

# AUUGN

The Journal of AUUG Inc.

Volume 23 • Number 2  
July 2002

## Features:

Dynamically Tune up a File System	9
Wine: Raising a Toast to your Windows Apps	10
Barrel Patrol 3D	12
The Penguin and the Hare	14
Cracking Open Proprietary Envelopes	16
Setting up a Squid Proxy Server	18
Building the Lo-fat Linux Desktop	22
Writing Documentation, Part 1: POD	26
Bookreview: The qmail Handbook	31
Compiler Design with Python	32
Configuring GDM 2.2	35
Improving Hard Disk Performance with <i>hdparam</i>	36
Discovering the Universe: Celestia and Open Universe	38
Mandrake 8.2: First Impressions	41
GUI Programming in C++ using the QT Library, Part 1	43
A Linux Fax Server for a Windows Network	45
Securing a Heterogeneous Network with Free Software Tools	47
Graphic Programming with <i>libtif</i>	53

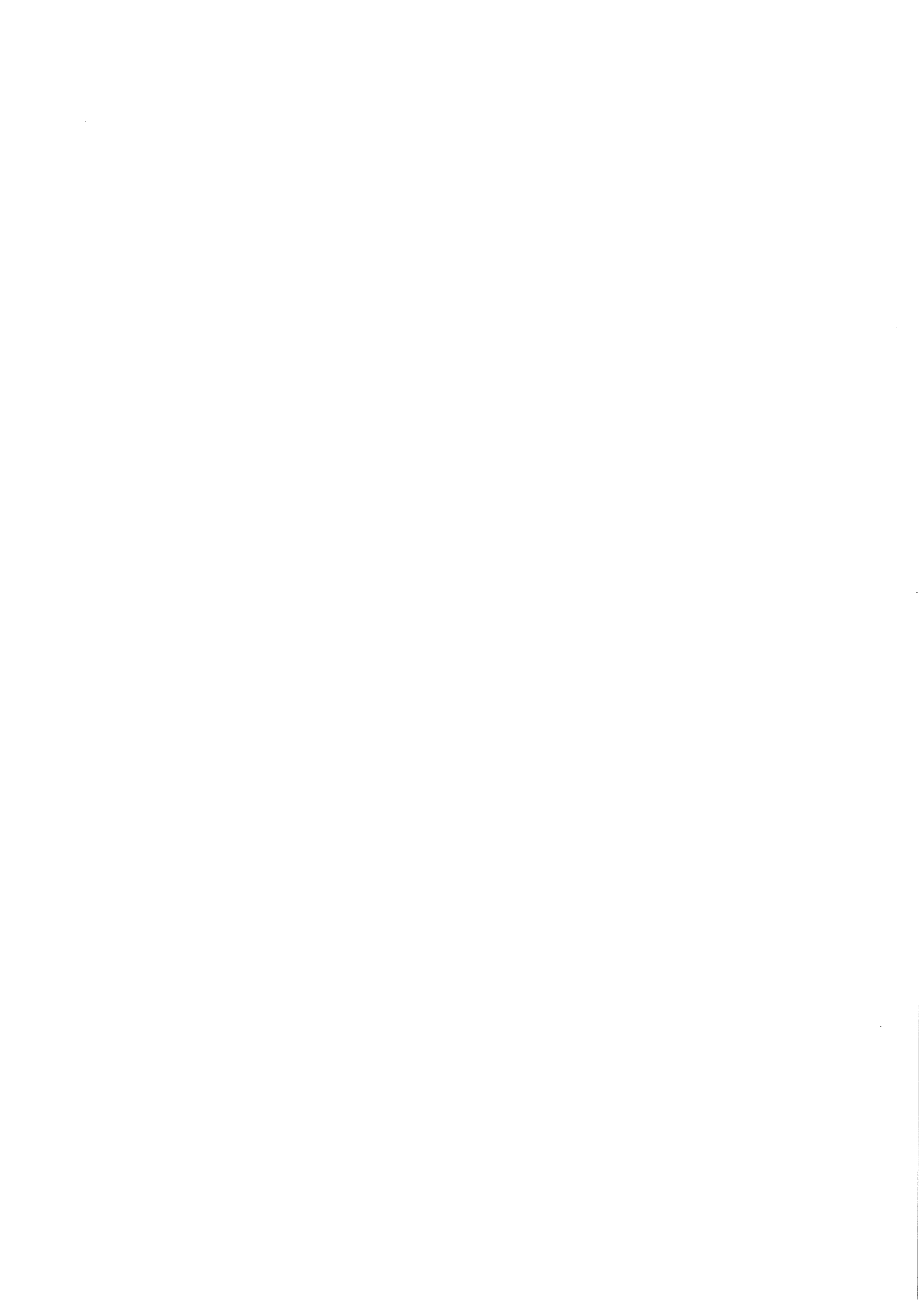
## News:

Public Notices	5
AUUG: Corporate Members	10
AUUG: Chapter Meetings and Contact Details	59

## Regulars:

President's Column	3
/var/spool/mail/auugn	4
My Home Network	5





**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](mailto:auug@auug.org.au)

**AUUG Management Committee**

Email: [auugexec@auug.org.au](mailto:auugexec@auug.org.au)

**President**

Greg Lehey  
IBM Australia  
PO Box 460  
Echunga, SA, 5153  
Bus. Tel (08) 8388 8286, Mobile 0418 838 708, Fax (08) 8388 8725  
<[Greg.Lehey@auug.org.au](mailto:Greg.Lehey@auug.org.au)>

**Immediate Past President**

David Purdue  
Sun Microsystems  
Level 6, 476 St Kilda Road  
Melbourne, Victoria, 3004  
Phone: +61 3 9869 6412, Fax: +61 3 9869 6288  
<[David.Purdue@auug.org.au](mailto:David.Purdue@auug.org.au)>

**Vice-president**

Malcolm Caldwell  
Bus. Tel (08) 8946 6631, Fax (08) 8946 6630  
<[Malcolm.Caldwell@ntu.edu.au](mailto:Malcolm.Caldwell@ntu.edu.au)>

**Secretary**

David Bullock  
0402 901 228  
<[David.Bullock@auug.org.au](mailto:David.Bullock@auug.org.au)>

**Treasurer**

Gordon Hubbard  
Custom Technology Australia Pty Ltd  
Level 22, 259 George Street, Sydney NSW 2000  
Bus Tel: 02 9659 9590, Bus Fax: 02 9659 9510  
<[Gordon.Hubbard@auug.org.au](mailto:Gordon.Hubbard@auug.org.au)>

**Committee Members**

Sarah Bolderoff  
FourSticks  
Suite 2, 259 Glen Osmond Rd,  
Frewville, South Australia, 5065  
<[Sarah.Bolderoff@auug.org.au](mailto:Sarah.Bolderoff@auug.org.au)>

**Adrian Close**

Fernhill Technology  
Level 1, 3 Colman Road, Warranwood VIC 3134  
Mobile: +61 412 385 201, <[adrian@auug.org.au](mailto:adrian@auug.org.au)>

**Stephen Rothwell**

IBM Australia, Linux Technology Center  
8 Brisbane Ave, Barton ACT 2600  
Business phone: 02 62121169  
<[Stephen.Rothwell@auug.org.au](mailto:Stephen.Rothwell@auug.org.au)>

**Andrew Rutherford**

Iagu Networks, 244 Pirie St  
Adelaide, SA, 5000  
Bus. Tel (08) 8425 2201, Bus. Fax (08) 8425 2299  
<[Andrew.Rutherford@auug.org.au](mailto:Andrew.Rutherford@auug.org.au)>

**Mark White**

apviva technology partners  
P. O. Box 1870, Toowong QLD 4066  
Bus Tel 07 3876 8779, Mobile 04 3890 0880  
<[Mark.White@auug.org.au](mailto:Mark.White@auug.org.au)>

**AUUG Business Manager**

Elizabeth Carroll  
PO Box 366  
Kensington NSW 2033  
<[busmgr@auug.org.au](mailto:busmgr@auug.org.au)>

## Editorial

Con Zymaris

[auugn@auug.org.au](mailto:auugn@auug.org.au)

We are in the midst of one of the most economically challenging periods in the history of the IT industry. It's the worst I've seen, while tracking it for the past 23 years. In the brief precipice afforded by this space, I wanted to broach and review some pertinent questions which likely interest us all: Why are we in this current state? Does it affect the consumers and purveyors of the advanced computing platforms which constitute the AUUG community? What do we do about it?

