-life experiences in the conference's fields of interest.

A written paper, for inclusion in the conference proceedings must accompany all presentations.

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 timetabled in the conference and speakers should indicate their willingness to participate, and may like to suggest panel topics.

Tutorials, which may be of either a technical or management orientation, provide a more thorough presentation, of either a half-day or full-day duration.

Representing the largest Technical Computing event held in Australia, this conference offers an unparalleled opportunity to present your ideas and experiences to an audience with a major influence on the direction of Computing in Australia.

Submission Guidelines

Those proposing to submit papers should submit an extended abstract (1-3 pages) and a brief biography, and clearly indicate their preferred presentation format. Those submitting tutorial proposals should submit an outline of the tutorial and a brief biography, and clearly indicate whether the tutorial is of half-day or full-day duration.

Speaker Incentives

Presenters of papers are afforded complimentary conference registration. Tutorial presenters may select 25% of the profit of their session OR complimentary conference registration. Past experience suggests that a successful tutorial session of either duration can generate a reasonable return to the presenter. Please note that with the GST changes to tax legislation we will be requiring the presentation of a tax invoice (which we will assist in producing) containing an ABN for your payment. If that is not provided then tax will have to be withheld from your payment.

Important Dates

Abstracts/Proposals Due
13 July 2001

Authors notified
27 July 2001

Final copy due
24 August 2001

Tutorials
23-25 September 2001

Conference
26-28 September 2001

Proposals should be sent to:
AUUG Inc.
PO Box 366
Kensington NSW 2033
AUSTRALIA
Email: auug2001prog@auug.org.au
Phone: 1800 625 655 or +61 2 8824 9511
Fax: +61 2 8824 9522

AUUG 2001 - Always On and Everywhere Sponsorship Opportunities

Diamond Sponsorship

Cost
A$10,000 (plus 10% GST applicable)

Includes
- 2 complimentary registrations for the conference
- 2 complimentary invitations for the cocktail reception
- 2 complimentary invitations for the conference dinner
- logo displayed in conference plenary hall
- acknowledged on all appropriate occasions in both print and verbally
- small display area
- listed and identified as a sponsor in the conference brochure and final programme, with company description
- logo displayed and identified as a sponsor on the AUUG website with a link back to organisation's site

Choice of

Conference Brochure
- wide distribution to key decision makers
- areas of exclusive advertising
- immediate impact prior to the conference

Welcome Reception
- prestigious event allowing sponsor to make first impression on the delegates
- reception identified as being sponsored by the XYZ company on all printed material
- signage on the evening
- opportunity to address delegates

Conference Dinner
- dinner identified as being sponsored by the XYZ company
- name printed on dinner menu
- opportunity to distribute mementos and address to the audience
- banner identifying the sponsoring company

Platinum Sponsorship

Cost

A$7,500 (plus 10% GST applicable)

Includes
- 1 complimentary registration for the conference
- 2 complimentary invitations for the cocktail reception
- 2 complimentary invitations for the conference dinner
- logo displayed in conference plenary hall
- acknowledged on all appropriate occasions in both print and verbally
- display space
- listed and identified as a sponsor in the conference brochure and the final programme
- logo displayed and identified as a sponsor on the AUUG website with a link back to organisation's site

Choice of

Conference Proceedings
- 2 A4 pages of exclusive advertising
- long term usage and shelf life as it is a reference material
- Tee-Shirts
- offering long term usage and company message to recipient
- Conference Satchel
- Satchel offering long term usage and company message to recipient

Gold Sponsorship

Cost
A$5,000 (plus 10% GST applicable)

Includes
- 1 complimentary invitation for the cocktail reception
- 1 complimentary invitation for the conference dinner
- logo displayed in conference plenary hall
- acknowledged on all appropriate occasions in both print and verbally
- display space available for one day at the conference
- listed and identified as a sponsor in the conference brochure and conference final programme
- logo displayed and identified as a sponsor on the AUUG website with a link back to organisation's site

Choice of

Speakers Reception
- event allowing sponsor to make first impression with speakers
- reception identified as being sponsored by the XYZ company on all printed material
- signage on the evening
- opportunity to address speakers

Pen'n'Paper

- Company logo on pens and writing pads distributed to delegates, offering long term usage and company message to recipient

Lapel Badges
- Company logo on delegate lapel badges, offering company visibility for duration of conference

Keynote Sessions
- opportunity to introduce the keynote session

Silver Sponsorship

Cost
A$2,500 (plus 10% GST applicable)

Includes

- 1 complimentary invitation to the cocktail reception
- logo displayed in conference plenary hall
- acknowledged on all appropriate occasions in both print and verbally
- rack space for promotional material
- listed and identified as a sponsor in the conference brochure and the conference final programme
- logo displayed and identified as a sponsor on the AUUG website with a link back to organisation's site

Choice of

Conference folder insert
- individual inserts in conference satchels

Advertisement
- A4 sized advertisement in conference proceedings

Registration desk handouts
- Promotional material to be available to delegates from the Conference Registration Desk

Additional Opportunities

Audio Visual *
Conference Network *
** Contact the AUUG Business Manager, for further details*

Further Information

Further information on these and other options is available from the AUUG Business Manager, Liz Carroll Ph: 1-800-625-655 or +61-2-8824-9511

Email: busmgr@auug.org.au

FreeBSD 4.2 RELEASE

Author: Greg Lehey <grog@lemis.com>

This issue of AUUGN includes a CD-ROM of FreeBSD 4.2. The CDs should have been distributed with the December issue, but thanks to the Customs, they arrived at the publishers literally a couple of hours too late.

The December issue of AUUGN contained the complete installation notes for this CD. If you don't have the issue handy, they are also on the CD in the file INSTALL.TXT. For the experienced, though, the procedure is simple. The following text is reproduced with permission from my book "The Complete FreeBSD":

- If you have another operating system on the machine, for example Microsoft, and you want to keep it,

    1. Make a backup! There's every possibility of erasing your data, and there's absolutely no reason why you should take the risk.
    2. Repartition your disk with FIPS, which is available on the CD at tools/fips.exe.

- Insert the CD-ROM in the drive before booting.

- Boot the FreeBSD system. The easiest way is to boot directly from the CD.

- Select the Custom installation: it's the only one which allows you to back up a step if you make a mistake.

- If you have repartitioned with FIPS, in the partition editor, delete only the second primary Microsoft slice. The first primary Microsoft partition contains your Microsoft data, and if there is an extended Microsoft partition, it will also contain your Microsoft data. Then create a FreeBSD slice in the space that has been freed.

- Otherwise delete whatever you may find in the partition editor and create new FreeBSD slices.

- On exiting from the partition editor, select the BootMgr MBR.

- In the disk label editor, select the FreeBSD slice. If you proceeded as above, it should be empty, but if it contains existing UNIX partitions, delete them. If you're not too worried about the exact size of the partitions, select automatically generated disk labels.

- Alternatively, if you want to specify your file systems yourself, start on the basis of a root file system with 50 MB, a swap partition with 256 MB, and allocate the rest of the space on the disk to the /usr file system.

- Note particularly that, if you don't create a /var file system, you'll need to create a symlink later on.

- Choose the distributions you want. Note that in this menu, you choose the distribution by pressing the space bar, not the Enter key.

- Select CD-ROM as the installation medium.

- If you intend to run the X window system, select the installation now. It's much easier than doing it after the system is up and running.

- Confirm installation. The system will be installed.

linux.conf.au

Author: Andre Joanisse <andrej@marpware.com>

[Editors Note: Those sharp readers among you will no doubt note that we have two separate articles of this conference in this issue. Andre's piece gives a more traditional conference write-up from a Linux insider's perspective, whereas Greg Lehey's piece has more of the flavour of an interested by-stander's. Regardless, I'm sure that between the two of them, you will have a good, stereoscopic view of the conference ;-)]

linux.conf.au 17th-20th January 2001
UNSW, Sydney.
http://www.linux.conf.au/

It finally is here. The linux.conf.au conference gets under way on a rainy day at the University of New South Wales. The conference is the follow up to the CALU conference of 1999 in Melbourne. It is not for the Linux newbie or for the "suits". There is only a few low--key sponsors and no exhibition floor. I noticed very few notebooks running Windows. Even those that ran Windows were quickly hidden! So it was just Linux served raw. :-)

It was a well attended conference with lots of quality presenters from Australia and overseas. Lots of Linux folks walking around in jeans and T-shirts, sharing ideas, finally meeting fellow "tuxes" in the flesh. Over the four days of the conference there was a continuous buzz around the place.

The debian and linux.org.au T-shirts proved to be a very popular purchase.

It was hard at times to pick which session to attend but choices had to be made. Here is a brief description of the various sessions that I managed to attend.

Wednesday 17/01/01

Introduction to Using DocBook for Application Documentation by Malcolm Tredinnick

Not knowing too much about DocBook put me at a disadvantage here but as the session went on, I started seeing the value in this interesting tool. DocBook is currently at v3.1 with v4 due soon, it may already be out by the time you read this. It uses SGML. v5 should be XML based. The 2.4 kernel documentation makes use of DocBook as does Gnome and many LDP's. Various tools support DocBook, openjade and happydoc to name a couple. For more info on Docbook, have a look at http://www.docbook.org/

Bonobo, the GNOME Component Model
by George Lebl and Maciej Stachowiak

Bonobo is layered on corba, it is similar to MS-COM. A history of how bonobo got to be was presented. Bonobo basically makes corba easy to use. Bonobo is now used in various applications such as Nautilus, Evolution, StarOffice (for Gnome 2.0) and GIMP 2.0, just to name a few.

BoFs

The BoFs were a great place to exchange ideas and catch with friends and colleagues. I sat in on the Calendering and Scheduling BoF. Various tools that support or would support the iCalendar standard were discussed. RFC2445 covers the ical standard. Skud talked about the Reefknot project that she is involved with. It's a perl toolkit compliant with iCalendar. More info can be found at http://reefknot.sourceforge.net/.

Thursday 18/01/01

Keynote - Alan Cox

Alan showed up disguised as a tourist so that he could come in to Australia to give us his secret mission briefing on World Domination. He discussed various new features of kernel 2.4 and that work on 2.5 would not start until 2.4 is stable. ReiserFS should be available in 2.4.1. Lots has changed in 2.4 and he outlines a lot of these. Part of his mission to Australia is also to find out if Australians drink any other beer then XXXX or Fosters!

OpenH323
by Craig Southeren

Here is another quality open source project started here in Australia back in 1998. It is a project to have the H.323 protocol stack available as Open Source. Up to now, the tele/video conferencing industry has been very tightly closed with every company developing their own stack. Craig and others have managed over a short period of time to develop and open source H.323 protocol stack that can connect to all kinds of proprietary systems. More info can be found at http://www.openh323.org/. The project was released under the Mozilla Public License.

Qt/Embedded
by John Ryland

As many of you would know, KDE makes use of the Qt toolkit. Qt/Embedded is aimed at the handheld market. With Linux 2.2, framebuffer driver was added. Qt/Embedded now supports the Compaq iPaq and the Casio Casseopia. Trolltech has done a lot of work for this. Have a look at http://www.trolltech.com/. John gave us a nice demo of his iPaq running Linux, very impressive.

The e-smith Server and Gateway
by Kirrily 'Scud' Robert

Scud is one of the many Ozzies that have moved to sunny Canada! The e-smith server and gateway is a cut down Red Hat distribution with a really nice interface added to the front. It's dead simple to install and even simpler to administer. This distribution is aimed at small/medium size companies that would not have a Linux expert on hand. With e-smith, they can maintain their system with ease. Adding users, printers, shared network files is a breeze via their web interface. Products like these will certainly help replace those hard to manage Microsoft SBE servers. More info at http://www.e-smith.com/

Friday 19/01/01

My notes have become a bit more sparse by now, sorry :-(

Bunyip at the ANU
by Bob Edwards

This is an impressive project that the ANU did to build a super computer with a bunch of Linux boxes. They had a budget of $250k. 80% was spent on PCs and the rest on the network. These were boxes that you could buy from your local reseller. The systems were all the same: Dual 550MHz PII, 3 NIC and 384Mb memory. They won the Gordon Bell prize for price/performance. They project came in at $.92/MFlops/sec. The way they racked all these PC's was with old library shelves. In the end, they had a 192 processor Beowulf cluster. For more info have a look at http://tuz.anu.edu.au/Projects/Bunyip/.

Conference Dinner
Sponsored by Aurema

John "maddog" Hall gave us a very entertaining, yet extremely relevant, talk on Mon&Pop(tm): At Home with Linux. A good time was had by all judging by the buzz in the room. Maddog showed us that it's not just the server that counts. For Linux to really succeed, the desktop has to be conquered and in order to do this, Linux will have to become a lot more user friendly. Once it gets to the point of being used by his Mom & Pop, we will know that Linux is a success. He talked about the Tivo system, which uses Linux, as an example of where Linux can be successfully used.

Saturday 20/01/01

There were a few heavy heads around this morning following the conference dinner. This did not stop most in attending the keynote by Tridge.

Inside the mind of ... TiVo
by Andrew Tridgell

Better know as the creator of Samba, Tridge literally showed us how he got inside the mind of TiVo. TiVo is a device that gets sold in the USA that you can program to record your favourite programs. It does not have a tape, just a hard disk, a motherboard and Linux. Tridge and friends got of few of these babies back from the USA and started doing their own modifications to it. They added a network card, a second hard disk and who knows what else. It was a very entertaining talk and you could see what can be done with a little bit of perseverance! For more info have a look at http://www.linuxcare.com.au/tridge/tivo-ethernet/ or http://tivo.samba.org/

Lunch

This was probably the most impressive display of pizzas that I have ever seen! Pizza Hut and Dominos should have been sponsors! There was more then enough for everyone. I am still to meet a Linux fan that does not like pizza. There were plenty of smiling faces for this lunch.

Conclusion

I must say that this conference exceeded my expectations. If you are into Linux, make sure you attend the event next year. I'm not sure where it's going to be but I'll do my best to be there.

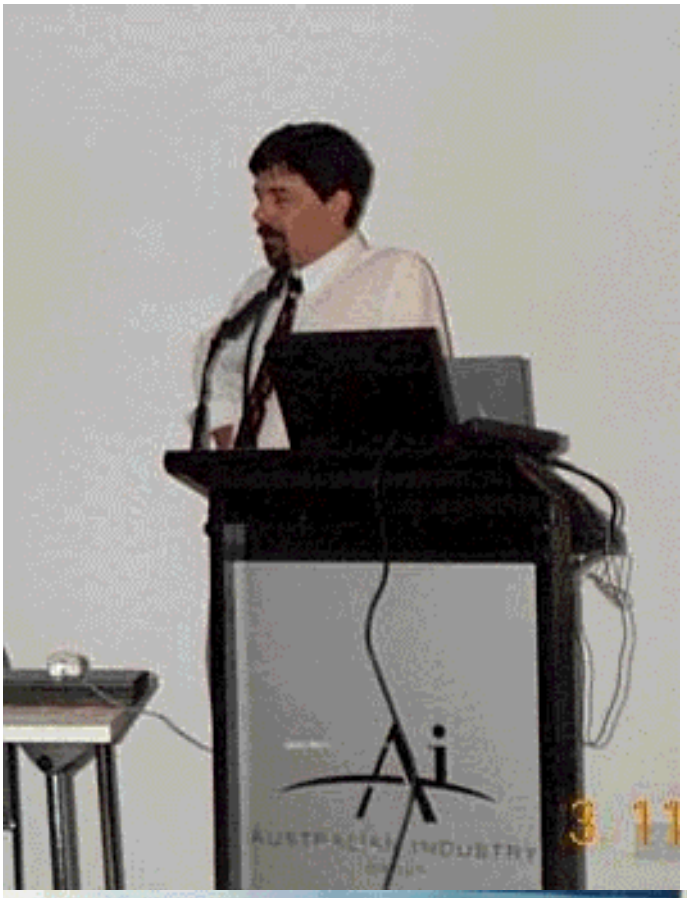
AUUG Security Symposium

Author: Con Zymaris <conz@cyber.com.au> and AUUG Photo Team (!)

While this is a few months old now, the photos have just made an appearance in my in-tray, thus their inclusion here. Enjoy!

Some of the attendees at the AUUG Security Symposium

Systems's Richard Chirgwin



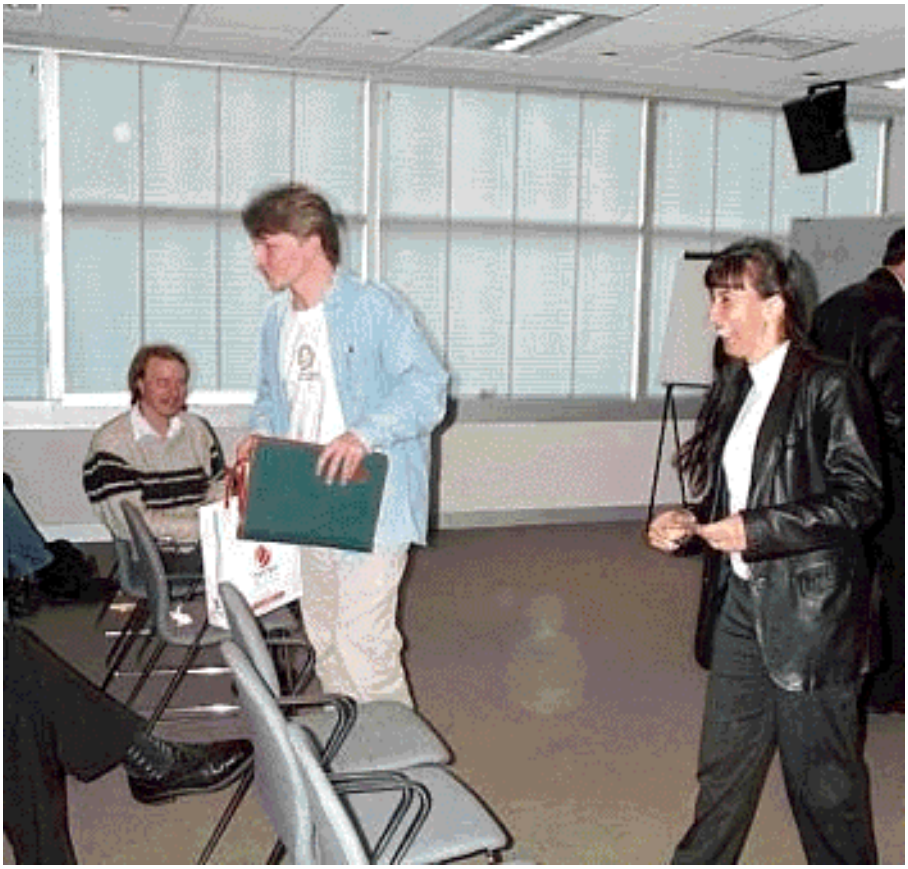
Impromptu Discussion

Liz Carroll with your Editor (no, not *vi*!)

Some of the speakers

Alan Cowie

Alan and Michael Paddon



Public Service Announcement

Thompson, Ritchie and Kernighan admit that Unix was a prank

In an announcement that has stunned the computer industry, Ken Thompson, Dennis Ritchie and Brian Kernighan admitted that the Unix operating system and C programming language created by them is an elaborate prank kept alive for over 20 years. Speaking at the recent UnixWorld Software Development Forum, Thompson revealed the following:

"In 1969, AT&T had just terminated their work with the GE/Honeywell/AT&T Multics project. Brian and I had started work with an early release of Pascal from Professor Niklaus Wirth's ETH Labs in Switzerland and we were impressed with its elegant simplicity and power. Dennis had just finished reading 'Bored of the Rings', a National Lampoon parody of the Tolkien's 'Lord of the Rings' trilogy. As a lark, we decided to do parodies of the Multics environment and Pascal. Dennis and I were responsible for the operating environment. We looked at Multics and designed the new OS to be as complex and cryptic as possible to maximize casual users' frustration levels, calling it
Unix as a parody of Multics, as well as other more risque! allusions. We sold the terse command language to novitiates by telling them that it saved them typing.

Then Dennis and Brian worked on a warped version of Pascal, called 'A'. 'A' looked a lot like Pascal, but elevated the notion of the direct memory address (which Wirth had banished) to the central concept of the "pointer" as an innocuous sounding name for a truly malevolent construct. Brian must be credited with the idea of having absolutely no standard I/O specification: this ensured that at least 50% of the typical commercial program would have to be re-coded when changing hardware platforms.

Brian was also responsible for pitching this lack of I/O as a feature: it allowed us to describe the language as "truly portable". When we found others were actually creating real programs with A, we removed compulsory type-checking on function arguments. Later, we added a notion we called "casting": this allowed the programmer to treat an integer as though it were a 50kb user-defined structure. When we found that some programmers were simply not using pointers, we eliminated the ability to pass structures to functions, enforcing their use in even the simplest applications. We sold this, and many other features, as enhancements to the efficiency of the language. In this way, our prank evolved into B, BCPL, and finally C.

We stopped when we got a clean compile on the following syntax:

```
for(;P("\n"),R-;P("|"))for(e=C;e-;P("_"+(*u++/8)%2
))P("| "+(*u/4)%2);
```

At one time, we joked about selling this to the Soviets to set their computer science progress back 20 or more years.

Unfortunately, AT&T and other US corporations actually began using Unix and C. We decided we'd better keep mum, assuming it was just a passing phase. In fact, it's taken US companies over 20 years to develop enough expertise to generate useful applications using this 1960's technological parody. We are impressed with the tenacity of the general Unix and C programmer. In fact, Brian, Dennis and I have never ourselves attempted to write a commercial application in this.

We feel really guilty about the chaos, confusion and truly awesome programming projects that have resulted from our silly prank so long ago."

Dennis Ritchie said: "What really tore it (just when ADA was catching on), was that Bjarne Stroustrup caught onto our joke. He extended it to further parody Smalltalk. Like us, he was caught by surprise when nobody laughed. So he added multiple inheritance, virtual base classes, and later ...templates. All to no avail. So we now have compilers that can compile 100,000 lines per second, but need to process header files for 25 minutes before they get to the meat of "Hello, World".

Major Unix and C vendors and customers, including AT&T, Microsoft, Hewlett-Packard, GTE, NCR, and DEC have refused comment at this time.

Borland International, a leading vendor of object--oriented tools, including the popular Turbo Pascal and Borland C++, stated they had suspected for Windows was originally written in C++. Philippe Kahn said: "After two and a half years programming, and massive programmer burn-outs, we re-coded the whole thing in Turbo Pascal in three months. I think it's fair to say that Turbo Pascal saved our bacon". Another Borland spokesman said that they would continue to enhance their Pascal products and halt further efforts to develop C/C++.

Professor Wirth of the ETH Institute and father of the Pascal, Modula 2, and Oberon structured languages, cryptically said "P.T. Barnum was right." He had no further comments.

By decree, this article must be read on 2001-04-01
All names are Registered Trademarks of their respective companies. This article was found on the USENET - its author could not be determined.

The Open Source Lucky Dip

Con Zymaris conz@cyber.com.au

Welcome back.

As you likely know, what each issue of OSLD contains are just a few snippets of the programs that have caught my attention over the past month or two. Some are more sys-admin oriented, some are primarily for programmers, and some are just plain kewl. If you come across any open source projects or apps which fit the bill, send them in: auugn@auug.org.au

###

Microsoft Ports Wine To Windows

REDMOND, WA -- Microsoft announced today that, after a month of intense development, it had successfully ported Wine to the Windows 9x operating system. Microsoft(R)(tm) Wine(R)(tm) for(R)(tm) Windows(R)(tm) is a closed-source fork of the Wine project (an open source Windows emulator). It consists of a standalone .exe file that uses less than 2 kB of diskspace.

-- From a recent Humorix piece.

###

Dialog

Dialogm by developer Vincent Stemen, lets you to present a variety of questions or display messages using dialog boxes from a shell script (or any scripting language). These types of dialog boxes are implemented: yes/no box,menu box, input box, message box, text box, info box, guage box, checklist box, file-selection box, and radiolist box. Dialog is GPL. Get it from: http://www.AdvancedResearch.org/dialog/

wxPython

Here's a tool which is both useful and cool. wxPython is a GUI toolkit for the Python programming language. It allows Python programmers to create programs with a robust, highly functional graphical user interface, simply and easily. It is implemented as a Python extension module (native code) that wraps the popular wxWindows cross platform GUI library, which is written in C++.

wxPython is a cross-platform toolkit. This means that the same program will run on multiple platforms without modification. Currently supported platforms are Microsoft Windows, and most Unix or unix-like systems. It's open source, and available from:
http://wxpython.org/what.php

BlueJ

Here's another local project which has garnered a solid reputation. BlueJ is an interactive Java development environment. It provides a unique user interface that presents a graphical display of the application classes and their relationships, and it lets users interactively create objects of any class. Once objects have been created, users can interact with them directly. This interaction mechanism allows for much greater testing and experimentation than in conventional environments. BlueJ is suited for teaching and learning OO and Java. Download free (for non-commercial use) from http://bluej.monash.edu/

The Network Audio System (NAS)

You can think of the Network Audio System (NAS) is the audio equivalent of an X display server. It was developed by NCD for playing, recording, and manipulating audio data over a network. Like the X Window System, it uses the client/server model to separate applications from the specific drivers that control audio input and output devices.Sounds intriguing? Give it a try: http://radscan.com/nas.html

PHP-GTK: PHP language bindings for GTK+

So, you've learned some PHP, but want to do more than just code up some web-apps? Too often PHP is thought of as only an HTML-embedded Web scripting language. But it is also a very full-featured general purpose language that can be used for much more. One of the goals behind this project was to prove that PHP can be used to write client-side GUI applications.

GNU VCDImager/VCDRip

If you've ever wanted to make your own VideoCDs, here your chance. Written by Valerio RiedelGNU VCDImager is a program for making Video CD (and Super Video CD, a.k.a. SVCD) images out of MPEG movie files. The images it creates are ready to use with programs which understand BIN/CUE images, such as cdrdao. GNU VCDRip allows for reversing the process, ripping mpeg tracks from (Super) Video CDs. Get it free from here:
http://www.hvrlab.org/~hvr/vcdimager/

###

Yes! Yes! Yes! Thank You, Jim Allchin!

Humorix's stock (Nasdaq: FAUX) soared 50% in heavy trading today after word spread that Microsoft's OS chief, Jim Allchin, had claimed that open source would "stifle innovation" and had boasted, "We can build a better product than Linux."

The latest Microsoft FUD barrage is expected to produce a windfall for online humor publications. "You simply can't make this stuff up," said one industry observer. "We haven't seen this kind of humor gold mine since Jesse Berst or Microsoft Bob."

###

linux.conf.au: An outsider's viewpoint

Author: Greg Lehey <grog@lemis.com>

As part of my work, I spend a lot of time with Linux people, but I'm not a Linux person myself: I'm a member of the FreeBSD core team and the author of the Vinum volume manager (http://www.vinumvm.org/). As a result, I found it particularly interesting to visit the linux.conf.au in Sydney from 18 to 20 January 2001. The following is an extract of my diary entries on the subject. You can read more at http://www.lemis.com/~grog/diary.html.

Thursday, 18 January 2001

Into the first day of the conference. I'm beginning to wonder whether multi-track conferences aren't overdoing things; you end up wanting to go to competing presentations. One way or another, there's not enough time to sit around and discuss things with people. The first keynote was with Alan Cox, who described the new 2.4 kernel. Judging by the title of the talk ("World Domination: Classified Progress Report and Briefing"), I suspect that Alan hadn't been expecting it to have been released either.

Then to Richard Gooch's devfs talk. He seems to have cleverly avoided the bikesheds which have held up FreeBSD devfs for so long, but traded it for surprisingly long pathnames. Something to follow up about.

In the afternoon, Daniel Phillip's talk about Tux2, a rearrangement of ext2fs which guarantees consistency. An interesting concept, but like all of these approaches, it's a tradeoff between performance and reliability. I'm still not convinced that soft updates are the way to go, but I'm left feeling that they're a better approach than this one.

Juan Quintela talked about his VM system test programs. Very much a work in progress, but it's good to see that somebody is doing it.

Rik van Riel spoke about the Linux VM system. It's interesting to see how much has been borrowed from FreeBSD, but the topics about which he spoke (very lucidly) don't seem to be the same topics that cause the heated discussions on the FreeBSD-developers mailing list.

In the evening, first to a "networking" reception, then to dinner at the Red Hat Chinese restaurant, chosen because of the name, though the food was good. Rusty Russell tried to buy one of their teeshirts, and was finally successful.

Back to the hotel in pouring rain. How could Paulus [Paul Mackerras] have claimed that the weather here is like Adelaide's weather a day or two later? The Great Divide makes its presence felt.

Friday, 19 January 2001

Started off this morning with David Miller's talk about zero copy file transfer for Linux. It was particularly interesting in that it seemed to be an adaptation of David Greenman's sendfile concept to Linux, somewhat complicated by the fact that Linux doesn't have mbufs. The details were above my head in the sense that, despite Jay Schulist's tutorial at the AOSS, I still don't understand Linux networking.

Then to Stephane Eranian's talk on IA64, which I personally found the most rewarding talk to date. Unfortunately got called out in the middle, and didn't get back until the talk was nearly over. I rather like some of the ideas of the IA64.

Next was Hugh Blemings' talk about reverse engineering. He had told me before that it was his first ever presentation of the kind, but he did an excellent job, demonstrating the setup he used to decode the Nokia serial protocol. Excellent. Excellent. Did I mention that Hugh is my boss?

Later to Rusty Russell's talk about how kernel hackers get the girls. Another enthusiastic talk about how hacking is so much fun. Rusty's a very different person from Hugh, of course, but it was interesting how both talks conveyed the enthusiasm that comes out of Canberra.

In the evening to the conference dinner. A pleasant, nay merry time was had by all. Late to bed.

Saturday, 20 January 2001

Up a little later than usual today, and found to my surprise that just about everybody else had survived the night unscathed. To Tridge's [Andrew Tridgell's] keynote, about the Tivo hacking. I suppose it was deliberate that Tridge, who doesn't drink, was put on first talk after the conference dinner. Still, the auditorium was packed, and of course he produced yet another enthusiastic talk from Canberra about the joys of hacking.

Tried to quickly grab my mail after that, and by the time I looked up again I found that I had missed the next session. In to hear Alan Au talking about TLB sharing in IA64. I'm not convinced of the approach.