Why we are here. In my view, a number of factors. Illogical exuberance on the part of the Wall Street types, who helped inflate the tech stock bubble to such an extent, that servicing the expectations of return on investment became the equivalent of the Herculean effort needed to clean the Augean stables. This had the flow-on effect of releasing the 'inner-child' in too many otherwise dour CEOs worldwide, allowing them to see and invest heavily in supposedly viable over-reaching projects in IT within their own organisations, many of which became software Titanics, oft-times sinking without a trace, sometimes taking the whole company with them. The result of all this Sisyphean toil is to create the current vibe which permeates many of our employers or clients, namely that IT has taken them for a ride, that we, the purveyors of IT are living the undeserved high-life, and that they will not be duped in this manner ever again.

The shocking truth is that they have a point. Our industry's delivery of successful projects, reliable products and measurable return on investment are sub-par at best, and verging on mischievous negligence at worst. How can these employers and clients not feel ripped if they are hit with new viruses each and every day of the year; when they are repeatedly told by vendors that the latest and greatest of their products are now, finally reliable and secure, only to be proven wrong days or weeks later, and when the biggest supplier of IT, and the richest company in the history of mankind, is extorting its 500 million clients worldwide through an effective doubling of the cost of licencing its software?

What do we, in AUUG do about this? The answer, my friend, is streaming down the ether; deliver the best value-for-money our employers and our clients have ever seen in systems solutions, practice and delivery. Solutions that are increasingly robust and secure. Solutions that are based on solidly engineered technologies: Unix and Linux.

Cheers,

Con

# Contribution Deadlines for AUUGN in 2002

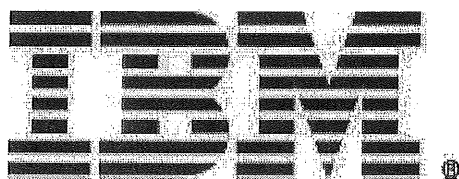
---

Volume 23 • Number 3 – September 2002: **August 30<sup>th</sup>, 2002**

Volume 23 • Number 4 – December 2002: **November 15<sup>th</sup>, 2002**

---

AUUG Incorporated gratefully acknowledges the support of its corporate sponsor:



---

## AUUGN Editorial Committee

The AUUGN Editorial Committee can be reached by sending email to: [auugn@auug.org.au](mailto:auugn@auug.org.au)

[auugn@auug.org.au](mailto:auugn@auug.org.au)

Or to the following address:

AUUGN Editor  
PO Box 366  
Kensington NSW 2033

Editor:  
Con Zymaris

Sub-Editors:  
Frank Crawford

Public Relations and Marketing:  
Elizabeth Carroll

---

## AUUGN Submission Guidelines

Submission guidelines for AUUGN contributions can be obtained from the AUUG World Wide Web site at:

[www.auug.org.au](http://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

Copyright © 2002 AUUG Inc.

All rights reserved. Portions © by their respective authors, and released under specified licences.

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

Copyright without fee is permitted, provided that copies are made without modification, and are not made or distributed for commercial advantage.

---

# President's Column

David Purdue, <David.Purdue@auug.org.au>

**farewell** *n.* **1** a wish of well-being at parting. **2** an act of departure. **b** a formal occasion honouring a person about to leave or retire. -- *Merriam-Webster Collegiate Dictionary*

So now I have come to the end of four years as the President of AUUG Incorporated, and so I want to reflect on what has been done, what has been missed, and where we should go from here.

I think it would be overstating the case to say that the organisation has turned around under my leadership. However, I do think we have focused on what we do well and that AUUG does deliver improved member benefit.

One of the main reasons for AUUG's existence is to provide information to member, to let them learn – primarily to learn from each other. And we have dramatically improved our performance here in two main areas: events and AUUGN.

The introduction of the one-day symposia has increased the opportunities for AUUG members to network and exchange ideas – in particular the Australian Open Source Symposium and the AUUG Security Symposium have gained a life of their own.

Under the capable hands of Con Zymaris AUUGN has become a mine of information. It has got to the stage that the board has had to impose a page limit on editions of AUUGN, something we never thought we would have to do. Many thanks to Con for the work he has done.

And we have established a lot of policy and procedure to enable smoother day-to-day running of AUUG and in particular the annual conference.

I do have some regrets – I think there is still a lot to achieve that I did not get to. I believe there is demand for a couple more one day symposia, but we need to find volunteers to run them. We also have continuing plans for improved delivery of electronic services, but have lacked the time to implement them. And while we have established procedures we need to spend some more time documenting them.

I think a problem that the whole board faces is finding enough time in the current work environment to do everything that needs to be done. Certainly over the past two years I have become busier at work rather, and work now requires much more of my time and other activities are getting squeezed out.

The newly elected board consists of a lot of new faces, and I think these are people at a stage in their career that they can devote more time to volunteer activities. However I believe their greatest challenge and their greatest opportunity is to increase member numbers.

This will have many benefits – it provides AUUG with the income stream needed to support new activities, it increases the value of AUUG membership since there are more AUUG members to network with, and it provides a larger pool of volunteers to organise activities.

I will still be around AUUG, since AUUG provides me with a lot of benefit and I wish to put something back. This year I will serve as Immediate Past President – keeping an eye on the board without too much active involvement. I have also taken a post on the Victorian Chapter Committee.

In closing I would like to thank all the AUUG board members who have supported me over the last four years – in particular I would like to thank Michael Paddon who has fed me ideas and advice, and Luigi Cantoni who has rationalised the way AUUG handles money. Thanks, of course, to Liz Carroll, without whom AUUG could not function.

Best of luck to Greg Lehey, your incoming President, and I hope to see many of you at our conference in September.

# /var/spool/mail/auugn

Editor: <auugn@auug.org.au>

Well, your editor's inbox has finally been deluged with AUUG-related email, a sign from providence that interesting things are a-foot on the auug-talk mailing list. If you too want to have your say in where the good ship AUUG should be heading, speak to our good pal, the *mailman*:

<http://www.auug.org.au/mailman/listinfo/talk>

**From: Conrad Parker <conrad@vergenet.net>**  
**Subject: Re: [Talk] AUUG'S Declining Membership**

On Wed, Jul 24, 2002 at 02:39:42PM +0930, david.newall@auug.org.au wrote:

> The recent discussion on AUUG and LUGs  
> prompted me to write this:

>  
> SUMMARY

> How do we reverse declining membership?

> o Mailing lists should be available to everybody

> o Conference CFPs sent to all Asia Pacific

> universities

> o AUUGN on web, not on paper

> o Exec to meet electronically

> o Can we afford a business manager?

Hi,

I just wanted to chime in on this last point. To make a comparison, outside of AUUG, I've been involved with SLUG and linux.conf.au, and with AUUG I was on the board last year and helped with AOSS4 which was earlier this month. Having a business manager makes all the difference. It means that if you decide to do an event or chase up sponsorship or whatever, it gets done.

Often in a purely voluntary (eg. LUG) environment, good ideas get passed over because there simply aren't enough motivated people to go around, especially to do all the non-technical work that is involved in organising events etc. LUGs do a lot of good work, but it's hard to guarantee consistency or that you will properly cater for all members -- fun, interesting stuff tends to get done well but there are always tasks that fall by the wayside.

What we really should be doing is exploiting the fact that we have a business manager who can help with sponsorship and venues and so forth. To reverse declining membership, I suggest we need to simply do stuff (and more visibly) that potential members would want, and having a business manager makes doing that a far more realistic proposition.

Getting such stuff happening requires nothing more than people who want it getting off their backsides

and making it happen. If someone, or even a LUG, wants a symposium in their town, it's far easier for them to get in touch with AUUG and let AUUG take care of the business hassles than to go it alone. This is the message we should be sending out, and offering to help LUGs etc. rather than pretending they are some kind of competition.

AUUG has a lot to offer that differentiates us from LUGs, and which would allow us to cooperate, not compete, with them. Differences include these three things:

- a printed quarterly newsletter, with an ISSN and a real live editor.
- a board that meet regularly (in a non-sterile "IRL" environment) and can coordinate things nationwide
- a business manager, which provides assurance that business stuff gets done, and allows the geeks to concentrate on being geeks.

Rather than cutting these things back, let's first make an effort to actually let people know they exist. AUUG does great stuff and complements what is offered by LUGs -- we'd all be better off if we cooperate and offer to share resources. If we cut these things back instead, I reckon we'd very quickly become irrelevant.

Conrad.

**From: Greg Rose <ggr@qualcomm.com>**  
**Subject: Re: [Talk] AUUG'S Declining Membership**

As one of the people who remember a time when AUUG \*didn't\* have a business manager, trust me... Liz is probably worth more than the cents we pay her, and is probably the glue that's holding the group together.

The problems facing AUUG are serious, and I don't know how to solve them. But slashing expenses can only go so far, and I think AUUG is about as bare bones as it can be.