After that, Neil Brown talking about his improvements to RAID-5 in the Linux md driver. His figures looked very impressive, but they were designed to optimize the sequential I/O case, and I can't see any direct application to Vinum. For example, he went to a lot of trouble to grab entire bands when writing, so that he didn't need a read before write phase. To do that he needed small stripe sizes; the largest he even tried were 32 kB in size, and he concentrated on stripes of 4 or 8 kB. The throughput was impressive, up to 50 MB/s as lied^H^H^H^Hreported by bonnie, but under BSD they would incur a much higher read and write I/O load for the more interesting multi-process random I/O case. I've recently been told that Linux still writes a maximum of 4kB at a time, though I thought I recalled hearing that they were doing much larger writes, up to a megabyte at a time. Once that happens, this hack will be worthless.

I came away with a number of interesting ideas to think about for Vinum, but gradually they faded as I realised they relied on these small stripes. It's interesting, though, that they allow multiple concurrent access to a stripe which is being written, depending on the state (i.e. progress) of the transfer. I wonder if this makes any real difference in random access situations. It obviously will if you're coalescing data for full band writes.

Sunday, 21 January 2001

Up a little later this morning, finally, and had a leisurely breakfast discussing kernel debugging with Rusty and Tridge. I'm still baffled that Linux doesn't have any facility for dumping core after a crash. It does display a stack backtrace on the console, but it suffers from the same problems with X as BSD does.

They kicked us out of the motel at 10 am for some obscure reason, so down to Coogee with Tridge to pick up Alan and Telsa Cox and into town to show them Yet Another Big City, then on to Andrew Van Der Stock's place for a barbecue. Had a pleasant time and took a number of photos.

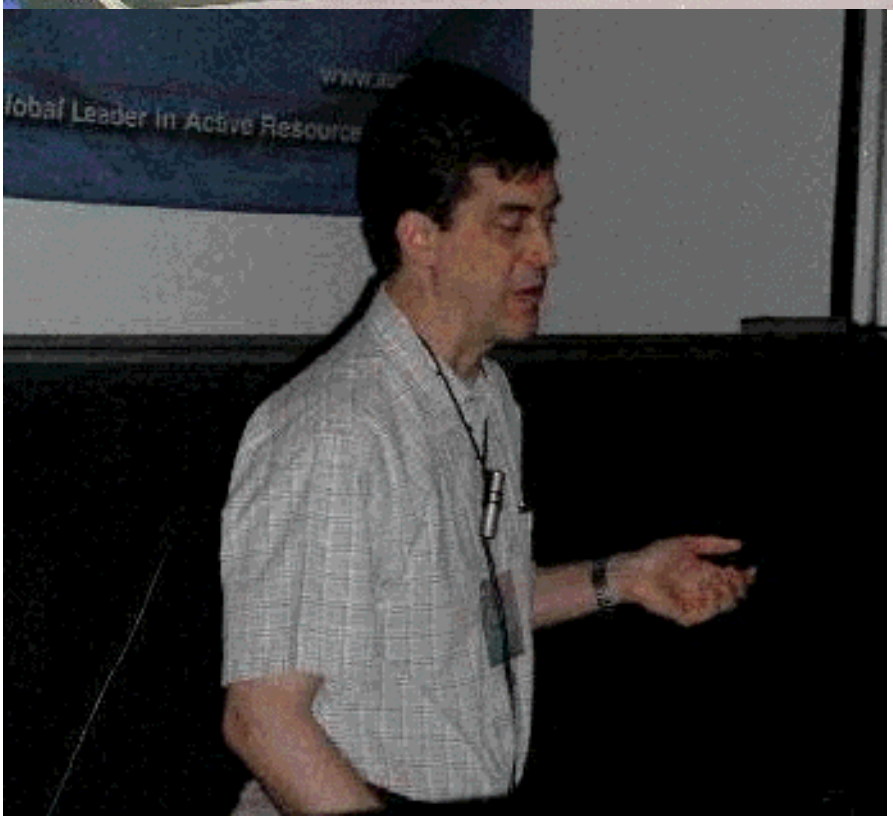
Photos from linux.conf.au

taken by Greg Lehey

Alan Au

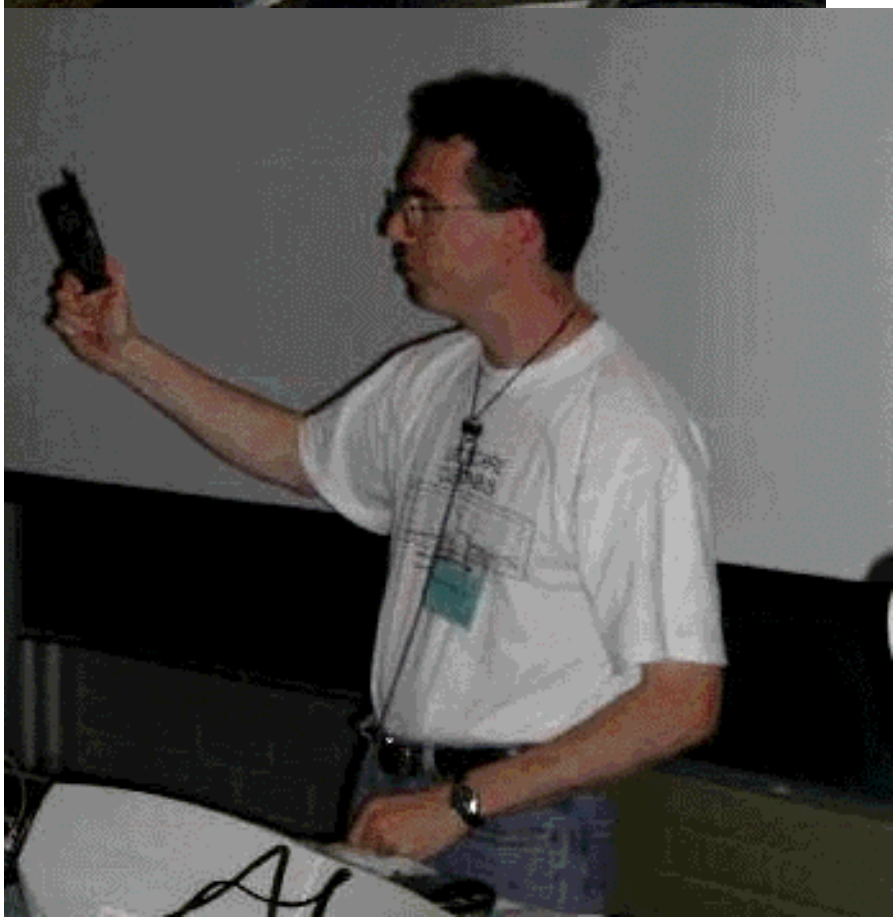
Alan Cox

Daniel Phillips



Bob Edwards

Hugh Blemings



David Miller

Neil Brown

Juan Quintela

Richard Sharpe and maddog

Jon ‘Maddog’ Hall

Alan Cox and
Paul Mackerras

Kirrily Robert and
Andrew van der Stock

Photos from BBQ after linux.conf.au

Andrew van der Stock's barbecue, 21 January 2001, taken by Greg Lehey

Richard Gooch and Andrew "Tridge" Tridgell

Daniel Philips on the left Telsa Cox on the right

From Power(point) to Magic(point) -- Presentations using your Linux box

Author: Trevor Warren <trevor@freeos.com>

At FreeOS.com, we have a simple rule: all work must be done on Open Source software as far as possible. The first victim of this rule was our CEO who would crib about not having a suitable application to make presentations on. Of course, we had Star Office, but there always was the feeling that there must be better ways of doing the job than using CPU hungry bloatware.

So we decided to look around and what better place than Freshmeat (www.freshmeat.net). A little sniffing around Freshmeat and we came up with some good tools to make simple and handsome presentations. We downloaded the following software:

Prestimel
PPresenter
Magic Point

Prestimel turned out to be a no-brainer as we could not install it. We were testing it on a SuSE 6.4 Installation running the 2.2.14 kernel. However, we just could not figure out why Prestimel would not compile on our box although all the dependencies were satisfied. We finally gave up and are therefore unable to review Prestimel. We however did have some previous experience with Prestimesl and would recommend that you give it a try (i.e. if you can compile it). Creating slides in Prestimel was a piece of cake as we remember. It was as simple as taking the sample XML file provided with the stock compilation of Prestimel and tweaking it to suit our needs. You then run Prestimel again with the appropriate command line switches parsing it a template for the background and color scheme according to your choice and lo..... Prestimel creates all the slides as separate HTML files interlinked to each other and created using the template mentioned by you during parsing of the XML file. All you then need is an HTML browser to view the slides.

That was as far as Prestimel goes. The next application to be chucked off our list was PPresenter. This was another pain in the .....? You know how cryptic some of these packages can be. We soon realized that getting started with PPresenter would require us to code our own slides in some scripting language that was almost equivalent to messing around with assembly code. Not that PPresenter was any different from the others in the way things worked, but we felt that it was too cryptic for the average user to get a hold on. And to expect a novice to start coding in some cryptic scripting language just to make a elegant presentation is asking quite a lot.

Than left us with Magic Point and fortunately for our CEO (or us!), we had a better experience.

MagicPoint is an X11 based presentation tool designed to make simple presentations easy and complicated presentations possible. Its presentation file (whose suffix is typically .mgp) is just text so you can create presentation files quickly with your favorite editor (e.g. vim, Emacs, pico, etc). MagicPoint is completely free and is offered under absolutely no warranty from it's developers. For more information, please refer to the Copy-write info bundled along with the package. Since MagicPoint extensively uses fonts in various sizes, the developers recommend installing good font rendering systems on the system. Please refer to README.fonts or README.fonts.jp for more details. However, in our experience, most stock installations of Linux from various leading distributions have no problem rendering the most common fonts required by MagicPoint.

Let us get started with down loading and compiling MagicPoint. The official home page of MagicPoint is http://www.mew.org/mgp/. Get the tarball from the homepage or look out for RPMs. The latest tarball available at the site is magicpoint-1.07a.tar.gz.

Untar the source as follows.

```
tar -xvzf  magicpoint-1.07a.tar.gz
```

Follow the steps below to compile MagicPoint:

```
./configure
xmkmf
make Makefiles
make
```

When you believe that everything is OK, install MagicPoint as follows after logging in as root:

```
make install
make install.man
```

In case, you were able to lay hands on the MagicPoint RPM, more the better. First login as root and type the following command

```
rpm -ivh mgp-*.rpm
```

This wraps up installing MagicPoint using either the sources or the RPMS?s.

Having installed MagicPoint we will test some of the sample presentations. In the directory in which you have down loaded the source, there is another directory called sample. Illustrated below is a snapshot of part of our MagicPoint directory structure.

```
-rw-r--r--   1 trevor   users       40668 Feb 11
09:47 print.o
drwxr-xr-x   2 trevor   users        1024 Feb 16
10:24 sample
-rw-r--r--   1 trevor   users       62701 Feb 11
09:45 scanner.c
-rw-r--r--   1 trevor   users        7094 Sep 10
1999 scanner.l
```

Inside the directory sample, you will find a lot of sample MagicPoint presentations. Run one of the presentations as follows:

```
mgp -g 800x600 sample.mgp
```

The above command will run the MagicPoint presentation for you in window of size 800x600.

Creating your own Presentations

We would suggest you first have a look at the various sample presentations, most of them are good enough to get you started. You can also create your own slides using the default ?sample.mgp? as reference. The basic funda of MagicPoint is all the slides are created in a single ASCII TEXT file stored with an extension of ?mgp?. Every slide is scripted using a very simple language as we are going to illustrate as follows. Open your favorite text editor and name your first presentation with an extension of mgp(*.mgp).

The first slide is illustrated as follows:

```
%include default.mgp          <--  1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%page               <--2
%nodefault        <--3
%fore yellow, size 5, font standard, back darkblue
<-- 4
%center, fore red, font thick         <--5
%font standard, rcutin                  <--6
WELCOME TO THE MAGIC POINT TUTORIAL             <-
-7
%center, size 4, fore white, lcutin
<--8
And also Welcome to Linux
<--9
%rcutin                                <--10
From FreeOS.com               <--11
%size 3, lcutin                     <--12
trevor@freeos.com               <--13
%size 4, fore yellow          <--14
!! Type SPC key to get to next page!!
<--15
```

Let’s look at a a line by line explanation of the presentation. The line numbers are the ones to the right of the script on each line. Neglect them when scripting your own presentation.

```
%include default.mgp    <--  1
```

One of the golden rules to remember while using MagicPoint is that every line begins with a "%? sign. Every line in the script line is of significance even the blank ones. The blank lines in the script file(*.mgp) are interpreted by MagicPoint as blank lines in the respective slide. So you know how to keep your distance.

The first line is a default line, which as in C/C++, is equivalent to a pre-processor directive and over here is made use of to include the basic template necessary to create your presentation. This default template will give your slide a lot of default values, but just as in C/C++ you can over ride a method, similarly even here you can over ride the default values by specifying your own values to the various parameters. We will be covering that shortly.

```
%page                   <--2
```

This marks the beginning of a new slide.

```
%nodefault             <--3
```

In short this means that this slide isn’t going to make use of the default parameters supplied by the default template(default.mgp). Therefore, you have some more scripting coming your way.

```
%fore yellow, size 5, font standard, back darkblue
<-- 4
```

Let us take these lines apart and understand it. The first part %fore yellow suggests that the slide should use yellow to draw the words that come up next. size 5 of course means use letters of height 5. font standard tells MagicPoint to make use of the standard font as included by the default.mgp file. back darkblue tells MagicPoint to draw the background as darkblue as the default is to draw the background as black. You could always change the colors to suit your own fancy. Just make sure you stick to the common range of colors and refrain from using any exotic names out here. The first line is used as a global default for the entire first slide.

Now let’s jump to the rendering of the first line.

```
%center, fore red, font thick       <--5
%rcutin
<--6
WELCOME TO THE MAGIC POINT TUTORIAL             <-
-7
%center, fore red, font thick       <--5
%
```

center of course is used to position the words in the center of the screen. Various combinations possible out here are as follows, center, right and left.

center
right   left

These positions are relative to the size of the frame you ask the MagicPoint to be drawn in. Let us illustrate with this example,

```
mgp -g 800x600 timepass.mgp   and
mgp -g 400x300 timepass.mgp.
```

Both of these lines would display the same slide but in a frame of reducing size. As a word of caution always test your slides for the positioning and the way they look as the frame size changes.

```
%center, fore red, font thick       <--5
```

The last part of the command is fore red, which mentions that MagicPoint should draw the words in RED. font thick specifies that the words should be drawn BOLD. So you see it’s as simple as it looks.

```
%rcutin                       <--6
```

You might have noticed the particular feature in PowerPoint, which allows your point to be shot in from the sides of the screen with a bullet effect. Similar is the rcutin and lcutin options out here. The rcutin option causes the following line to be shot in from the right of the screen while lcutin causes the line to be shot in from the left of the screen. Try out some of these exquisite effects for your presentations, it makes things look really impressive.

```
WELCOME TO THE MAGIC POINT TUTORIAL             <-
-7
```

All the commands that we mentioned in lines 5 and 6 were applicable to line number 7.

```
%center, size 4, fore white, lcutin
<--8
And also Welcome to Linux          <--9
```

Let us have a look at the following lines as mentioned above. There is nothing new out here in lines 8 and 9. All the points as explained earlier apply themselves similarly.

```
%rcutin  <--10
From FreeOS.com                             <--11
%size 3, lcutin
<--12
trevor@freeos.com                            <--13
%size 4, fore yellow                       <--14
!! Type SPC key to get to next page!!
<--15
```

Similar is the situation for the following lines as illustrated above. Also note that the functionality is mentioned by the last line, i.e line number 15.