USENIX has somewhat transmogrified itself into a much bigger and more diverse organisation, incorporating SAGE, security, the free Unix movement, and other stuff. Except for them, AUUG is the last remaining "large" UUG; the others that are still going, like NLUUG, are smaller than most of our regionals! Their life is easier, with more concentrated populations.

Dunno what to suggest, except that it's a hard problem.

Greg.

Greg Rose  
INTERNET: ggr@qualcomm.com

# Public Notices

## Upcoming Conferences & Events

### SAGE-AU 2002 Conference

Melbourne, August 5 - 9, 2002

### AUUG'2002 Annual Conference

Melbourne, September 4th - 6th

### LISA '02

16th Systems Administration Conference  
November 3-8  
Philadelphia, PA

### IMW 2002

Internet Measurement Workshop 2002  
November 6-8  
France

### OSDI '02

5th Symposium on Operating Systems Design and Implementation  
December 9-11  
Boston, MA

# My Home Network (July 2002)

By: Frank Crawford <[frank@crawford.emu.id.au](mailto:frank@crawford.emu.id.au)>

Welcome to another edition of my column. This one will be a bit of catching up on some old items. After nearly three years of writing about my home network, there are a few things that I've planned, or changed or just plain completed and I want to give you some idea what happened.

## SPAMASSASSIN - KILLING SPAM

Firstly, following on from last issue's antivirus work, another huge problem is SPAM, or unsolicited email. I'd like to hear from anyone who has an Internet accessible email account who hasn't received SPAM at some time. It is a big problem and one that is getting bigger. Legal measures are proposed, technical solutions are being developed and action groups are acting. Unfortunately, most often it is up to the end-user to eliminate the SPAM.

One commonly used tool is a package called SpamAssassin, which can be found at <http://www.spamassassin.org>, or the Australian mirror at <http://au.spamassassin.org>. It is basically a Perl module, together with a couple of programs that use this module and examine each mail item. It can be used either site-wide or on an individual basis, and has both global and individual configuration files.

Installation is fairly simple if you've had some experience with Perl, being basically the same as any other Perl module, i.e.

```
cd Mail-Spamassassin-*
perl Makefile.pl
make
make install
```

or via the CPAN module. Of course, like most large Perl modules, there are a host of other modules that it depends on, in particular:

- Net::DNS - available from CPAN, and which I already had for other things,
- Razor - from <http://razor.sourceforge.net/>, used from checking against an external SPAM filtering network, and which I didn't implement for my setup, and
- Mail::Audit, Mail::Internet, Net::SMTP - needed in certain cases, and again, I didn't need for my setup (although I think some were already installed).

Aside from the module, this will also install `spamd`, `spamd`, `spamassassin` and `spamproxyd`. The program `spamassassin` is for interactive and batch processing of mail files, `spamd` is a daemon to do the same thing which communicates via `spamd`, while `spamproxyd` is designed to work directly from some

Linux, Unix  
and Windows

## Cybersource

Consulting, Training  
and Development



*Cybersource is a professional services consultancy specializing in the areas of Unix, Linux, and Windows. We provide network consulting, staff training, and application development services and have over 10 years experience in the industry.*

*So if your organization has a need for systems and network administration, security and auditing, or web based application development, you know who to call.*

Web: [www.cyber.com.au](http://www.cyber.com.au)  
Mail: [info@cyber.com.au](mailto:info@cyber.com.au)

Phone: +61 3 9642 5997  
Fax: +61 3 9642 5998

MTA's (Mail Transport Agent), in particular "postfix". The entire message is piped through spamassassin, and via a set of rules and heuristics, it generates a score for the mail, higher values are likely to be SPAM, lower values are likely to be good. You then set a threshold and items above this level are tagged as SPAM.

The recommended mechanism to do all this processing is with "procmail", i.e. you set up your ".procmailrc" with a line like:

```
:0fw
| /usr/bin/spamassassin -P
```

This will then add a header line of the form: "X-Spam-Status: Yes" (or "No") to the mail, which you can then process further with procmail, e.g.

```
:0:
* ^X-Spam-Status: Yes
  caughtspam
```

I won't go further into the use of "procmail" here, except to say it can either automatically delete or store the SPAM so you never see it. I will also mention that the standard sendmail configuration for Red Hat Linux uses procmail for the final delivery so there are no installation issues for it.

Now I have to admit, I haven't let SpamAssassin totally take over my mailbox, rather, I let it tag my mail (aside from the headers, it adds the words "\*\*\*\*\*SPAM\*\*\*\*\*" to the Subject). From this I can quickly scan my mailbox for non-spam and then handle the SPAM at my leisure. In fact, spamassassin isn't infallible, there are a few mail items I receive that are flagged as SPAM, that aren't (e.g. mail from BigPond regarding my ADSL connection).

Luckily, this can be fixed in the local configuration file, "~/.spamassassin/user\_prefs", where you can include details of address that my look like SPAM but aren't. Unfortunately, this is almost always a reactive item, as you are not aware how close things look until you see why they are marked as SPAM. To make matters more fun, some items (like the BigPond mail) try so hard to look like SPAM that you can't differentiate it. For example, BigPond's mail comes with no sender (i.e. "From: <>") no recipient (i.e. "To: Bigpond Customer <>") and various SPAM triggering words (e.g. "Click Here", "Dear Customer", etc.). Oh, well, most of the time it is useless information anyway.

For all those other lists that aren't SPAM, despite matching, you add lines in "~/.spamassassin/user\_prefs", such as:

```
whitelist_from RedHat@redhat.*.com
```

## PPPD - AUTODIALING AND DISCONNECTING

In the October 1999 column I mentioned that `pppd' now support dialing on demand, but at the time I

wasn't able to use it because of a few features I couldn't use. In particular, while `pppd' will disconnect when the line has been idle for some time, any traffic on the line such as broadcast packets, is sufficient to hold the line up. For `pppd' on \*BSD system, there is an option to ignore certain packets, it requires kernel support, which, at the time (i.e. the Linux 2.2 Kernel), wasn't available for Linux.

With the release of Linux 2.4, the kernel support of this feature was added. To make it available, you need to configure your kernel with both "CONFIG\_FILTER" and "CONFIG\_PPP\_FILTER" both set to "Y". This is the case for my home built kernel, and also true for the standard Red Hat Kernel.

So, at this point, the kernel support is available, and the next point is to configure the options for `pppd'. Unfortunately, as this point Red Hat falls down. While `pppd' now supports filtering, the version included with Red Hat 7.3 does not have it compiled in. This gave me three options: go back to dial; get the `pppd' source and do it myself; or get a source RPM and modify it appropriately. While the last two options sound the same, it is well known that Red Hat make a lot of configuration changes and other modification, which it would be good to include in any version on the system. So, just for the fun of it, I chose the third option, i.e. modify the source RPM.

So, for this I downloaded the source RPM "ppp-2.4.1-3.src.rpm", unpacked it with the command "rpm2cpio ppp-2.4.1-3.src.rpm | cpio -idvm" (note: you could also install the RPM, but that puts it in a system directory).

To understand the next step, you need to understand a bit about how RPM's built. Each source RPM contains the original distribution tar file (e.g. ppp-2.4.1.tar.gz), a number of patches, and a "spec" file, which outlines all the info, changelog and what patches to apply. When the binary package is built, each patch is applied and then "make" is run. (In fact this is a bit simplified, as again it is all controlled by the spec file.)

In this case, once unpacked, I modified one of the patch files by hand to enable the required option and rebuilt the binary RPM. I could have generated a new patch file, and this probably would have been the more correct approach, but .... I also made a change to the spec file to give it a unique version number (from "3" to "3fpc").

The final part of construction is of the binary RPM, using rpmbuild, with:

```
rpmbuild -ba --define=_topdir `pwd` /ppp_rpm"
ppp_rpm/SPECS/ppp.spec
```

This will use the spec file I created in the directory structure "ppp\_rpm" to build a complete binary RPM. With this RPM, I can then install it with the normal tools, i.e.

```
rpm -Fvh ppp-2.4.1-3fpc.i386.rpm
```



## UPGRADES - WINXP & RED HAT 7.3

To set up `pppd` to use its filtering capability you need to add the option "active-filter" which controls which packets are used to mark the link as active. There is another option "pass-filter" which drops packets that don't match. The syntax for packet matching is the same as used by "tcpdump", for example in "/etc/ppp/peers/active-filter" I have:

```
active-filter 'not ip multicast'
```

This will cause all packets that are IP Multicast to be ignored for link activity status.

With this option, you can setup an on-demand PPP link, which will connect on activity, and then drop the connection after the link goes idle.

Again here Red Hat drops the ball, as their scripts for managing PPP connections do not correctly handle dial-on-demand links. In fact, it can cause major problems, because it hangs on startup, waiting for the connection to complete. Of course this may never happen.