This brings us to the end of the first slide. We now look at some more concepts.

```
%page <--16
%bgrad 0 0 16 0 0 blue darkblue
<-- 17
%size 5, font standard, fore yellow, lcutin
<--18
Moving around the presentation     <--19
%size 4, font standard, fore yellow, rcutin
<--20
# To view the next slide     -> press left mouse
button             <--21
%rcutin <--22
# To view the previous slide -> press right mouse
button         <--23
%rcutin <--24
# To quit                    -> press q key
<-- 25
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

The second slide as mentioned above gives the effect of drawing a color gradient on the background. The color gradient is between blue and darkblue. These are just some of the stunning effects possible with MagicPoint. To know more about some of the default templates available check out the MagicPoint homepage for good links.

Let's now go through line numbers 16-25 in this section.

```
%page <--16
%bgrad 0 0 16 0 0 blue darkblue
<-- 17
```

Line 16 denotes the starting of a new slide. Line 17 creates a color gradient between blue and dark blue. The grade variable takes 5 parameters, which can be described as follows.

```
%bgrad
:: width of generated image(0-100%)
0 means physical display size
:: height of generated image(0-100%)
0 means physical display size
:: number of colors, 0 means no reduction.
default 256 colors(8bit)
:: gradation direction (0-360 degree)
0 :Top to Bottom   90 :Left to Right
180:Bottom to Top 270 :Right to Left
default 0
negative value means non-liner gradation
:: zoom to max size
0 nozoom, 1 zoom
default 0
:: colors in the gradation image
```

For simplicity sake, just follow the values that we have given you and tweak them a bit to get satisfactory results.

```
%size 5, font standard, fore yellow, lcutin
<--18
Moving around the presentation     <--19
%size 4, font standard, fore yellow, rcutin
<--20
# To view the next slide     -> press left mouse
button            <--21
%rcutin <--22
# To view the previous slide -> press right mouse
button        <--23
%rcutin        <--24
# To quit                    -> press q key    <--
25
```

the rest of the statements are quite simple enough and it's functionality is the same as illustrated earlier.
Let's have a look at the next slide.

```
%page                              <--26
%back orange                 <--27
%size 4, font standard, fore darkblue, lcutin
<--28
Font sizes and zooming effects<--29
%size 3, font standard, fore darkblue, rcutin
<--30
Open source ROCKZ ......... :-)
<--31
%CENTER                        <-- 32
%SIZE 10,FORE orange, lcutin
<-- 33
Hello World                      <--34
%SIZE 7,FORE gray, lcutin <--35
Hello World                      <--36
%SIZE 5,FORE blue, lcutin <--37
Hello World                      <--38
%SIZE 4,FORE yellow, lcutin
<--39
Hello World                      <--40
%SIZE 3,FORE green, lcutin
<--41
Hello World                      <--42
%SIZE 2,FORE red, lcutin   <--43
Hello World                      <--44
%SIZE 1,FORE pink, lcutin <--45
Hello World                      <--46
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Lines 33-46 display the words Hello World in reducing font sizes and varying colors. Also we have made it a point to use the lcutin bullet effect for elegance and style.
Now we take a look at the next slide.

```
%page    <--47
%fore red, font thick          <--48
Visualize inline images!           <--49
%center <--50
%image dad.jpg                    <--51
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

The above lines insert an image, dad.jpg into the slide and notice how closely the image is center justified.

Now a look at the next slide.

```
%page         <--52
%size 4, lcutin      <--53
Pause            <--54
%size 4, lcutin     <--55
Type SPC key (twice) to proceed.
<--56
%center, fore red
<--57
Happy hacking!  <--58
%pause, fore blue
<--59
Happy hacking!  <--60
%pause, fore green
 <--61
Happy hacking!  <--62
```

The important feature that we want to highlight out here is as follows,

```
%center, fore red                          <--57
Happy hacking!                               <--58
%pause, fore blue                          <--59
Happy hacking!                               <--60
%pause, fore green
<--61
Happy hacking!                               <--62
```

If you notice carefully, lines 57-58 are of the normal type. However, after displaying lines 57-58 the presentation will pause due to the use of the pause keyword in line number 59. To continue the presentation you can use the SPACEBAR key. This feature is important if you don't want all the points barging onto the screen in one go and you would want to decide when the next point should appear on screen.

The next slide is quite interesting. Using this particular feature, you could embed output of live system commands into your presentation.

```
%PAGE <--63
Grab command output into foils        <--64
ls -l /boot                                    <--65
%size 3, prefix "     "                           <-
-66
%filter ls -l /boot  "                            <-
-67
%endfilter      <--68
echo this is test | rev               <--69
%filter rev <--70
this is test  <--71
%endfilter      <--72
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%filter ls -l /boot  "              <--67
%endfilter      <--68
```

Take a closer look at the lines 67-68, these lines incorporate a feature called FILTERS into the slide. Using this particular feature, you can pipe the output of a live system command into your presentation. The above line number 67 pipes the output of the ls -la /boot command into your slide. This com<-mand is equivalent to displaying the following data on screen.

```
trevor@freeos.com:~ > ls -al /boot
total 1062
drwxr-xr-x   3 root     root         1024 Sep  7
06:51 .
drwxr-xr-x  21 root     root          387 Jan  7
23:55 ..
-rw-r--r--   1 root     root       254408 Mar 25
2000 System.map-2.2.14
-rw-r--r--   1 root     root          512 Sep  7
06:51 boot.0300
-rw-r--r--   1 root     root         4568 Mar 25
2000 boot.b
-rw-r--r--   1 root     root          612 Mar 25
2000 chain.b
drwxr-xr-x   2 root     root        12288 Sep  7
06:25 lost+found
-rw-------   1 root     root        11776 Sep  7
06:51 map
-rw-r--r--   1 root     root          620 Mar 25
2000 os2_d.b
-rw-r--r--   1 root     root       792796 Mar 25
2000 vmlinuz
```

Last but not the least we will demonstrate how you can embed an X application into your slide. This powerful feature allows you to display a movie or an mpeg video during your presentation or maybe even run some other application as a demo during the slide.

```
%page                <--69
Wanna See an X APP (multimedia!)
<--70
%system xeyes -geometry %50x20+25+60"  <--71
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

This slide uses the system variable to startup a X application during this particular slide. You could startup any application using the same command line arguments you would have used, had you been working at the CLI (Command Line Interface).
For a start you could use the script given at the bottom of the article. Run it using the following command:

```
mgp -g 800x600 sample.mgp
```

The above command will run the MagicPoint presentation for you in window of size 800x600.

Of course, you could always vary the window size according to your screen resolution. If everything works fine, you know you have taken another step forward towards making your life a little simpler.

This is as far as we go for now. We sincerely hope this tutorial has helped you discover some of the finer aspects of making some elegant presentations on Linux using MagicPoint.

Till next time CIAO !

This article is re-printed with permission. The original can be found at
http://trevor.freeos.com/articles/3648/

What Are You Gonna Do? 'Make' Me?

by Ray Hernandez <hernan43@msu.edu>

Introduction

That's what I used to tell my brother, when he wanted me to clean my half the room. And as lazy as I used to be, I still don't like to exert myself more than I have to. That's the beauty of the make utility, it's like my own personal programming Punjab(sans turban.)

Like Baking a Harley

Make has always been described to me as if one were baking a cake. You have a prescribed list of ingredients, each one with its own place in the recipe. The directions are followed and out pops a cake. Make is great for cake.

But it really shines when you're baking things like a motorcycle. Let me explain. Your basic cake has a handful of ingredients, at best, but a motorcycle contains hundreds, if not, thousands of parts. Make can execute compiler/linker commands, shell commands, other makefiles, and even has some nifty scripting features of its own. All of this makes it easy to use make to build large programming projects, not just cakes.

Cake for Breakfast

Makefiles are made up of a series of structures that define targets, dependencies, and the commands that build said targets. The most basic ingredient for a makefile is structured like:

```
target: dependency1 dependency2 ... de-
pendencyN
     command(s)
```

Depending on the version of make that you are using, each line containing commands may have to be preceded by a tab. I always use a tab to avoid any possible gang confrontations. Let's take a look at what our makefile might look like if we were compiling a really nasty cake.

```
#Our Cakefile(yuk yuk)
#Our cake
cake: eggs.o flour.o milk.o frosting.o
     gcc -o cake eggs.o flour.o milk.o
frosting.o
#Ingredient One
eggs.o: eggs.c eggs.h
     gcc -c eggs.c
#Ingredient Two
flour.o: flour.c flour.h
     gcc -c flour.c
#Ingredient Three
milk.o: milk.c milk.h
     gcc -c milk.c
#Ingredient Four
frosting.o: frosting.c frosting.h
     gcc -c frosting.c
```

Assuming the makefile was written correctly, and the source code had no errors, we would type 'make' and watch the magic happen. The make utility would first compile our four ingredients into object code, and then link them all together in the final step. By first compiling them into object code, it allows the make utility to save us some time.
Let's say we had an error in eggs.c. If all our makefile did was to run one huge compile command, we would have to re-compile and re-link the entire project. By having it divided like in the above makefile, we would only have to re-compile eggs.c into eggs.o and run the linking command, taking much less time in the process.

So Are There Keebler Elves or What?

When the make command is run, it searches for one of two files named either makefile or Makefile. When it locates one of these files, it then searches that file for the first target, in our case cake. It examines cake's dependencies and then proceeds to build cake if any of its dependencies are newer than cake itself. If any of the dependencies do not exist it attempts to build them in the same manner it tries to build cake.

You Call That Fine Cuisine?

The cakefile was a very crude and simplistic implementation of make. The problem with the cakefile way of doing things is that when our makefiles get larger and more complex, changes are going to be a royal pain. For instance, if we had a much larger makefile that shared the same technique as the above cake makefile and we wanted to use a different compiler command, we would have to change every instance of it line by line. I can hear my brother already.

Who Wants Cake? I Want Macro-ni Instead

One of the greatest features of makefiles, and one that I exploit to the fullest, are macros. In the cakefile, we talked about how much of a pain it was to have to change multiple lines of our makefile by hand if we needed to make one measly change. Using macros, we can define common values in one spot and only have to change them there. Here is the basic structure for a macro:

```
MACRONAME=MACROVALUE
It can be accessed using:
${MACRONAME} or $(MACRONAME)
Here is the re-written cakefile using a
couple macros:
#Our NEW Cakefile
CC=gcc
CFLAGS=-c
OFLAGS=-o
#Our cake
cake: eggs.o flour.o milk.o frosting.o
     ${CC} ${OFLAGS} cake eggs.o flour.o
milk.o frosting.o
#Ingredient One
eggs.o: eggs.c eggs.h
     ${CC} ${CFLAGS} eggs.c
#Ingredient Two
flour.o: flour.c flour.h
     ${CC} ${CFLAGS} flour.c
#Ingredient Three
milk.o: milk.c milk.h
     ${CC} ${CFLAGS} milk.c
#Ingredient Four
```

```
frosting.o: frosting.c frosting.h
     ${CC} ${CFLAGS} frosting.c
```

A little better. Now if we wanted to change the compiler all we would have to do is change the CC macro at the top of the file. You can macro just about anything in a makefile, but don't get too carried away. There is such a thing as over-macro.

Man that Hit the Target!

Some popular targets:

- clean - normally used to delete core dumps and object code files

EXAMPLE: (The '@' sign tells make not to print to console)

```
clean:
@rm -irf *.o core
```

- very-clean - like clean but you hose pretty much everything the compiler commands create

- all - makes all potential executables

EXAMPLE: (note the 'all' dependencies are the targets we want built)

```
all: cake ice-cream presents
```

install - copies the executable to a bin directory or some other install point

Buurrrrpp!

This article is far from the end all be all on make. There is way too much functionality in make for me to even try and tackle it in one sitting. However, this article does provide a good start and I encourage you to visit the GNU Make Homepage. It offers a complete look at all of the features and functionality that make provides. Soon you'll be baking Harley's like the pros do. :)

Links

GNU Make Homepage
http://www.gnu.org/software/make/make.html

Makefile Conventions
http://www.gnu.org/prep/standards.html#SEC48

Little Orphan Annie
http://www.liss.olm.net/loahp/

This article is re-printed with permission. The original can be found at:
http://www.linuxlookup.com/html/articles/make.html

PHP on Speed!

Author: Dan Barber <dan@mojolin.com>

This is a down and dirty guide to compiling Apache with PHP, MySQL and SSL support, as well as how to install SGI's 10xpatch for apache_1.3 series and the

ZendOptimizer.

I've done this numerous times now, so have it down to quite a science (which these things should be, anyway). I came up with this framework primarily so when I needed to upgrade, add a patch, or build another system, I could just follow this step by step list and not have to expend any real brain power (this should be very comfortable for former MS Windows users).
The steps I have listed must be (with few exceptions) executed in the order listed, or the install will not work.

A disclaimer:

- I guarantee nothing. Follow these steps at your own risk.

- I make no claims on the speed or security of the resulting executables. This may turn out to be a very slow rootkit.

Some notes:

- I am assuming the reader knows the basic commands, ie tar, cp, mv, etc.

- I use Slackware.

- All packages must be installed from source because, either:
  - other packages use the source when compiling, or
  - because you must add extra options when compiling

- I use Apache and the apache-ssl patch, not mod_ssl
  - Ensure that you have the correct apache-ssl patch for the version of Apache your are using!!

- All instructions begin with the source untarred into it's own directory.
  - shown version numbers are only for example.
  - you will want to use the latest stable version of each package

- If you make an unsuccessful attempt to patch Apache, you may have to start over with clean Apache source before you can move on.
- Pay close attention to which directory each command is executed in!

Installing mysql [MySQL]

Okay, here we go. tar zxvf mysql-3.23.28--gamma.tar.gz in /usr/local/src. Change directory into /usr/local/src/mysql-3.23.28-gamma. Run the following commands. These are pretty much straight out of MySQLs manual. Change the prefix to wherever you want the resulting package to be installed. Again, you need the MySQL source code because PHP needs it for when it is compiled.

```
myhost:/usr/local/src/mysql-3.23.28-
gamma# ./configure --
prefix=/usr/local/mysql
myhost:/usr/local/src/mysql-3.23.28-
gamma# make
myhost:/usr/local/src/mysql-3.23.28-
gamma# make install
myhost:/usr/local/src/mysql-3.23.28-
gamma# scripts/mysql_install_db
myhost:/usr/local/src/mysql-3.23.28-
gamma# cd /usr/local/mysql/bin
myhost:/usr/local/src/mysql-3.23.28-
gamma# ./safe_mysqld &
myhost:/usr/local/src/mysql-3.23.28-
gamma# ./mysqladmin -u root password
'new-password'
```

Build/install openssl [OpenSSL]

Very simple... again according to package instructions.

```
myhost:/usr/local/src# tar zxvf openssl-
0.9.6.tar.gz
myhost:/usr/local/src/openssl-0.9.6#
./config --prefix=/usr/local/openssl
myhost:/usr/local/src/openssl-0.9.6#
make ; make test ; make install
```

Initial apache config with ssl patch and SGI 10x patch [10xpatch] [Apache] [Apache-SSL]

Make sure to have the latest/matching Apache and apache-ssl releases!!!!!