I've developed a fix for this, which, although not the most efficient, does work. The patch for this is:

```
--- ifup-ppp.dist      Mon Apr 15 12:35:32 2002
+++ ifup-ppp          Sun Jun  9 16:04:21 2002
@@ -15,7 +15,14 @@
 [ -f "${CONFIG}" ] || CONFIG=ifcfg-${1}
 source config
 # don't start ppp-watch by xDSL
- [ "$TYPE" = "xDSL" ] || exec /sbin/ppp-watch
"${DEVICE}" "$@"
+ if [ "$TYPE" != "xDSL" ] ; then
+   if [ "$DEMAND" = "yes" ] ; then
+     /sbin/ppp-watch "${DEVICE}" "$@" &
+     exit 0
+   else
+     exec /sbin/ppp-watch "${DEVICE}" "$@"
+   fi
+ fi
CONFIG=$1
```

So, with these various steps in place, to bring up an on-demand link, all that is needed is to define a configuration file, such as /etc/sysconfig/network-scripts/ifcfg-ppp1:

```
DEVICE=ppp1
IPADDR=192.168.1.250
NETMASK=255.255.255.0
MODEMPORT=/dev/modem
LINESPEED=115200
HARDFLOWCTL=yes
DEFROUTE=no
IDLETIMEOUT=1800
PPPOPTIONS="call active-filter"
PEERDNS=no
DEMAND=yes
ONBOOT=yes
```

One interesting item I found was that the `active-filter` keyword would not work in PPPOPTIONS, as it was difficult getting the quoting correct through all the scripts called.

As longtime readers of this column will remember, aside from Linux, I also have Microsoft systems. Up until recently they were all running Windows ME, but I've taken the opportunity to upgrade to Windows XP. I have to admit, that XP is a real operating system, providing multitasking, security and all the feature you see in even the oldest Unix systems. In fact, Windows XP comes from the Windows NT line of operating systems, not the Windows 95 line.

So, how did I find the upgrade. I had the opportunity to install Windows XP Professional edition on one system and Windows XP Home edition on another. Previous comments indicated that, while upgrading from Windows ME is supported, reinstalling may be preferred. Given the fun and games I had, it is certainly true.

The first step in the upgrade is a compatibility check, and this flagged a large number of system and other related programs that would need to be re-installed or modified. Even more of a concern, this was not only system level programs, but also certain games (for example Ages of Empires) and, of course, Office XP. In hindsight, this isn't surprising, as many of the original settings had no knowledge of users or permissions. One interesting trick that Microsoft play is to replace some of the shortcuts with ones to a script that says that the requested program needs to be upgraded before use.

One other note, at the start of this compatibility check, the install program requests to download an update for the install programs. I presume this is an update to the compatibility matrix.

The actual install ran as do most Windows installations copying, rebooting, checking, waiting, rebooting, etc. In general, it took about an hour to complete and get to a position that I was again in control of the system.

The first noticeable difference was that there are now real users and a requirement to log in. Of course as part of the cosmetic changes there is a new login screen, with a "graphical login" feature, although this is disabled when the system is used in any co-operative environment (e.g. Win2K domain or dialup before login).

As well, initially there are only two accounts, "administrator" and one named after the hostname, both of which are in the "administrators" group. Setting up these accounts is easy, however, changing them is another matter.

This was the first place I found a difference between the Professional and Home editions. Under Professional, the user manager, which is familiar to Windows 2000 users works, but under the Home edition all you get is a message saying it is not supported for this version.

A second and more pleasant change is the new multi-user facilities now built in. Unfortunately, again there are some limitations. For a start, while both versions allow a user to "switch", i.e. disconnect their session and log in as someone else, only WinXP Pro allows you to connect remotely. Even more, you can only have one person logged on at any time. If someone tries to connect remotely, while someone is on the systems console, the other person is disconnected.

As you can imagine, not everything worked as expected. On a "recent" machine, i.e. my 450MHz Pentium III, with recent cards, motherboard, etc, most hardware was identified fine, software installed correctly, and/or compatibility issues were correctly flagged. On my older system (a 466MHz Celeron with old hardware) the network card was entirely ignored, causing a lot of follow on errors, as expected. Digging deeper, the network card was ignored because it was a non-PNP ISA card, i.e. It had jumpers to set the IRQ, etc (I told you it was old hardware), and WinXP only supports PNP-ISA or PCI network cards. Due to the number of different systems and components I have in the house, I was ultimately able to find a suitable card to use. One I stole out of my Linux server. Even more embarrassing, after a bit of further study, I found that it did have an option to enable the card to be PNP. However, the program to enable it only ran under DOS, not WinXP, hence my lack of success earlier in configuration. (This is an obvious side-effect of a proper O/S, i.e. user level programs are not able to access the hardware directly).

One other piece of hardware that no longer worked was an old QIC Floppy tape drive. No matter where I look, I can't find a suitable driver, and I guess my long term solution is to put it on a Linux system.

Software wise, not all the compatibility issues were diagnosed. Certainly, I had to spend a fair bit of time finding CD's to re-install programs that previously worked. Not all of this was Microsoft's fault, for example, one game installed the wrong executable, because we were not in the USA, and wouldn't do anything. To make matters worse, the support people couldn't find a problem, because they were all in the USA!

The other big issues I still have is with accounts and permissions. Firstly, WinXP Home edition does have a much simple authentication mechanism. In particular, there are only two groups, "administrators" and "limited". There is also a special configuration for "guest".

Under WinXP Pro edition, all the features found in Win2K are available, but the simplified GUI only show the same limited groups (i.e. "administrators" and "limited").

Even more of a problem, almost all software is installed for an individual. For example any users who want to share access to a game need to be part of the "administrators" group, which defeats the purpose

of security.

A second problem I've had with accounts is under WinXP Home edition, is that the original account created for the machine name cannot be removed or changed, while I would like to change it to be my account. Under WinXP Pro, I was able to use the User Manager to change it.

All in all, I'd have to say that WinXP is a good operating system, with lots of potential, but too many of the programs installed still act as if it is Win95.

It is interesting to compare the upgrade of WinXP to my recent upgrade to Red Hat 7.3. Of course, this is a bit unfair, as Red Hat was only a minor change, from 7.2 to 7.3, whereas the WinXP upgrade was a major change.

The open nature of Linux has meant that all the old hardware I have is still supported, and in general just continues to work. Of course, this isn't always true, and, for example, after upgrading to Red Hat 7.3, the cursor under X no longer worked correctly. Eventually I traced this down to a problem with the XFree86 driver from Matrox. Replacing it with the standard one from the distribution fixed the problem. Similarly, tracking down minor changes and difference in configuration files always brings in problems.

One such issue was with the update to ntpd, and particular ntp.conf, which now includes access restriction, including blocking by default, and much to my surprise blocked the access by the ntp server.

To cater for these issues, whenever I do such an upgrade, I run a simple find across the entire system to locate all saved configuration files. Given that on installation of an RPM should save any changed configuration files with an extension of either ".rpmnew" or ".rpmsave" (or occasionally with other similar extensions), I run:

```
find / -name "*.rpm?*" -print
```

and then examine each file listed (usually not many) to find the differences (usually using `diff`). Where possible, I try to keep a copy of the original configuration file before I make a change, which makes it much easier to reapply the change to the new files.

Of course, not every change is saved in this fashion, and occasionally I've had to go to my backups to find a change that I have made, particularly for scripts.

So, that kind of brings up to date a number of things that have gone on over the last while, but I certainly have plans for future changes.

As a final point, I'd like to remind you of all the exciting activities coming up for AUUG. The Fourth Australian Open Source Symposium will be in Sydney on 20th July, while AUUG'2002 will be in Melbourne

in the week of the September 1st. I hope to see you at one if not both of them.

# Dynamically tune up a file system

Author: Joseph Gan <joseph.gan@abs.gov.au>

Nowadays, the performance becomes a issue in terms of the file system tuning in Solaris. Changing some of file system's parameters that usually will destroy the data on the file system.

For instance, changing the cache segment block size in the volume of a T3 requires that you delete the existing volume; deleting volume will destroy the data. And the volume on the T3 has to be reinitialised, which can take a significant amount of time for a large disk space.

And also, change the segment size of a LUN in a raid box which needs to delete the existing LUN etc.

Even if changing the parameter of a metadevice, or rename a metadevice under SDS (Solstics DiskSuite) which needs to un-mount the file system.

How to dynamically change the parameters of the file system without destroy the data on it?

First, the file system has to be created and mounted as one-way mirror metadevice, in the following example, d100 which contends d101 as its submirror:

```
# metastat d100
d100: Mirror
  Submirror 0: d101
  State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 10261520 blocks

d101: Submirror of d100
  State: Okay
  Size: 10261520 blocks
  Stripe 0: (interlace: 32 blocks)
    Device          Start Block  Dbase
  State      Hot Spare
  Okay       c1t12d0s0          0         No
  Okay       c1t13d0s0       1520        No
  Okay       c1t14d0s0       1520        No
  Okay       c1t15d0s0       1520        No
```

Next step is to create a new metadevice d102, which must be the same size of the submirror d101 with a set of new parameters.

For T3, you need a spare disk volume. For the raid box, you need a set of spare disks and so on.

Then, add the new metadevice d102 as the second submirror to d100, resync will automatically take place.

After the resync has done, you have a two way mirrors. One submirror has the old parameters, and the other has the new parameters.

Finally, you can detach the old submirror d101 from d100, and remove metadevice d101 all together.

Now, you have got the file system with a set of new parameters dynamically.

# AUUG Corporate Members

as at 1st June 2002

- ac3
- Accenture Australia Ltd
- ANSTO · ANU
- Aust Centre for Remote Sensing
- Australian Bureau of Statistics
- Australian Industry Group
- Bradken
- British Aerospace Australia
- Bureau of Meteorology
- C.I.S.R.A.
- Cape Grim B.A.P.S
- Centrelink
- CITEC
- Computer Science, Australian Defence Force Academy
- Computing Services, Dept Premier & Cabinet
- Corinthian Industries (Holdings) Pty Ltd
- CSC Australia Pty. Ltd.
- CSIRO Manufacturing Science and Technology
- Curtin University of Technology
- Cybersource Pty. Ltd.
- Deakin University
- Department of Land & Water Conservation
- Energex
- Everything Linux & Linux Help
- Fulcrum Consulting Group
- IBM
- ING
- IT Services Centre, ADFA
- Land and Property Information, NSW
- LPINSW
- Macquarie University
- Multibase WebAustralis Pty Limited
- Namadgi Systems Pty Ltd
- NSW National Parks & Wildlife Service
- NSW Public Works & Services, Information Services
- Peter Harding & Associates Pty. Ltd.
- Rinbina Pty. Ltd.
- Security Mailing Services Pty Ltd
- St. John of God Health Care Inc.
- St. Vincent's Private Hospital
- Stallion Technologies Pty. Ltd.
- Sun Microsystems
- TAB Queensland Limited

- The University of Western Australia
- Thiess Pty Ltd
- Tower Technology Pty. Ltd.
- Uniq Advances Pty Ltd
- University of Melbourne
- University of New England
- University of New South Wales
- University of Technology, Sydney
- Victoria University of Technology
- Workcover Queensland

## Wine: Raising a toast to your Windows Apps on Linux

Author: Gaurav Taneja  
[tech@gauravtaneja.com](mailto:tech@gauravtaneja.com)

When I first started working on Linux some years back I did rely on a standby Windows OS installed in another partition in case something would fail. But soon I realized that this Open Source OS had everything to offer without my spending a penny. However, there are still times when I feel the need to fire up some application that's written exclusively for Windows. What do you do in such a situation? The answer is Wine.

### WINE HAS NOTHING TO DO WITH LIQUOR !

There are products like VMWare & Win4Lin that will let you run another OS (usually Windows) on a running Linux machine so that you can run your Windows programs. You could also go in for a more traditional approach of having another partition with Windows installed on it. However, these alternatives are more of an overhead on your system than a solution. Wine stands apart from all these options, Wine, which stands for "WINE Is Not an Emulator", doesn't require you to buy a Windows licensed copy. It accomplishes this by rewriting the complete Win32 APIs which differs from the Microsoft Code.

### LET'S RAISE A TOAST

If you would like to try WINE, you can get the latest sources from the WINE headquarters at <http://www.winehq.com>. Building from source may not be necessary. The site has links to daily builds in many different formats. Should you wish to obtain the source and build for yourself, you'll find that it is pretty straightforward.

The following steps can be taken to accomplish your task:

```
gunzip Wine-20020411.tar.gz
tar -xvf Wine-20020411.tar
cd wine-20020411
./configure
make depend
make
```

```
make install
```

If you are interested in the bleeding-edge version of Wine and as a matter of fact any major software you should follow the path of CVS. The latest source in the CVS tree might prove to be more efficient in terms of raw performance.

The following procedure can be followed to grab the latest source:

```
export
CVSROOT=:pserver:cvscvs@winehq.com:/home/wine
cvsnpres login
```

When asked for password, provide 'cvscvs':

```
cvs -z 3 checkout wine
```

You will see a steady stream of files coming into a directory called "wine" relative to your current directory. After the whole process is complete you can follow the same procedure of compilation as above.

### WINE CONFIGURATION

We will need a configuration file called "config" in the "~/.wine/" directory. You can copy the a sample of the same from the source directory:

```
cp documentation/samples/config ~/.wine/config
```

The "config" file might appear daunting at a first glance but you better stick to the defaults and change only the critical parts that relate to your system. You will encounter a section something like this:

```
[Drive A]
"Path" = "/mnt/fd0"
"Type" = "floppy"
"Label" = "Floppy"
"Serial" = "87654321"
"Device" = "/dev/fd0"
[Drive C]
"Path" = "/c"
"Type" = "hd"
"Label" = "MS-DOS"
"Filesystem" = "win95"
[Drive D]
"Path" = "/cdrom"
"Type" = "cdrom"
"Label" = "CD-Rom"
"Filesystem" = "win95"
; make sure that device is correct and has
proper permissions !
"Device" = "/dev/cdrom"
```

Wine actually tries to emulate a DOS-like drive and folder structure so the section which starts with "[Drive C]" indicates the mapping of a hypothetical drive C: to your linux directory which in our case is "/c".

Next, some system folders like "windows" and "system" are also mapped like this in the "[wine]" section:

```
"Windows" = "c:\\windows"
"System" = "c:\\windows\\system"
"Temp" = "e:\\"
```

```
"Path"=
"c:\\windows;c:\\windows\\system;e:\\;e:\\test;f:\\
\"
"Profile" = "c:\\windows\\Profiles\\Administrator"
"GraphicsDriver" = "x11drv"
; Wine doesn't pass directory symlinks to
Windows programs by default.
; Enabling this may crash some
programs that do recursive lookups of a whole
; subdir tree in
case of a symlink pointing back to itself.
;"ShowDirSymlinks" = "1"
"ShellLinker" = "wineshelllink"
```

We will have to create some of the basic Windows directory structure ("windows", "system" directories as mentioned in the "[wine]" section in the "config" file:

```
cd /c
mkdir -p windows/system
mkdir -p windows/Start\ Menu/Programs
```

The "[DllOverrides]" section of the config file handles the DLLs that are supplied with Wine to be used in place of their Windows counterpart and some native Windows DLLs that you might want to use:

```
; Be careful here,
wrong DllOverrides settings have the potential
; to pretty much
kill your setup.
[DllOverrides]
"commdlg" = "builtin, native"
"comdlg32" = "builtin, native"
"ver" = "builtin, native"
"version" = "builtin, native"
"shell" = "builtin, native"
"shell32" = "builtin, native"
"shfolder" = "builtin, native"
"shlwapi" = "builtin, native"
"shdocvw" = "builtin, native"
"lzexpand" = "builtin, native"
"lz32" = "builtin, native"
"comctl32" = "builtin, native"
"commctrl" = "builtin, native"
"advapi32" = "builtin, native"
"cryptdll" = "builtin, native"
"mpr" = "builtin, native".....
```

Various Ports and devices also can be configured in the "[serialports]" section:

```
Com1=/dev/ttyS0
Com2=/dev/ttyS1
Com3=/dev/modem,38400
Com4=/dev/modem
```

The general appearance of the windows can be changed in the "[Tweak.Layout]" section.

```
;; supported styles are
'Win31'(default), 'Win95', 'Win98'
;; this has *nothing* to do
with the windows version Wine returns:
;; use cmdline option
--winver if you want that.
"WineLook" = "Win98"
```

## REMEMBER THE WINDOWS REGISTRY ?

Next, we need to install a default registry which will exactly match the way registry exists on a Windows Box. But before you do this we need to make minor changes to "/etc/ld.so.conf". We'll add a line

"/usr/local/lib/wine", which relates to all the libraries used by the software to mimic a Windows atmosphere. Don't forget to run "/sbin/ldconfig" after this step.

Next, we will use regapi to install a default registry. From the Wine source directory issue the following command:

```
programs/regapi/regapi setValue < winedefault.reg
```

## LET'S FIRE IT UP!!

Without waiting any further let's try our hands on our Wine installation to run a simple Windows App. We will try to run the standard Calculator which comes with Windows ("calc.exe").