Here's where things get interesting: In order to compile PHP, you must do an initial ./config of Apache. But, I also wanted to get that SGI 10xpatch working. I found out that I needed to do both the SSL patch and the 10x patch in this initial config, otherwise, Apache would fail to compile later on. The 10x patch also says that it must be applied after all other patches, so here is the resulting order. Again, pay close attention to which directory you are in as you execute these commands; both patches should be applied from inside the Apache source directory.

```
myhost:/usr/local/src# tar zxvf
apache_1.3.14.tar.gz
myhost:/usr/local/src# cp
apache_1.3.14+ssl_1.42.tar.gz
apache_1.3.14
myhost:/usr/local/src/apache_1.3.14# tar
zxvf apache_1.3.14+ssl_1.42.tar.gz
myhost:/usr/local/src/apache_1.3.14#
./Fixpatch
```

(will ask two questions; I answer no to first and yes to second)

```
myhost:/usr/local/src/apache_1.3.14#
patch -p1 < 10xpatch-1.3.14-0
myhost:/usr/local/src/apache_1.3.14#
./configure --prefix=/var/lib/apache
```

Build/install php and zendoptimizer [PHP] [ZendOptimizer]

If you made it through that last part, you're probably 70% there. Here comes the next 20%. The main task here is to decide which additional options you want to add into PHP when you compile. You can see the options I chose, below. Keep in mind that I'm not a real guru on these options; you'll want to do some research for yourself. One other option that you may choose to add is the --enable-trans-sid which, if you are using PHP's built-in cookie functionality (I chose not to), will automatically keep track of a user's session. Mojolin and Mojosco do not use cookies at all. For installing the ZendOptimizer, 1) make sure you get the ZO for the version of PHP you are using; it only works with versions 4.03 and 4.04. You will need to add a couple lines into php.ini which point to the location of your ZO.so file.

```
myhost:/usr/local/src# tar zxvf php--
4.0.3pl1.tar.gz
myhost:/usr/local/src/php-4.0.4pl1#
./configure --with--
mysql=/usr/local/mysql --with-xml --
with-apache=../apache_1.3.14 --enable--
track-vars --enable-register-globals
myhost:/usr/local/src/php-4.0.4pl1#
make; make install
myhost:/usr/local/src/php-4.0.4pl1# cp
php.ini-dist /usr/local/lib/php.ini
myhost:/usr/local/src/php-4.0.4pl1/libs#
tar zxvf ZendOptimizer4.04
myhost:/usr/local/src/php-4.0.4pl1/libs#
cp ZendOptimizer4.04/ZendOptimizer.so ..
myhost:/usr/local/src/php-4.0.4pl1/libs#
(add two lines to php.ini)
```

Build/install Apache

After all that, this part should be simple. Change directory back into the Apache source, configure, make and make install.

```
myhost:/usr/local/src/apache_1.3.14#./c-
onfigure --activate-
module=src/modules/php4/libphp4.a --
enable-module=php4 --
prefix=/var/lib/apache
myhost:/usr/local/src/apache_1.3.14#
make
myhost:/usr/local/src/apache_1.3.14#
make install
```

Summary listing of links to source home pages

MySQL	http://www.mysql.com/
OpenSSL	http://www.openssl.org/
Apache	http://www.apache.org/
Apache-SSL	http://oss.sgi.com/projects/apache
10xpatch	http://oss.sgi.com/projects/apache
PHP	http://www.php.net/

ZendOptimizer
http://www.zend.com/zend/optimizer.php

That should do it. The results were noticeably faster on my machine, though I did not do any

benchmarking. I'll leave that to you. If Apache compiles without error, you'll want to move along to configuring your httpd.conf file and load up a php page. A useful resource is the phpinfo() script which shows you all kinds of info about your environment.

This article is re-printed with permission. The original can be found at:
http://mojolin.com/articles/php_install.php

Dcument Processing

Gene Wilburn <njo@interlog.com>

Contrary to popular belief, word processors go about their business in the wrong way. They're interlopers in the land of Unix--violating four time-proven Unix principles:

1. You should be able to create all source files (including word processing documents) with your preferred text editor (vi and emacs being the prevailing standards).
2. All files should be portable and accessible to the Unix toolkit; that is, they should exist as ASCII text. You should have no trouble running your source files through grep, sed, awk, wc or whatever you choose.
4. Structure is more important than appearance.
5. Look and feel (appearance) processing should be handled by programs external to your source files.

This stands the world of WYSIWYG word processing on its head. You cannot work directly on Word or WordPerfect files due to their binary nature. You must use the program that created them to manipulate the contents--a dangerous strategy for long-term use. You eventually get releases of a product that no longer willingly load older binaries. Cross-platform support among word processors is an ongoing issue.

Another serious problem with word processors is that they are weak in several key areas: long document creation (theses, books and long reports), mathematical equations, and indexing. While support for these features is often present, it's not particularly sophisticated or robust. Word processors are also weak in the area of typesetting--the output is passable, but not up to serious typesetting standards.

So what's the pro-Unix answer to this? You guessed it: text processing, better known these days as document processing. Document processing works in much the same way that the Web does: i.e., on the Web, HTML documents are text files containing markup code. You can use any simple editor to create a web page. A web browser turns the source file into an attractive screen display. Depending on your skill as a web author, the final HTML display can be plain or sophisticated, but the tools needed to create them are dead simple.

The original document processing tools for Unix were troff and nroff--programs that took the contents of marked-up source files and formatted them for printers and typesetting machines. The troff program still exists in Linux in the form of groff, but it is now used mainly for creating man pages.

The primary document formatting engine for Linux is TeX (pronounced TECH--the X being the Greek symbol chi). TeX was developed by Donald Knuth, the godfather of algorithms, to create beautiful documents, especially documents that contained mathematical equations. He succeeded admirably and TeX has proven popular for general document creation, even for non-mathematical documents.
To make TeX easier to use, a set of coherent, simplified TeX macros called LaTeX were developed by Leslie Lamport. LaTeX has been highly popular for over a decade and may be one of the most mature, thoroughly debugged programs available on any computing platform. TeX and LaTeX are available for Linux, Unix, VMS, Windows, Macintosh, and OS/2. The version of TeX most often used in Linux is called TeTeX, a modernized version of TeX.

So, let's say you don't find this document processing concept totally retro and you're willing to try it out. If you've already created HTML documents by hand, you're well on your way. You just need to change the nature of the markup tags.

Most LaTeX instructions are preceded by a backslash ("\") and have fairly intuitive names. As with an HTML page, LaTeX documents have begin and end tags, a header, and a body, as well as additional structural elements that go beyond HTML. Here's an example of a simple LaTeX letter source file which we'll call myletter.tex:

```
\documentstyle{letter}
\address{586 Linux Drive \\ Port Debian,
ON \\ L5G 9X9}
\signature{Ima Texhead, Jr.}
\begin{document}
\begin{letter}{Ima Texhead, Sr. \\
               486 BSD Way \\
               Berkeley, CA 95587}

\opening{Dear Dad,}

Hope you're proud to see I'm using \La-
TeX, just like you, Dad. Now that I've
arrived, I could use some cash for a new
Linux system. Sorry I didn't use email
but I love the look of \LaTeX\ output.

\closing{Thanks,}
\end{letter}
\end{document}
```

Notice that the document begins with a "\documentsyle{letter}" tag. This is similar to using HTML CSS (cascading style sheets). It uses a predefined format for the letter (which you can override should you wish). The address and signature structural elements are located near the top of the document. The double backslashes ("\\") tell TeX to insert a line break between these elements (like the
 tag in HTML). The backslashes preceding LaTeX ("\LaTeX") form a macro that typesets the word LaTeX in a special way.

There are two nested "\begin" statements followed by their corresponding "\end" statements. The first is "\begin{document}" then "\begin{letter}" (the type of document). Because this document is a letter, it supports the standard structural elements of a letter, such as "\opening" and "\closing". Paragraphing is utterly simple: simply insert a blank line between paragraphs. We now have a source document that is ASCII text. How does this get turned into a printer output file? The next step is to run the LaTeX source file through TeX which we do at the command line by typing:

```
$ latex myletter.tex
```

Assuming you have LaTeX on your Linux system (it is usually installed by default), you will witness some screen activity. When it's over you will see a few files with the same base name but different extensions. The main one you're looking for is myletter.dvi.

DVI stands for "device independent". By default LaTeX always creates a device-independent file. It can be used for viewing or printing to a printer driver.

Viewing? Yup, and unlike the so-called WYSIWYG word processors, viewing a dvi file is WYSIWYRG ("what you see is what you Really get"). If you're in an X Window session, type the following to view your letter:

$ xdvi myletter.dvi &

[mages are shown at the end of this article --ed

Xdvi works somewhat like a combination web browser/PDF viewer in showing how the output looks. You can flip from page to page in a long document and you can magnify the page to see small details. Printing the file requires a dvi-aware printer driver. Here's what I type for an HP Laser Printer:

$ dvi2lj myletter.dvi ; lp myletter.lj

Another approach is to turn the LaTeX dvi file into a Postscript file (using dvips) and executing gv (Ghostview) to view the contents. LaTeX is Postscript friendly; in fact most serious typesetting work with LaTeX is done with Postscript fonts and files.

Philosophy

The philosophy behind LaTeX is that it's better to concentrate on structure than on looks. Don't worry, LaTeX will make the output look highly professional and you can tweak the looks considerably once you gain experience, but the emphasis is on getting the structure right.

This is particularly important for long documents, with chapters, section, and subsection headings, footnotes, bibliographic references, captions and illustrations. LaTeX lets you can break a report or book into component chapters or sections and tie them all together with a master document. This keeps chapter sizes manageable. The indexing capabilities of LaTeX are particularly strong and there is a well-developed accompanying bibliographic file format called BibTeX that allows you to cite bibliographic materials in a scholarly fashion. LaTeX has no peers when it comes to displaying mathematical formulae--it's simply the best.

Inserting figures and illustrations into LaTeX is not particularly difficult. It's akin to using IMG tags in HTML. You create the illustrations in a separate package then use LaTeX statements to place them in the text. One of the payoffs of using LaTeX is that you can use your single source document for multiple outputs.

Let's say you've written a book in LaTeX. You can typeset the book by creating Postscript files to be sent to a printing house or you can print the book on a laser printer. By using a program called pdflatex you can turn your book into a PDF document. And with latex2html you can turn your book into an entire linked website, including linked index, footnotes, and table of contents elements.

LyX

Let's say you agree with all this in principle but it sounds hard and your time is short. You need "LaTeX with training wheels" in the form of LyX, an open--source document processor that has the look and feel of a GUI word processor but which outputs LaTeX files. If you know how to use any modern word processor, you can start using LyX immediately.

LyX has a lovely interface and gets you into LaTeX without having to learn a single LaTeX tag. It will also take care of your previewing and printing needs (a click of a mouse on a pull-down menu) without having to type anything at the command line. The product has an excellent built-in tutorial and user guide and, in short, removes the barrier to doing document pro cessing the right way. LyX can be easily adapted for general office use. Lyx is available as source code or in binary format at www.lyx.org.

Resources

If you would like to explore LaTeX and LyX beyond this simple introduction, there are several resources available to you. It's essential to become acquainted with CTAN (Comprehensive TeX Archive Network) at www.ctan.org. If you want to do anything special in LaTeX, check CTAN first. As a very mature product, LaTeX libraries and special macros abound in plentiful numbers and variations. In addition to all the normal things you might want to use for books, articles and reports, you can find LaTeX macros and stylesheets (.sty files) for creating musical scores, booklets, pamphlets, barcodes, chess diagrams, and even for typesetting crossword puzzles.

There is an excellent Internet newsgroup devoted to LaTeX discussions: comp.text.tex.

A key reference to LaTeX is, not surprisingly, LaTeX: A Document Preparation System, by Leslie Lamport, published by Addison Wesley Longman, 1994 (ISBN 0201529831 Cdn$55.50). Note that Lamport's book, although a very good introduction, is somewhat out of

date. There are two excellent supplementary books by Michel Goossens, et al., both published by Addison Wesley Longman: The LaTeX Companion, 1994 (ISBN 0201541998 Cdn$56.95) and The LaTeX Graphics Companion, 1997 (ISBN 0201854694 Cdn$59.95). A search on "LaTeX" at the online sites for Chapters, Indigo and Amazon will list more than a dozen other books on the subject.

So, Contrarians of the world, feel relief. You have at your fingertips one of the most sophisticated and refined document preparation systems on the planet---and it's free.

This article is re-printed with permission. The original can be found at:
http://www.northernjourney.com/opensource/newbies/newb020.html

Fig. 1 Xdvi Preview:

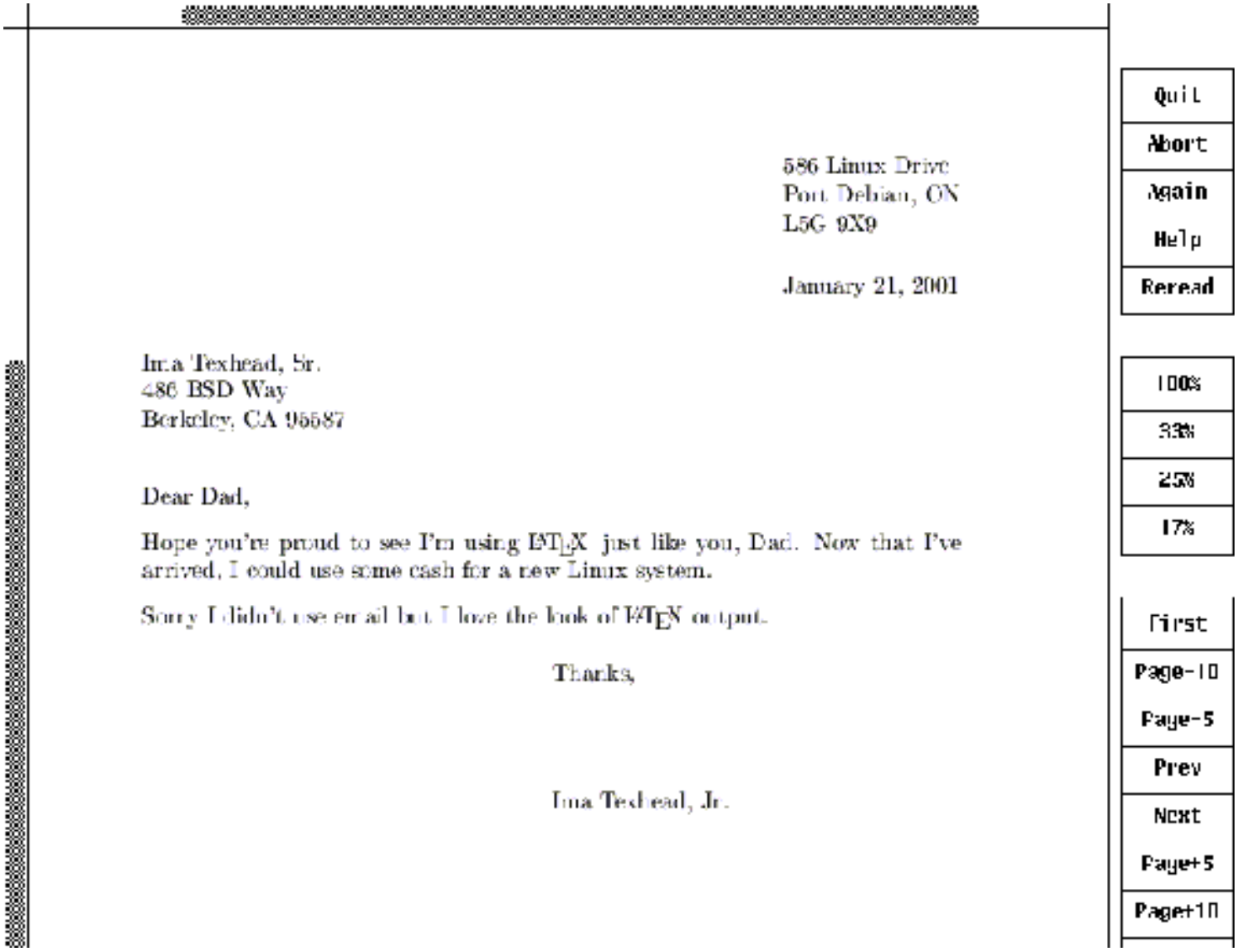
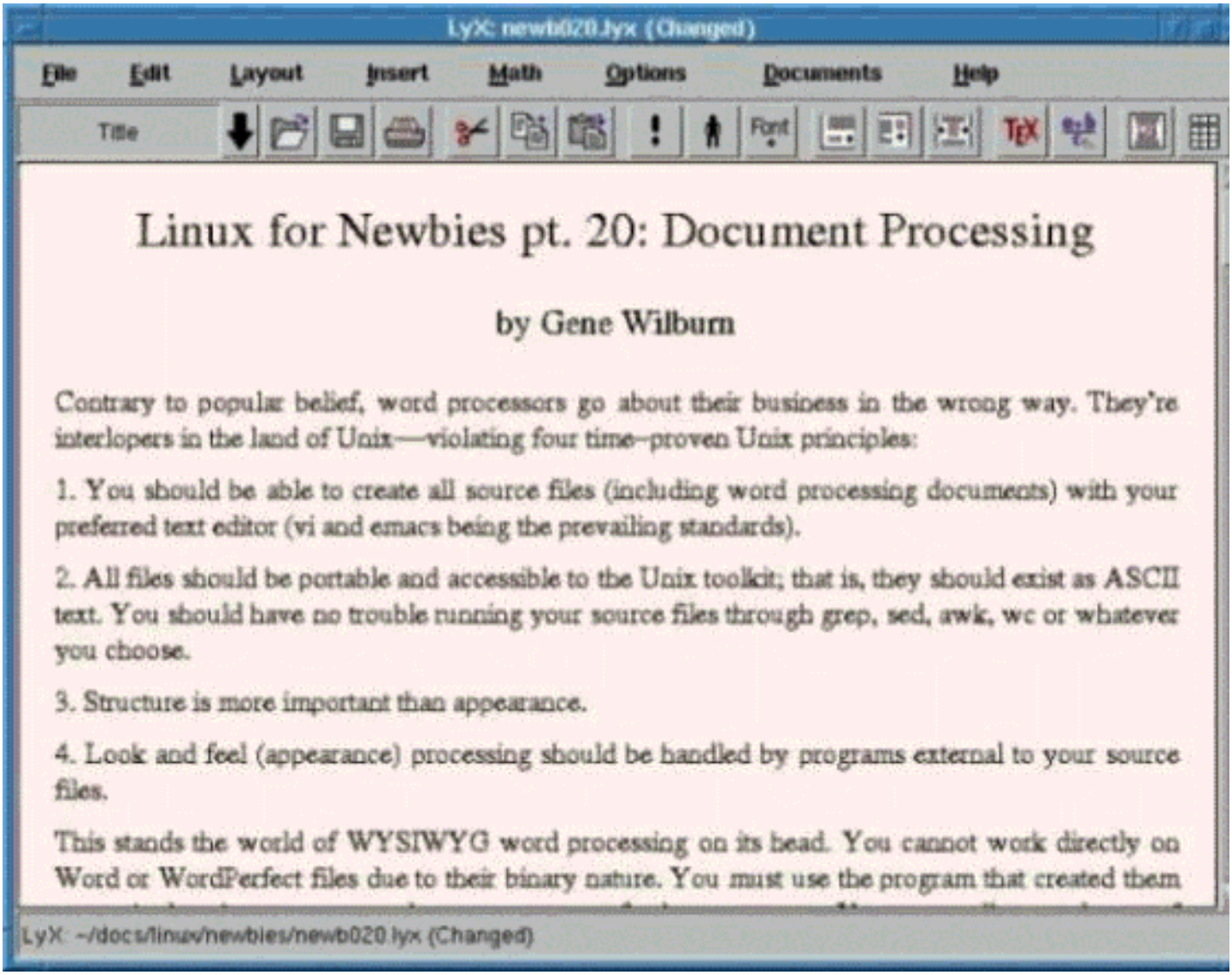


Fig. 2 LyX Screenshot:



Network monitoring, access control & booby traps using TCP Wrappers: Parts 1 and 2

By Trevor Warren <trevor@freeos.com>

TCP Wrappers is one of the most common methods of access control on your Unix box. A wrapper program 'wraps' around existing daemons and interfaces between clients and the server. Good access control and logging are strong points. In this first part, we introduce you to the concept behind TCP Wrappers.

Will opens the door to success.
- Louis Pasteur

What according to you would be the best way to fortify your machine from the anarchy of the Internet? Let's look at a few options.

You could put in a well configured firewall. Though this is quite a complex procedure, it's worth going through the pain. And there will be pain. Putting down your organizational policies and framing rulesets that keep the bad guys out and lets the good guys in, requires a lot of thinking on your part. Experience has taught us to verify our rulesets again and again, because even one slip during the framing of the rulesets can bring the house down. At the end of the day, you'll be the only punching bag around.

OR

You could be too lazy to have any security measures in place. You justify this by saying that among the millions of machines out there, you're not going to be hit.

OR

You could be totally paranoid about security concerns on the Internet. Securing yourself from the Internet is a simple affair of disconnecting your machine from the network. But then, you wouldn't be reading this article.

OR

You could put security measures in place that GRANT or DENY access to various services on your machines depending on the privileges that you have setup using TCP Wrappers. TCP Wrappers by itself isn't a complete solution as far as securing you machine is concerned. But it does fit into the overall scheme of framing a security policy for your enterprise.

In this first part of the series, we will be introducing to you the whole concept surrounding the working of TCP Wrappers. We will leave out the implementation part for later. For now, we will help you get a foot hold on the use and importance of TCP Wrappers on a Linux/UNIX system. One more point that comes to mind, and which warrants clarification is the uniformity of concept that we are dealing with in relevance to the various UNIX operating systems . Of course, the procedure of implementation of the TCP Wrapper differ across various Unix systems, but we can assure you that if can find yourself GCC/G++ for your Unix machine, there's no stopping you from implementing this latest version of TCP Wrappers on your machine.

Getting down to business .....

Are any of you Linuxers familiar with Eindhoven University of Technology, Netherlands? If you are, you would be fairly familiar with the product that originated from the labs of the "Mathematics and Computing Science Department, Eindhoven University of Technology". By now, you should have figured out our object of concern. TCP Wrappers was born under an interesting set of circumstances. It won't be relevant for us to bore you with those circumstances but it sure warrants reading. It's a typical administrators saga trying to track down a hardened cracker whose ultimate goal was to obtain a remote shell to run "rm -rf /". Though the cracker was never brought to justice, the occurrence of such episodes throws light on the serious vulnerabilities on Unix architectures.

Before getting to know why we really need an application called TCP Wrappers, lets look at the protocols that our Internet/ Intranet is based upon. Most of the Intranets we come across and probably yours too is based upon the Ethernet standard offering 10/100 MBps data transfer duplex / half duplex on our Local Area Networks (LAN's). More recently, we have seen the emergence of Gigabit Ethernet in use for our corporate backbones. Ethernet as we know of is situated at the lower level in reference to the OSI model.

OSI MODEL

Application
Presentation
Session
Transport
Network
Link
Physical

Ethernet is structured in such a way that it functions at the bottom of the OSI model. The Internet as we commonly know it is structured on the TCP protocol or Transmission Control Protocol. TCP/UDP works at the upper layers of the OSI model. We won't get into too much of the nitty gritty out here, but to be specific, TCP/UDP will work at the transport layer of the OSI model.

Most of the applications being used on the Internet today are based on the Client - Server model. This client server model is our prime focus of our discussion from now on. The author of TCP Wrappers has written the application in such a way that it intervenes in the functioning of the required TCP application, which is based on the client-server model, and for which access control has to be administered. Depending on the access control lists specified for the particular protocol, TCP Wrappers lets the client initiate a connection to

the server or just drops the connection. Either way, it logs all attempts to access the particular service. TCP Wrappers is written in such a fashion, for the simple reason that the author - a brilliant guy that he is - never wanted to re-engineer any of the hundreds of client-server applications just to make sure that they were compatible with TCP Wrappers. In fact, he did things exactly the other way around - made TCP Wrappers compatible with all other client-server applications.

This tool has been successfully used for shielding off systems and for detection of cracker activity. It has no impact on legal computer users, and does not require any change to existing systems software or configuration files. The tool has been installed world-wide on numerous UNIX systems without any source code change. Such is the beauty of TCP Wrappers.

Almost every application of the TCP/IP protocols is based on a client-server model. For example, when someone uses the telnet command to connect to a host, a telnet server process is started on the target host. The server process connects the user to a login process. A few examples are shown in table 1.

client	server	application
telnet	telnetd	remote login
ftp	ftpd	file transfer
finger	fingerd	show users
systat	systatd	show users

Table 1. Examples of TCP/IP client-server pairs and their applications.

The usual approach is to run one daemon process that waits for all kinds of incoming network connections. Whenever a connection is established this daemon (usually called inetd on our Linux boxes) runs the appropriate server program and goes back to sleep, waiting for other connections. See the example as illustrated below.

client	server	application
telnet (foo1.bar)	telnetd (foo2.bar)	remote login

We are on a client Linux box called foo1.bar and want to connect to a remote Linux box called foo2.bar which resides on a remote network. We then use the telnet client application from my box i.e foo1.bar to connect to the remote telnet server box foo2.bar. Have a look at the graphical illustrations as given below.

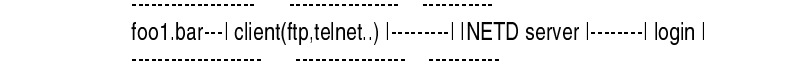


Figure 1. The inetd daemon process listens on the ftp, telnet etc. network ports and waits for incoming connections. The figure shows that a user has connected to the ftp/telnet port.

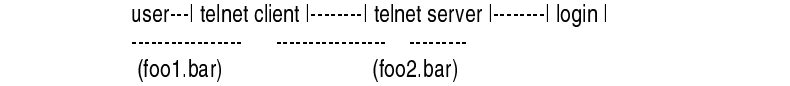


Figure 2. The inetd process has started a telnet server process that connects the user to a login process. Meanwhile, inetd waits for other incoming connections. This illustrates an unprotected machine.

Fortunately, the author of TCP wrapper came up with a simple solution that did not require any change to existing software, and that turned out to work on all UNIX systems that were ever tried it on. The trick was to make a swap. Move the vendor-provided network server programs to another place, and install a trivial program in the original place of the network server programs. Whenever a connection was made, the trivial program would just record the name of the remote host, and then run the original network server program.

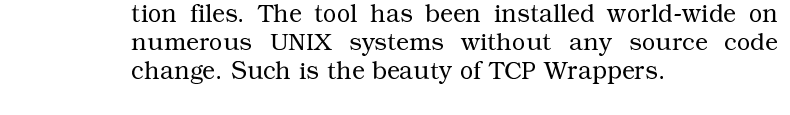


Figure 3. The original telnet server program has been moved to some other place, and the tcp wrapper has taken its place. The wrapper logs the name of the remotehost to a file. This illustrates a protected machine.

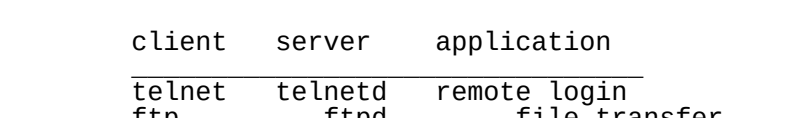


Figure 4. The tcp wrapper program has started the real telnet server and no longer participates. The user cannot notice any difference.

Lets look at the logs capable of being generated by our TCP wrapper application.

```
May 22 14:43:29 tuegate: systatd: connect from monk.rutgers.edu
May 22 15:08:30 tuegate: systatd: connect from monk.rutgers.edu
May 22 15:09:19 tuewse:  fingerd: connect from monk.rutgers.edu
May 22 15:14:27 tuegate: telnetd: connect from cumbic.bmb.columbia.edu
May 22 15:23:06 tuegate: systatd: connect from cumbic.bmb.columbia.edu
May 22 15:23:56 tuewse:  fingerd: connect from cumbic.bmb.columbia.edu
```

Some of the first cracker connections observed with the tcp wrapper program by the author. Each connection is recorded with: time stamp, the name of the local host, the name of the requested service (actually, the network server process name), and the name of the remote host.

Automatic reverse fingers had proven useful in the authors fight against the cracker, so he decided to integrate the "ad hoc" reverse finger tool with TCP Wrappers. To this end, the access control language was extended so that arbitrary shell commands could be specified.

```
/etc/hosts.allow:
in.tftpd: LOCAL, .foo.bar
/etc/hosts.deny:
in.tftpd: ALL: /usr/ucb/finger -l @%h 2>&1 | /usr/ucb/mail wswietse
```

This is an example of a booby trap on the tftp service. The entry in the first access control file says that tftp

connections from hosts within its own domain are allowed. The entry in the second file causes the TCP Wrapper to perform a reverse finger in all other cases. The "%h" sequence is replaced by the actual remote host name. The result is sent to the administrator by email.

Our discussion till now gives only a limited illustration of the use of booby traps. Booby traps can be much more useful when installed on firewall systems, whose primary purpose is to separate an organizational network from the rest of the world. A typical firewall system provides only a limited collection of network services to the outer world. For example, telnet and smtp. By placing booby traps on the remaining network ports one can implement an effective early--warning system.

Conclusions

The TCP Wrapper is a simple but effective tool for monitoring and controlling network activity. Probably it has been installed in almost every part of the world, and that it's use is picking up almost every day.

Some of the documentation and illustrations that we have made use of for this article accompanied the documentation that comes along with the TCP rapper package. In case you are interested in knowing more about the TCP Wrapper package, look at the following sites.

ftp.uu.net:/comp.sources.misc/volumexx/log_tcp
ftp://cert.org:/pub/tools/tcp_wrappers/tcp_wrapper-s.*
ftp.win.tue.nl:/pub/security/log_tcp.shar.Z.

In our next article on this series of TCP Wrappers we will continue our discussion on the implementation of TCP Wrappers and we will help you create a minimal security policy using these TCP Wrappers.

The secret of success is working
with things the way they are, not
with the way you wish they were
or they ought to be.
- Anon

In the second part of our series on TCP Wrappers, we look at its various features, implementation and configuration.

Last week, we had a look at the concept of TCP Wrappers from the theoretical perspective. As we have already mentioned, TCP Wrappers isn't meant to fulfill the security measures you would want for an enterprise network. But it surely does fall into the greater scheme of rule sets that would make up a comprehensive strategy to protect an enterprise network. The author of TCP Wrappers mentions this stating, that TCP

Wrappers could be made use of along with a firewall box on your corporate gateway with minimum services running. While building a firewall, we suggest, that you pipe all the firewall logging off the gateway. Although complicated to set up, this is the best way to secure your logs incase your firewall machine is compromised.

Features

With the TCP Wrapper package you can monitor and filter incoming requests for the SYSTAT, FINGER, FTP, TELNET, RLOGIN, RSH, EXEC, TFTP, TALK, and other network services. It supports both, 4.3BSD-style sockets and System V.4-style TLI. Count yourself lucky if you don't know what that means.

The package provides tiny daemon wrapper programs that can be installed without any change to the existing software or to existing configuration files. The wrappers report the name of the client host and of the requested service. Neither do they exchange information with the client or server applications, nor impose overhead on the actual conversation between the client and server applications.

Optional features include:
- Access control to restrict what systems can connect to what network daemons
- Client user name lookups with the RFC 931 etc. protocol
- Additional protection against hosts that pretend to have someone else s host name or address

The programs are portable. 'Build procedures are provided for many common (and not so common) environments and guidelines are a great help incase your environment is not among them.

Requirements:
- Network daemons should be spawned by a super server such as the inetd 4.3BSD-style socket programming interface and/or System V.4-style TLI programming interface
- Availability of a syslog(3) library and of a syslogd(8) daemon.

The wrappers should run without modification on any system that satisfies these requirements. Workarounds have been implemented for several common bugs in systems software.

TCP Wrapper vulnerabilities

The TCP Wrapper program, as we all know, is intelligent enough to perform a reverse finger on the client from where the connection originates and logs all the data to disk, if asked to do so. But, for instance, if the source IP address were spoofed, TCP Wrapper, being totally ignorant about such malpractices, wouldn't suspect any foul play. The wrapper programs rely on source address information obtained from network packets. This information is provided by the client host. It is not 100 percent reliable, although the wrappers do their best to expose forgeries.

Recap

Let us take a quick look at the functioning of TCP Wrappers.

Almost every application of the TCP/IP protocols is based on a client-server model. For example, when a user invokes the telnet command to connect to one of

your systems, a telnet server process is executed on the target host. The telnet server process connects the user to a login process. A few examples of client and server programs are shown in the table below:

client	server	application
telnet	telnetd	remote login
ftp	ftpd	file transfer
finger	fingerd	show users

The wrapper programs rely on a simple, but powerful mechanism. Instead of directly running the desired server program, the inetd is tricked into running a small wrapper program. The wrapper logs the client host name or address and performs some additional checks. If there are no glitches, the wrapper executes the desired server program and goes away.

The wrapper programs neither interact with the client user or the client process nor with the server application.

This has two major advantages:

1. The wrappers are application-independent. Therefore, the same program can protect different kinds of network services.
2. Lack of interaction also means that the wrappers are invisible from outside (at least for authorized users).

Another important property is that the wrapper programs are active only when the initial contact between client and server is established. Once a wrapper has done its work, there is no overhead on the client--server conversation.

But like everything else, this mechanism too has its drawbacks. A major one being that since the wrappers go away after the initial contact between client and server processes, they are of little use with network daemons that service more than one client. The wrappers only see the first client attempt to contact such a server. The NFS mount daemon is a typical example of a daemon that services requests from multiple clients.

Using TCP Wrappers

There are two ways to use the wrapper programs:
The easy way, where you move network daemons to some other directory and fill the resulting holes with copies of the wrapper programs. This approach involves no changes to system configuration files, and hence the risk of breaking things is minimal.

and

The advanced way: where you leave the network daemons alone and modify the inetd configuration file.

For example, an entry such as:

```
tftp  dgram  udp  wait  root  /usr/etc/tcpd  in.tftpd -s /tftpboot
```

When a tftp request arrives, inetd will run the wrapper program (tcpd) with a process name 'in.tftpd'. This is the name that the wrapper will use when logging the request and scanning the optional access control tables. 'in.tftpd' is also the name of the server program that the wrapper will attempt to run when all is well. Any arguments, ('-s /tftpboot' in this particular example) are transparently passed on to the server program.

Logging information route

The wrapper programs send their logging information to the syslog daemon (syslogd). The disposition of the wrapper logs is determined by the syslog configuration file usually /etc/syslog.conf. Messages are written to files, to the console, or are forwarded to a @loghost. Some syslogd versions can even forward messages down a |pipeline.

Older syslog implementations only support priority levels ranging from 9 (debug-level messages) to 0 (alerts). All logging information of the specified priority level (or more urgent) is written to the same destination. In the syslog.conf file, priority levels are specified in numerical form. For example,
8/usr/spool/mqueue/syslog causes all messages with priority 8 (informational messages), and anything that is more urgent, to be appended to the /usr/spool/mqueue/syslog file.

Newer syslog implementations support message classes in addition to priority levels. Examples of message classes include mail, daemon, auth and news. In the syslog.conf file, priority levels are specified with symbolic names: debug, info, notice, ..., emerg.
For example,

```
mail.debug              /var/log/syslog
```

causes all messages of class mail with priority debug (or more urgent) to be appended to the

/var/log/syslog file.

By default, the wrapper logs go to the same place as the transaction logs of the sendmail daemon. The disposition can be changed by editing the Makefile and/or the syslog.conf file. Send a 'kill -HUP' to the syslogd after changing its configuration file. Remember that syslogd, just like sendmail, insists on one or more TABs between the left-hand and right-hand side expressions in its configuration file.

Configuring TCP Wrappers

The first step towards configuring Wrappers on your systems is to make sure your INETD daemon is properly configured to accept and forward connections to the respective SERVER applications through which, you plan to offer various services. Let's have a look at a sample INETD configuration file.

```
/etc/inetd.conf
# The inetd will re-read this file whenever it gets that signal.
#
#
ftp    stream  tcp     nowait  root    /usr/sbin/tcpd  wu.ftpd -a
```

This is the inetd daemon configuration file wherein you will specify the server to be monitored. The above entry is for the FTP server, which causes the INETD

server to accept connections and pass on the connection to the wrapper program /usr/sbin/tcpd. TCP Wrapper, then depending on the ACL's set from the files /etc/hosts.allow and /etc/hosts.deny, ALLOW or DENY connections to the respective server daemons.

Now, a look at some sample ACL's using our /etc/hosts.deny and /etc/hosts.allow files.

```
/etc/hosts.allow:
in.tftpd: LOCAL, .foo.bar
ypserv: 127.0.0.0/255.0.0.0 10.0.0.0/255.0.0.0
```

You could always check out these entries on your machine by editing the configuration files as mentioned above. The first entry in the hosts.allow file is as follows:
in.tftpd: LOCAL, .foo.bar

This very clearly states that all connections to the TFTP server daemon should be allowed if the connection originates from the local machine or the foo.bar domain.

```
ypserv: 127.0.0.0/255.0.0.0 10.0.0.0/255.0.0.0
```

This very clearly states that all connections to the YPSERV server daemon should be allowed if the connection originates from the local machine (127.0.0.0/255.0.0.0) or from the IP 10.0.0.0/255.0.0.0.

```
/etc/hosts.deny:
in.rshd: ALL: /usr/ucb/finger -l @%h 2>&1 | /usr/ucb/mail foobar
in.telnetd: 202.54.11.23 192.168.1.
in.rshd: ALL: /usr/ucb/finger -l @%h 2>&1 | /usr/ucb/mail foobar
```

The first entry in the above configuration tells TCP Wrappers that all connections to the RSH daemon should be dropped and a reverse finger should be sent to client, logging all the information obtained.

```
in.telnetd: 202.54.11.23 192.168.1.
```

This entry simply denies all connection attempts from the IP address 202.54.11.23 and all machines from the subnet 192.168.1.*.

Conclusion

We hope this up-to-date information on the usage and working of TCP Wrappers will serve you well against the crackers and script kiddies who target unprotected hosts. Just before we wind up, a reminder that securing and fortifying your machines is an on going process and any lax can cost you more than what you bargained for.

This article is re-printed with permission. The originals can be found at:
http://www.freeos.com/articles/3729/
http://www.freeos.com/articles/3768/

Unified Logons between Windows NT and UNIX using Winbind

Authors: Tim Potter <tpot@samba.org>, Andrew Tridgell <tridge@samba.org>

Abstract

Integration of UNIX and Microsoft Windows NT through a unified logon has been considered a "holy grail" in heterogeneous computing environments for a long time. We present winbind, a component of the Samba suite of programs as a solution to the uni.ed logon problem. Winbind uses a UNIX implementation of Microsoft RPC calls, Pluggable Authentication Modules, and the Name Service Switch to allow Windows NT domain users to appear and operate as UNIX users on a UNIX machine. This paper describes the winbind system, explaining the functionality it provides, how it is configured and how it works internally.

1. Introduction

It is well known that UNIX and Microsoft Windows NT have different models for representing user and group information and use different technologies for implementing them. This fact has made it diffcult to integrate the two systems in a satisfactory manner.

One common solution in use today has been to create identically named user accounts on both the UNIX and Windows systems and use the Samba suite of programs to provide file and print services between the two. This solution is far from perfect however, as adding and deleting users on both sets of machines becomes a chore and two sets of passwords are required both of which which can lead to synchronization problems between the UNIX and Windows systems and confusion for users.

We divide the unified logon problem for UNIX machines into three smaller problems:

- Obtaining Windows NT user and group information
- Authenticating Windows NT users
- Password changing for Windows NT users

Ideally, a prospective solution to the unified logon problem would satisfy all the above components without duplication of information on the UNIX machines and without creating additional tasks for the system administrator when maintaining users and groups on either system. The winbind system provides a simple and elegant solution to all three components of the unified logon problem.

What Winbind provides

Winbind unifies UNIX and NT account management by allowing a UNIX box to become a full member of a NT domain. Once this is done the UNIX box will see

NT users and groups as if they were native UNIX users and groups, allowing the NT domain to be used in much the same manner that NIS+ is used within UNIX-only environments.

The end result is that whenever any program on the UNIX machine asks the operating system to lookup a user or group name the query will be resolved by asking the NT domain controller for the specified domain to do the lookup. Because Winbind hooks into the operating system at a low level (via the NSS name resolution modules in the C library) this redirection to the NT domain controller is completely transparent.

Users on the UNIX machine can then use NT user and group names as they would use \native" UNIX names. They can chown files so that they are owned by NT domain users or even login to the UNIX machine and run a UNIX X-Window session as a domain user.

The only obvious indication that Winbind is being used is that user and group names take the form DOMAIN n user and DOMAIN n group. This is necessary as it allows Winbind to determine that redirection to a domain controller is wanted for a particular lookup and which trusted domain is being referenced. Additionally, Winbind provides a authentication service that hooks into the Pluggable Authentication Modules (PAM) system to provide authentication via a NT domain to any PAM enabled applications. This capability solves the problem of synchronizing passwords between systems as all passwords are stored in a single location (on the domain controller).

2.1 Target uses

Winbind is targeted at organizations that have an existing NT based domain infrastructure into which they wish to put UNIX workstations or servers. Winbind will allow these organizations to deploy UNIX workstations without having to maintain a separate account infrastructure. This greatly simplifies the administrative overhead of deploying UNIX workstations into a NT based organization.

Another interesting way in which we expect Winbind to be used is as a central part of UNIX based appliances. Appliances that provide file and print services to Microsoft based networks will be able to use Winbind to provide seamless integration of the appliance into the domain.

3 How Winbind Works

The winbind system is designed around a client/server architecture. A long-running winbind daemon listens on a UNIX domain socket waiting for requests to arrive. These requests are generated by the NSS and PAM clients and processed sequentially.

The technologies used to implement winbind are described in detail below.

3.1 Microsoft Remote Procedure Calls

Over the last two years, e.orts have been underway by various Samba Team members to decode various aspects of the Microsoft Remote Procedure Call (MSRPC) system. This system is used for most network related operations between Windows machines including remote management, user authentication and NT print spooling. Although initially this work was done to aid the implementation of Primary Domain Controller (PDC) functionality in Samba, it has also yielded a body of code which can be used for other purposes.

Winbind uses various MSRPC calls to enumerate domain users and groups and to obtain detailed information about individual users or groups. Other MSRPC calls can be used to authenticate NT domain users and to change user passwords. By directly querying a Windows PDC for user and group information, winbind maps the NT account information onto UNIX user and group names.

3.2 Name Service Switch

The Name Service Switch, or NSS, is a feature that is present in many UNIX operating systems. It allows system information such as hostnames, mail aliases and user information to be resolved from di.erent sources. For example a standalone UNIX workstation may resolve system information from a series of at files stored on the local filesystem. A networked workstation may first attempt to resolve system information from local files, then consult a NIS database for user information or a DNS server for hostname information.

The NSS application programming interface allows winbind to present itself as a source of system information when resolving UNIX usernames and groups. Winbind uses this interface, and information obtained from a Windows NT server using MSRPC calls to provide a new source of account enumeration. Using standard UNIX library calls, one can enumerate the users and groups on a UNIX machine running winbind and see all users and groups in a NT domain plus any trusted domain as though they were local users and groups.

The primary control file for NSS is /etc/nsswitch.conf. When a UNIX application makes a request to do a lookup the C library looks in /etc/nsswitch.conf for a line which matches the service type being requested, for example the "passwd" service type is used when user or group names are looked up. This config line specifies which implementations of that service should be tried and in what order. If the passwd config line is:

```
passwd: files example
```

then the C library will first load a module called /lib/libnss_files.so followed by the module /lib/libnss_example.so.

The C library will dynamically load each of these modules in turn and call resolver functions within the modules to try to resolve the request. Once the request is resolved the C library returns the result to the application

This NSS interface provides a very easy way for Winbind to hook into the operating system. All that needs

to be done is to put libnss_winbind.so in /lib/ then add "winbind" into /etc/nsswitch.conf at the appropriate place. The C library will then call Winbind to resolve user and group names.

3.3 Pluggable Authentication Modules

Pluggable Authentication Modules, also known as PAM, is a system for abstracting authentication and authorization technologies. With a PAM module it is possible to specify different authentication methods for different system applications without having to recompile these applications. PAM is also useful for implementing a particular policy for authorization. For example a system administrator may only allow console logins from users stored in the local password file but only allow users resolved from a NIS database to log in over the network.

Winbind uses the authentication management and password management PAM interface to integrate Windows NT users into a UNIX system. This allows Windows NT users to log in to a UNIX machine and be authenticated against a suitable Primary Domain Controller. These users can also change their passwords and have this change take effect directly on the Primary Domain Controller.

PAM is con.gured by providing control files in the directory /etc/pam.d/ for each of the services that require authentication. When a authentication request is made by an application the PAM code in the C library looks up this control file to determine what modules to load to do the authentication check and in what order. This interface makes adding a new authentication service for Winbind very easy, all that needs to be done is that the pam winbind.so module is copied to /lib/security/ and the pam control files for relevant services are updated to allow authentication via winbind. See the PAM documentation for more details.

3.4 User and Group ID Allocation

When a user or group is created under Windows NT is it allocated a numerical relative identi.er (RID). This is slightly different to UNIX which has a range of numbers which are used to identify users, and the same range in which to identify groups. It is winbind's job to convert RIDs to UNIX id numbers and vice versa.

When winbind is configured it is given part of the UNIX user id space and a part of the UNIX group id space in which to store Windows NT users and groups. If a Windows NT user is resolved for the .rst time, it is allocated the next UNIX id from the range. The same process applies for Windows NT groups. Over time, winbind will have mapped all Windows NT users and groups to UNIX user ids and group ids.

The results of this mapping are stored persistently in a ID mapping database (held in a tdb database). This ensures that RIDs are mapped to UNIX IDs in a consistent way.