You can mount your windows partition or copy the file "calc.exe" with a floppy to your system in the folder "/c/windows" and use any one of the ways to start it up:

```
cd /c/windows; wine calc.exe
wine
/c/windows/calc.exe
wine "c:\\windows\\calc.exe"
```

This is the way it appears on my Linux box. Pretty amazing isn't it!

## BYE BYE FOR NOW!

Wine is pretty indispensable when you have to run Windows executables on your Linux box but one thing to note here is that not all of your Applications will work on Wine, you will have to figure out ways and tweaks to make your favorite App. work fine. But for many cases Wine proves to be of a great help.

**Gaurav Taneja works as a Technical Consultant in New Delhi, India in Linux/Java/XML/C++. He is actively involved in open-source projects, with some hosted on SourceForge. His favorite leisure activities include long drives, tennis, watching movies and partying. He also runs his own software consulting company named BroadStrike Technologies.**

*This article is re-printed with permission. The originals can be found at:*

<http://www.linuxgazette.com/issue80/taneja.html>

# Barrel Patrol 3D

Author: Harald Radke <harryrat(at)gmx.de>

OpenGL  
a 3D graphics adapter (recommended)

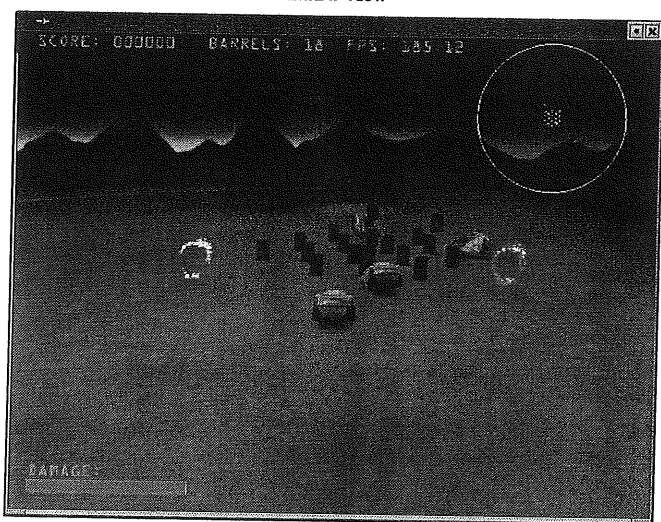
## ABSTRACT:

Barrel Patrol 3D is a cute little 3d game, based on Ripoff, a classical arcade game from the early 80's.

## Introduction

For all of you who have never played Ripoff (just like me): your mission is to protect barrels from being stolen by enemy tanks. You yourself also control a tank equipped with a gun. Basically you have to shoot down those tanks in order to avoid your barrels being taken away while not being hit by the enemy tanks which are also armed. In this article version 0.90 of Barrel Patrol 3D will be reviewed.

Standard View



Standard View, you are the nearest one

## GETTING READY TO PLAY

Barrel Patrol 3D is available for several platforms, including of course Linux. You can get it from the Fathom Entertainment Website (<http://www.fathomgames.com/>). There is no source code package, just a zipped tar-ball with a precompiled binary and the data file. The archive file has a size of approx. 1 MB, so no big deal to download it even with a slow connection.

After having saved it onto your harddisk you can extract it with `tar -xzf <ARCHIVNAME>`, a new directory containing the game files will be created.

Simply change into this directory and enter `./barrel_patrol_3d`. Besides the binary and the data file, there is a README with some nodes.

## Requirements

X window system  
Simple DirectMedia Layer (SDL)

*Note: I had some problems to get the game running with SDL 1.2.0, basically it crashed. After upgrading to SDL 1.2.3. however it worked without a flaw*

## PLAYING THE GAME

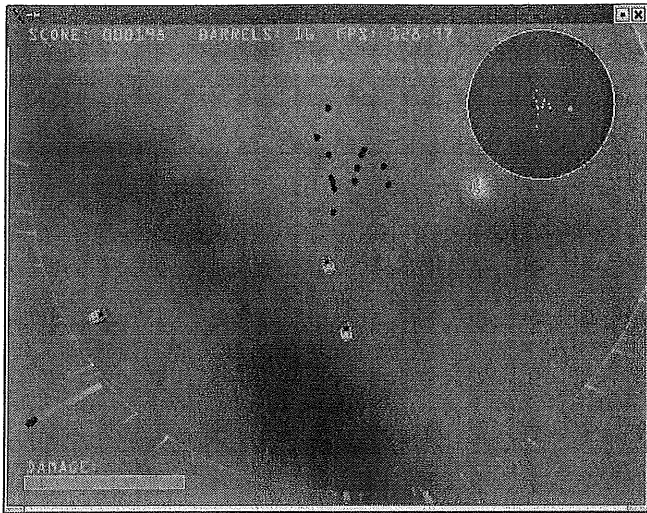
The game takes place inside a circular arena. The game field is bounded by a forcefield which you as the player cannot pass. You control a futuristic tank, equipped with a gun to shoot down those marauding enemy tanks which try to take away your barrels and carry them outside the arena. The enemy in turn can also shoot at you (and other tanks), so it's not only hunting them down but also trying to avoid being hit. From time to time add-ons appear and can be collected to improve firepower. However these add-ons count for *all* vehicles, so enemy tanks are also able to get better weapons. All add-ons are functional till the tank gets torn into pieces. Additionally they are cumulative.

The game is quite easy to play, you can accelerate your tank into the direction it currently heads to, slow down and turn around. The gun is mounted to fire always forward and there is no special support to point it at a target. Add-ons appear as different big letters and vanish after some time so be quick or they are gone, either after time has expired or, even worse, collected by the enemy.

The game consists of several levels, with a fixed number of barrels and enemies. After all enemies have been destroyed you proceed to the next level. After each level you get points for all barrels remaining inside the arena. From level to level the game becomes more difficult, the enemy aims better and remaining in one place from which you shoot at those tanks will surely result in being hit. One hit is enough to destroy an enemy tank. Your vehicle on the other hand is a little bit tougher and can take more damage. If you have been hit too often, your tank explodes. However, the only consequence of this is, that you lose all your add-ons and some kind of time penalty (a few seconds), in which you have to watch those other tanks moving around and maybe taking barrels away without you stopping them. There is no level restart or any "life" taken away every time your tank was shot into pieces. Collecting add-ons improves your firepower, for example your missiles fly faster, let them rebound off the forcefield or gives you homing missiles. Do not forget, same applies for the enemy! Besides the gameview itself, your actual scoring, the number of remaining barrels, a radar screen and a damage indicator are displayed.

The game ends if all of your barrels during one level have been stolen and taken outside the arena.





setting, isn't it?). Barrels being beamed into the arena at the beginning of each level and being beamed up after the enemy took them outside the forcefield is as well animated as the appearance of add-ons and exploding vehicles. Each action and event is accompanied with particular sound effects. The game comes with sound effects only, there is no music played in the background. The framerate which is displayed in the upper right corner is quite high on an Athlon Thunderbird 1.4 Ghz and a GForce 2 MX. Really, it is quite high, so even on older machines with an 3d graphics accelerator which is not state of the art, this game should run fast and smooth enough to have a lot of fun playing it.

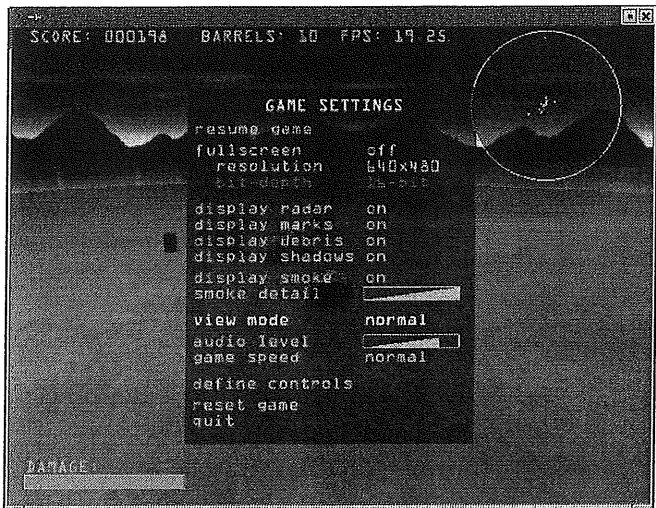
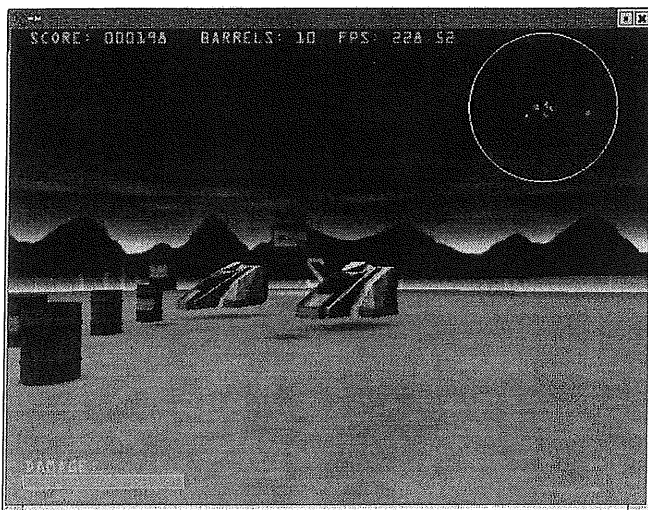
## CONCLUSIONS

Barrel Patrol 3D really is a nice little arcade game. It takes you maybe a minute to get into gameplay and fun will last for hours if you like this kind of games. Graphics are great, I think they are as good as those in *Descent 3*, the only thing that let you maybe disagree is the fact that there is less to see compared to that legendary 3D shooter.

The game is still being developed (hey, it is a Linux game!), more features are planned. Nevertheless it is fully playable. There are two things I missed so far about Barrel Patrol 3D: some atmospheric music played in the background and some kind of multiplayer mode. That would for sure make Barrel Patrol 3D one of the best Linux games.

## OPTIONS

Pressing *ESC* during the game will bring up an options menu where you can modify graphics and sound issues as well as changing the game view between the standard view where you can see your tank, view from inside the tank or from the top. Additionally you can toggle between window and full screen mode and change tank controls. By default you move your tank with the arrow keys and fire with *SPACE*. Graphics options include displaying debris, shadows and radar as well as the level of detail smoke is displayed.



## APPEARANCE

Though today games like *Return to Castle Wolfenstein* are state of the art and Barrel Patrol 3D can't stand a comparison with them, graphics are really neat and maybe one of the best inside the Linux gaming scene. The bounding forcefield is animated with moving light reflections, tanks and barrels are nicely modeled and textured. The background showing a dark mountain scene creates a really nice surrealistic atmosphere (hey, stealing barrels with a futuristic tank from a forcefield bounded arena is a quite surrealistic

## REFERENCES

- The website of Fathom Entertainment (<http://www.fathomgames.com/>)
- The Linux Game Tome entry (<http://www.happypenguin.org/show?Barrel%20Patrol%203D>) of Barrel Patrol 3D featuring several comments of linux gamers

*This article is re-printed with permission. The originals can be found at:*

<http://www.linuxfocus.org/English/March2002/article237.shtml>

# The Penguin and the Hare....

Author: Con Zymaris <[conz@cybersource.com.au](mailto:conz@cybersource.com.au)>

## ABSTRACT

In this analysis, we examine the similarities in the early-phase growth of Microsoft's Windows and Linux as desktop operating systems and show that Linux, far from being out of the mainstream desktop race, is moving smoothly, growing in stature as a performer and is starting to eye the finish line, which is just a few years' down the track, with confidence.

Throughout 2001, a substantial number of industry pundits took it upon themselves to deflate or denounce Linux's chances as a desktop Operating System contender. They pointed to the fact that even after two or three years of what they called 'hype', Linux still had a minuscule proportion of the mainstream desktop market. What are its chances of catching the naturally advantaged Windows platform on the desktop, and combating the arrogant and aggressive Microsoft?

Circa 600BC, the Greek, Aesop, wrote a collection of fables, which, while simple on the surface, proffer much hidden depth and a level of truism which sometimes only becomes apparent through the maturation of the memes which they deliver. One such fable, a very famous example, relates to a tortoise, a slow moving, measured creature, and the hare, all pace, flitty and somewhat arrogant. The story goes that when challenged to a race by the tortoise, the hare, with consummate ease, uses its natural advantages on the particular measure at hand (running) to tear ahead and enjoys a mammoth early lead. However, through the hare's arrogance and over-confidence, it is eventually defeated by the tortoise, fair and square, *\_in the long run\_*. I'd like to apply a variation of this simple parable to the desktop operating platform race that is being contested at present, between the seemingly plodding penguin and the tear-away hare.

To begin our analysis of the competition at hand, we need need a little history of the contestants.

Windows, as with many of Microsoft's technologies, was pre-announced by two years in 1983. This oft-used ploy has the effect of neutering any first-to-market competitive advantages bestowed on Microsoft's competitors, who have often come out with more original products. Windows was based largely on the concepts demonstrated by Apple's development groups, in turn re-working Xerox PARC, who were influenced by SRI's Doug Engelbart's ideas. After an invite for a site visit from Apple, Microsoft was able to glimpse this brand new future of Graphical User Interfaces, particularly embodied by what was phlegmatically code-named SAND by Microsoft,

(Steve's Amazing New Device,) the glorious Macintosh. Microsoft were shown this technology early on, as it was a significant player in the microcomputer software industry, whose application software support was eagerly sought by Apple, to help cement the availability of business apps for the fledgling PC-killer. This is indeed ironic when one considers this situation replicated in the present-day, specifically the leveraging power that Microsoft has over Apple through the existence (or non-existence) of Microsoft's Office product for the Macintosh. Regardless, Microsoft took Apple's GUI ideas, and as happened on numerous subsequent occasions (for example with their replication of Go Corporation's ideas on the first pen-tablet palm computers in the early 90's) copied them. The GUI was so extremely compelling that other firms, such as Digital Research, Inc., the then king of operating systems platforms and purveyor of CP/M & CP/M-86, (which Microsoft itself was to compete with when it purchased their clone, DOS, product from Seattle Computer Products), IBM, Quarterdeck and Geoworks, all came out with variations on desktop-metaphor interfaces for x86-based computers.

Windows 1.0 itself was released in mid-1985, to very little enthusiasm. In fact, Windows uptake was so underwhelming, that Microsoft had problems selling the Windows-based apps (like Excel) that it had ported across from the Mac. To overcome this embarrassing problem, Microsoft effectively bundled the OS with the application as a run-time environment, a reverse of what it does nowadays. This ploy wasn't particularly successful either. Most users kept using DOS-based products like Wordperfect and Quattro Pro. However, when the wave for Windows (as a desktop interface) did eventually break, Microsoft's efforts in making it's core applications available under Windows (because, let's face it, hardly anyone else wrote Windows apps) paid off handsomely. It effectively had built the right-shaped surfboard, and more importantly, had helped drum up the wave, which it has since ridden to absolute financial power and glory within the software industry. It's this maneuver which caught its competitors (Ashton Tate, Borland, Lotus, WordPerfect, Software Publishing Corp.) all off-guard. By the time these firms had released feasible versions of their marquee applications under Windows, Microsoft had entrenched its own file formats and application interfaces as standards. What little market share was available to these once powerful and monied software vendors, was snuffed-out when Microsoft decided to 'crowbar' the market penetration of its less-successful applications by leveraging the more successful ones, through the masterstroke of bundling them all into Microsoft Office. It could afford to make less money per application for a short period of time, as it could in effect rely on its massively lucrative PC 'tax', MS-DOS, through the then prevalent per-processor licence agreements with OEM hardware vendors, which would eventually catch the attention of the US Federal Trade Commission (FTC) in the mid 90s. Before anyone noticed, Microsoft had almost total control of all the major 100+ and 10+ million seller



applications: word processors, spreadsheets, databases, presentation programs etc.

So, we have a time spanning from a pre-history of Windows in 1983, through delivery in 1985, and the mid to late 80's where Windows was constantly in the mainstream IT press, heavily marketed but with minimal success. Only a minute portion of the hundred million PC users actually purchased Windows, and even fewer used it. It was not until the release of Windows 3.0 (1989), and probably, more succinctly, Windows 3.1 (1991) that a non-trivial portion of the great mainstream of computer users started to move across to using Windows as their mainstay desktop OS. Even then, most users kept relying heavily on DOS programs for core business requirements, and well into the late 90s for games. Finally, with Windows 95, released a decade after Microsoft's initial release, that Microsoft can be said to have 'attracted' the majority of desktop users to its Windows platform. We therefore have a 10+ year timespan of non-linear adoption, from initial availability through to substantial domination. It took exactly that long, with growth almost entirely happening in the last few years of this span. This, for a product that has the most expensive marketing and advertising campaign in industry history. How does Linux uptake compare?

KDE 1.0 came out in July 1998. It was soon followed by Gnome. Between them, KDE and Gnome are the first real attempts by the free software community to create a \_desktop\_ oriented towards the expectations of the great mainstream of computer users; namely by harking back to these users' knowledge of and experience with MacOS or Windows 95. It's the small things in operation, key-bindings, window focus that separate KDE and Gnome from previous windowing environments under Linux. Afterstep, Blackbox, FVWM (of various incarnations) were all more influenced by Unix and Unix-like workstation desktops, and are generally unlike what most users from the PC realm understood. While there is absolutely nothing wrong with these other windowing environments, they were obviously not going to be the mechanism through which Linux could win comfortable converts from the desktop PC world. Both KDE and Gnome started with functional yet uninspiring desktops, but now, after several major and myriad minor releases, are close to matching industry best-practice in GUI operating environments.

We are now in early 2002, and by our comparison to Microsoft Windows' time-line, we are where Windows 2.0 was in about late 1988 or about 4 years into the Windows path to ascendancy. Please note, this does \_not\_ mean to imply that Linux as it presently stands is being equated technically to Windows 2.0. We are equating market penetration. Back in 1988, Windows had to compete with both its x86 PC-based GUI brethren (GEM from DRI, Geoworks etc.) as well as its largest entrenched competitor, DOS, ironically, also from Microsoft. The fact that Windows eventually allowed for the seamless operation of the most important of users' DOS-based application was

crucial to the uptake and eventual success of Windows. Without this ability, many users might just have moved over to Desqview 386 or OS/2, which had arguable better DOS emulation functionality. As a side-note at this juncture, it should be obvious that if Linux were to allow for the seamless operation of the most important of users' Windows-based applications, it would greatly assist it in its race for desktop supremacy. WINE is thus of utmost importance to Linux. Regardless, it must be stated clearly and forcefully, that at this stage of its market penetration, Windows was considered a joke as a desktop operating platform. It had a minuscule following amongst the technology innovators; few of them took it as a serious contender in the space. These people didn't adopt Windows for another couple of years. And where these people lead, others, more often than not, follow.

Linux has perhaps the best shot at unseating the desktop OS incumbent for a number of reasons. Past contenders, such as the Apple Mac and IBM OS/2 had a number of inherent market shortcomings which hampered their penetration. In short, the MacOS could never become a great volume player due to its availability in a single hardware range, produced by a single supplier. OS/2 competed head-on with Windows in the early 90's and lost; partly through (in an industry where platform monopolies are the natural course of things) the adage of 'their can be only one' holds; partly because IBM was on the nose for many in the PC industry in a similar way that Microsoft is now; but mostly through the onerous per-processor licences encumbered upon PC vendors by Microsoft, making the activity of bundling any alternative OS with their hardware economically non-viable. Linux suffers from none of these hindrances, and what were actual restrictive issues for the adoption of Linux (installation complexity, lack of GUI polish, applications) have been methodically resolved, one by one. As things stand now, there are no valid technical or logical reasons for eschewing Linux; only politics and religion remain and these cannot withstand the ever-present pressures of cost-efficiency and competitiveness, demanded by business economics, for too long.

Perhaps the most important reason why Linux has the best chance at becoming the de-facto desktop standard in the medium-to-long term is this: there is no obsolescence with Linux. As long as there are users and a user community, there will be support and ongoing development. Contrast this to Microsoft's recent actions in earmarking the removal of support for both Windows 98 and Windows NT (their most popular OSes) over the coming year or two, either stranding hundreds of millions of users, or forcing costly, ongoing upgrades in a never-ending cycle.

Finally, there is another important factor which greatly enhances Linux's chances as a desktop platform, and makes it far more attractive than previous contenders. Price. Never underestimate the immediate attraction of 'free beer'.

# Cracking Open Proprietary Envelopes

Adrian J. Chung <[ajchung@email.com](mailto:ajchung@email.com)>

This combination of positive attributes is dawning on the industry. Up until a few months ago, and perhaps for almost a year now, we were seeing an increasing number of industry pundits decry Linux's role or position on the desktop. There were claims that it's assault on the desktop was stillborn. In recent months, this Cassandra-like chorus of doom has been subsiding. This may be partly through the fact that Linux hasn't, in reality, been washed away with the dot-bomb crowd, which the pundits were expecting. In fact, as each month goes by, Linux on the desktop is starting to draw the kinds of grudging respect that was wrung from the pens of the industry nay-sayers about Linux as a network infrastructure platform 4 years ago, and Linux as an embedded and real-time OS 2 years ago. It's very likely that this pattern of acceptance is coalescing in the minds of the pundits. Every time they've thrown rocks at Linux, they've regretted it. They are learning not to underestimate the staying-power of the penguin.

Where is Linux at present? Is it as far along with market penetration on the desktop as Windows was 4 years into its push? Does it have as much market recognition as Windows did at the equivalent time? It is my strong belief that the answer to both these questions is a resounding 'Yes!'. As someone who was in the IT industry both then and now, I can tell you that the recognition among both IT professionals and average users of Linux is far higher than that of Windows in 1988. Cite the Internet as the ultimate in guerrilla-marketing tools; cite the phenomenal evangelistic efforts undertaken by Linux enthusiasts worldwide; cite the growing disenchantment with Windows specifically and Microsoft generally, it makes no difference. Linux has achieved far more 'brand' and aura of quality and value in its 4 years of desktop ascendance than Microsoft's Windows had in the same juncture, regardless of the gold-lined coal that Microsoft shovelled into the marketing and advertising grist-mill. Importantly, Linux reigns near-supreme amongst many of the technology-innovators, now.

Based on the time-frame example set by Windows' own march to domination, we can see that the race between the penguin and the hare for the desktop OS blue ribbon has only just begun, and we should settle into a few more years of both contestants running the course. We have hopefully shown, however, that Linux, far from having run its race against Windows on the desktop, is analogous to the the tortoise, and is conscientiously keeping pace, moving slowly at first, but inexorably forwards, towards mainstream acceptance and perhaps dominance of the desktop market.

*This article is re-printed with permission. The originals can be found at:*

<http://www.cyber.com.au/users/conz/>

Anyone with an email address can expect to receive attachments in a multitude of formats. Unfortunately, some formats cannot be read using free software. This is especially true if our email buddies are still involved in the arguably risky practice of using proprietary programs in conjunction with their email readers.

Many free software advocates adopt a policy of ignoring all email with attachments dependent on closed source software, opting instead to lecture the sender on the importance of open standards. Others may not like missing out on the fun to be had from attachments being forwarded amongst their peers. If you find yourself in this situation, the techniques outlined in this article may serve as a partial solution. There is not much a Linux user can do if the entire contents of an attachment are encoded using a jealously guarded secret algorithm. Very often however, the problematic file is merely a thin proprietary envelope enclosing a loose collection of data objects that use well-known encoding standards. For instance, some MS Word documents being forwarded around the Net contain ordinary JPG and PNG images embedded within the file. If we can find a way to remove the envelope, reading these enclosed files would be a straight forward matter. The following sections describe how this can be accomplished using a little Python scripting together with a few image viewing and manipulation tools available on most Linux distributions.

## EXTRACTING THE TEXT

Before tackling the problem of the embedded images we can easily view any readable text using the *strings* utility:

```
strings proprietary.file | less
```

This will output any strings of at least 4 bytes in length that consist of readable ASCII characters. Naturally, a lot more than just intelligible sentences will be returned. Most will be junk, but the readable text is easily spotted. The *strings* tools will also pick up the readable header information within the embedded images themselves. JPEG files contain the string "JFIF" in the header. This gives us a quick way to check what types of images a file may contain, and gives an indication of how many there are.

```
strings proprietary.file | grep JFIF
strings -n 3 proprietary.file | grep PNG
strings proprietary.file | grep GIF8
```

The `-n 3` allows us to detect readable strings as short as 3 characters. Not every occurrence of "JFIF" is necessarily a JPEG image since the document itself may have mentioned JFIF in a paragraph of text --

though this is rare among the email attachments most commonly forwarded.

## LOCATING THE IMAGES

We need to find where exactly each image is located within the file. A little Python will help to find possible embedded images and report their positions as a byte offset:

```
from string import find

#read in proprietary data
fh = open( "proprietary.file" )
dat = fh.read()
fh.close()

#search for JFIF
x = -1
while 1:
    x = find(dat,"JFIF",x+1)
    if x<0: break
    #file actually started 6 bytes earlier
    print x - 6
```

This will find the byte offsets of every embedded JPEG file though not every offset is guaranteed to be for a valid file. This can easily be extended to handle GIF and PNG images:

### Listing 2

```
#!/usr/bin/python
from string import find
from sys import argv

headers = [("GIF8",0), ("PNG",1), ("JFIF",6)]
filepath = "proprietary.file"
if len(argv)>1: filepath = argv[1]

fh = open(filepath )
dat = fh.read()
fh.close()

for kw,off in headers:
    x = 0
    while 1:
        x = find(dat,kw,x+1)
        if x