3.5 Result Caching

A active system can generate a lot of user and group name lookups. To reduce the network cost of these lookups winbind uses a caching scheme based on the SAM sequence number supplied by NT domain controllers.

User or group information returned by a PDC is cached by winbind along with a sequence number also returned by the PDC. This sequence number is incremented by Windows NT whenever any user or group information is modified. If a cached entry has expired, the sequence number is requested from the PDC and compared against the sequence number of the cached entry. If the sequence numbers do not match, then the cached information is discarded and up to date information is requested directly from the PDC.

4 Installation and Configuration

The easiest way to install winbind is by using the packages provided in the pub/samba/appliance/ directory on your nearest Samba mirror. These packages provide snapshots of the Samba source code and binaries already setup to provide the full functionality of winbind. This setup is a little more complex than a normal Samba build as winbind needs a small amount of functionality from a development code branch called SAMBA TNG.

Once you have installed the packages you should read the winbindd man page which will provide you with con.guration information and give you sample con.guration .les. You may also wish to update the main Samba daemons (smbd and nmbd) with a more recent development release, such as the recently announced Samba 2.2 alpha release.

5 Limitations

Winbind has a number of limitations in its current released version which we hope to overcome in future releases:

- Winbind is currently only available for the Linux operating system, although ports to other operating systems are certainly possible. For such ports to be feasible, we require the C library of the target operating system to support the Name Service Switch and Pluggable Authentication Modules systems. This is becoming more common as NSS and PAM gain support among UNIX vendors.
- The mappings of Windows NT RIDs to UNIX ids is not made algorithmically and depends on the order in which unmapped users or groups are seen by winbind. It may be difficult to recover the mappings of rid to UNIX id mapping if the file containing this information is corrupted or destroyed.
- Currently the winbind PAM module does not take into account possible workstation and logon time restrictions that may be been set for Windows NT users.
- Building winbind from source is currently quite tedious as it requires combining source code from two Samba branches. Work is underway to solve this by providing all the necessary functionality in the main Samba code branch.

6 Conclusion

The winbind system, through the use of the Name Service Switch, Pluggable Authentication Modules, and appropriate Microsoft RPC calls have allowed us to provide seamless integration of Microsoft Windows NT domain users on a UNIX system. The result is a great reduction in the administrative cost of running a mixed UNIX and NT network.

References

1. For more details see the nsswitch.conf(5) man page
2. On most Linux systems you will .nd detailed PAM documentation in /usr/doc/pam*/
3. On UNIX systems with a 32 bit uid/gid space it would be simpler to just use a linear algorithmic mapping. With the release of the Linux 2.4 kernel Linux systems will be ready for 32 bit UIDs. At that time we expect to release an update to winbind to use a linear mapping and dispense with the mapping database
4. We are working on removing this requirement for a future release
5. The appliance releases are based on earlier development versions of Samba 2.2

This article is re-printed with permission. The original can be found at:
http://open-projects.linuxcare.com/research-papers/winbind-08162000.html

Summary of Minutes from AUUG Exec Meeting

By: Liz Carroll <busmgr@auug.org.au>

4 November 2000, 10:00am - 4:00pm, Melbourne, VIC

ATTENDEES:			
Elizabeth Carroll	EC	Luigi Cantoni	LC
Michael Paddon	MP	Malcolm Caldwell	MC
Peter Gray	PG	Alan Cowie	AC
Sarah Bolderoff	SB	Greg Lehey	GL
APOLOGIES:			
David Purdue	DP	David Newall	DN
GUEST:			
Con Zymaris	CZ		
NOTETAKER:			
Elizabeth Carroll	EC		

President's Report

A copy of the President's report was posted prior to the meeting to the Exec Committee.

The good news is that the President believes the strategies that we have set out are bearing fruit. Bringing membership processing in house has certainly paid off, with a much lower churn rate in renewals.

The two symposia set for November are well in hand, with arrangements for the Security Symposium finalised, and arrangements for AOSS II nearly completed.

The bad news is that AUUG is still suffering from the exec being overly busy. Hopefully, some of this can be corrected at this meeting.

Arrangements for AUUG'2001 have started, and we have appointed Lucy and Peter Chubb* (note: this has since changed with David Newall taking the position), as the programme committee chair. We have not appointed a conference chair, and it has been proposed that as for the last two years the AUUG management committee manages the conference directly.

Invitations need to go out to prospective speakers

Con Zymaris was thanked for volunteering as the new AUUGN editor.

Secretary's Report

Current membership statistics (as at 2000/10/27).

The data in our current membership database now looks extremely clean. Well done to Liz for getting the numbers up to date.

As a consequence, there has been an apparent drop in membership, however, this is simply a truer snapshot than before. Obviously, AUUG needs to work out a better way of attracting members. It seems that areas of decline are generally related to the amount of local activities, however the ACT's numbers are surprising given the recent winter conference.

Numbers from the last report are in brackets. Percentages may add up to more than 100 due to rounding errors.

	* 561	(664) members:	
Individual Member	368	(432)	66%
Corporate Member	155	(187)	28%
Student Member	16	(21)	3%
Freebies	17	(17)	3%
Subscription	2	(2)	<1%
Life Member	2	(2)	<1%
Corporate Sponsor	1	(0)	<1%
NSW	177	(194)	32%
VIC	123	(151)	22%
QLD	87	(91)	16%
ACT	83	(116)	15%
WA	36	(44)	6%
SA	17	(25)	4%
OTHER	17	(19)	3%
TAS	15	(17)	3%
NT	6	(7)	1%

Lower membership numbers than last time, due to a clean up of the database - clearer figures. Some members have taken memberships for more than one year, could be an idea for some in the future.

Company returns - ongoing investigation as to which returns need to be submitted. Expected completion date: December 2000

No correspondence of note was received during the last quarter.

Treasurer's Report

The Treasurer presented the AUUG Budget including quarterly figures.

Discussion ensued on the question of GST on Tutorial speakers both National and International.

Business Manager's Report

A copy of the Business Manager's report was posted prior to the meeting to the Exec Committee.

MEMBERSHIPS

Membership processing is up-to-date, with June renewals still coming in, having sent reminders to those not yet received. The December renewals will be sent out in mid November.

A spin-off from the database now being accurate, is the fact that the correct number of AUUGN's are being printed, thus reducing the printing and postage costs, as well as wastage.

ACCOUNTS

AUUG accounts are all up-to-date. Procedures have been put into place which now means AUUG's financial situation can be seen at a glance. AUUG's Telstra bill should be reduced over the next quarter, through cancelling some unnecessary services.

AUUG2K - There is currently $8,529 outstanding from AUUG2K registrations, made up from two government departments, both of which are being followed up.

AUUG 2001- Currently following up the venue. Further discussion on AUUG 2001 should see some issues resolved.

SECURITY SYMPOSIUM -3 NOVEMBER, MELBOURNE - At the time of writing this report, we currently have 39 people who will be attending the Security Symposium. There are 3 sponsors, being eSec Ltd, Check Point and Trend Micro. AUUG is in profit for this event; at this point in time.

AOSS2- 25 NOVEMBER, MELBOURNE - There are 2 sponsors for this event, being Red Hat and VA Linux, with support from ISOC and SAGE.

REGISTRATIONS FOR EVENTS - A standard document has been set up for all registrations received for any events, and monies received and outstanding can be seen at a glance. Registrations, are confirmed within 48 hours of receipt and serve as a Tax Receipt.

AUUGN - The new editor for AUUGN is Con Zymaris. Craig Macbride has volunteered to be in charge of the book reviews.

SYSTEMS MAGAZINE - Columns have been regular for Systems Magazine.

SUMMARY - AUUG is now running smoothly, much of that is due to the procedures that are now in place. The key areas being membership processing, accounts and event management.

Some problems are still arising from Exec members' busy schedules, however, that said, there has been an improvement over the last quarter in getting things done.

Minutes of Previous Meeting

The Minutes were accepted.

Action Items

Action Items were addressed and current status discussed.

AUUGN

Vote of thanks to Con Zymaris to be the new editor.

Sponsorships

From recent experience, it has appeared that sponsorship for AUUG has proved most successful in relation to events.

Chapters

Sarah Bolderoff wants to start up the SA chapter. The Exec has said they will help out - she needs to come to the board with a proposal.

Unix Traps and Tricks: Internet Printing Revisited

Author: Graham Jenkins, grahjenk@au1.ibm.com

You are doing some work on your home PC, connected to the Internet through your favourite ISP – and you decide you want to print a Word document on the high speed colour printer at your office. That printer is connected to the corporate LAN, but can't talk to it using the LPR or IPP protocols, because it is hidden behind the corporate firewall.

You could perform a print-to-file operation, then email the resultant file to somebody at your office, and get them to send it to the printer. But there are a few steps involved.

If your printer happens to have an appropriate Castelle print server, you can install a "Castelle Internet Printer" port on your PC, then configure a local printer which emails its output via a nominated SMTP server direct to the print server. The details can be found at www.castelle.com – and the driver can be found in the downloadable "LANpress Full CD" file under the "Driver/lpd" directory. Similar drivers can be found at www.kingston.com, and at www.brother.com .

You can still play this game, even if you don't have an appropriate print server. All you need do is direct your CIP job to an email address (e.g. printer1@r2d2.acme.com) on a Unix machine which can talk to the printer. You will need a program like that shown hereunder to perform base64 decoding, and a mail alias like that shown near its start to direct messages to it.

I'm not sure how the Castelle/Kingston/Brother people feel about people using their client software without purchasing their hardware, but if it raises customer awareness of product capabilities, its got to be good for them.

There are a couple of issues. Of primary concern is that there is no authentication performed, so anybody with an Internet email account can send jobs to your printer. I don't have an easy solution for this.

The CIP protocol allows acknowledgment emails to be transmitted to any designated email address, and the use of an incorrect address could cause real havoc; this behaviour can be circumvented by commenting the lines in which the "From:" address are collected.

If there is a restriction on the length of emails which can be received through your corporate firewall, you may have a problem printing long documents. The Brother client includes a capability to accommodate this, and I am hoping to supply an extended decoder program for publication in a future edition.

```
#!/bin/sh
# CIPdecode
# Decoder for Castelle/Kingston/Brother Internet
# Printing
# schemes.
#
# Intended for invocation through entry as follows # in
# /etc/alias (or /etc/mail/aliases):
# printer1 "|/usr/local/bin/CIPdecode |lpr -P
# printer1"
#
# Graham K Jenkins, IBM GSA, February 2001.

PATH=/bin:/usr/bin:/usr/pkg/bin                         # Machine dependent.

# Start after line ending in "base64".
# "==" padding is required to overcome Castelle/Kingston client limitation.
awk '{ # Flag set to 1 when "base64" seen, 2 when next blank line seen
       if( Flag == 2 ) { if(length == 0) { Flag=9 ; print "==" }
                         else print $0                                    }
       if( Flag == 1 ) if(length == 0) Flag=2
       if( Flag == 0 ) { Count=split($0, A, "=")
                         if($NF == "base64"   ) Flag=1  # For better security,
                         if($1  == "To:"      ) Dest=$2 # comment next 2 lines.
                         if($1  == "From:"    ) Sour=$2
                         if(A[1]=="BRO-REPLY" ) Sour=A[2]
                         if($1  == "Notify:"  ) Noti=$2
                         if(A[1]=="BRO-NOTIFY") if(A[2]!="None") Noti="Yes" }}
 END { if( Noti == "Yes" ) if ( length(Sour) > 0) if (length(Dest) > 0) {
       print "mailx -s\"Print Job Submitted: "Dest"\" "Sour"|"sh>/dev/null"  }}'|
base64 -d 2>/dev/null
```

Chapter News

While some of these items may be old new by the time you are reading this, it still pays to know the kinds of activities that the various AUUG chapers are organising in your part of the country.

AUUG-CANB

Steve Jenkin <sjenkin@pcug.org.au>

The next meeting of the Canberra AUUG group is on Tuesday 13th March at 7pm. If you may come - could you please e-mail me.

If you have any suggestions for speakers or topics - please! mail me.

The IIDB also meets 5:30 for 6pm every Tuesday night at the National Press. This is meant to be a 'networking' event for all levels of IT people in Canberra.

CLUG - Canberra Linux Group meets the 4th or last Thursday of the month at ANU. Room N101 of the new Computer Sciences Building.

The informal agenda is:- Discussion of next years activities & meeting topics.

The meeting is at the University House (ANU), Cellar Bar - our normal Christmas time venue. For full details, see:-

http://www.canb.auug.org.au/cauug/

"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."

The meetings will be 'informal' for a while until regualr meetings etc are re-established.

See you there for a "refreshing ale", meal perhaps, and catching up on things Unix...

AUUG-CANB

Tuesday evenings, 5:30pm for 6pm, the ACT IIDB, Information Industries Development Board, hosts 'hotspots' at the National Press Club, Barton.

I've been for the past two weeks and can highly recommend them. The IIDB is actively seeking to establish a regular informal networking event for people in IT (and the 'internet' industries too). Their focus is both managers/salespeople _and_ technical types, like me :-) You can register to be on a mail list if you like.

The meeting format is informal - just turn up to the Press Club. There are drinks at the bar, and 'canapes' are brought round. The speakers have no props, no powerpoint, only a microphone and a lecturn. The talks are short - seem like 5 or 10 minutes, and not just marketing spiel. I've heard CISCO talk about networks and creating a future for our children here, and IPEX about their security services and what special things they do... Next week, Feb 20, Kate Carnell will talk on TRANSACT, bound to be popular.

URL's: :http://www.npc.org.au/hotspot/IIDB.HTM for the Press Club info and next talk.

http://www.npc.org.au/hotspot/HotspotDIARYDATE-S.html for general info and full Diary.
http://www.iidb.act.gov.au for the IIDB.

AUUG-VIC

AUUG Vic held a very successful summer BBQ at Albert Part Lake recently. Well over a dozen people attended, ate and drank to Pythonesque proportions and flung frisbees at each other. Also discussed were the various machinations of NetBSD commercial development, SunOS Solaris history the forthcoming F1 Grand Prix at that same venue, and the prions which result in Mad Cow (Jacob-Creutzfeldt) disease outbreaks.

AUUG-SA

On behalf of AUUG I would like to invite you to join David Newall, Greg Lehey and myself (the local AUUG exec members) for some ale and pizza, whilst we do our best to revamp the South Australian chapter of AUUG.

For the time being we're admitting non-members, until they're hooked :)

Agenda
Welcome
Discussion of the aims of AUUG-SA.
Petition to reestablish the chapter.
Relationships with LinuxSA.
Wireless Network Fest.
National InstallFest
Some random technical paper.
Suggestions anyone?
Details
What:
Beer, Pizza, Geeks.
When:
Tuesday 13th March, 7:00pm
Where:
Tellurian, Level 7 North Tce House, 19 North Tce, Hackney.

How:
Optimizations for consumption of beer & pizza to be discussed.
Why:
I was preprogrammed for this sort of thing.
RSVP: Sarah.Bolderoff@cs.unisa.edu.au. Please let us know if you plan to come. There's no obligation, but we'd hate to run out of beer.

If you would like your local AUUG Chapter news and reviews to be listed here, send mail to auugn@auug.org.au

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	
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	TBA	

For up-to-date details on meetings, including those in all other Australian cities, please check the AUUG website at http://www.auug.org.au or call the AUUG office on 1-800-625655.

Membership Application

FRONT

Membership Application

BACK

Volume 22 • Number 1
March 2001

Features:

News:

Regulars: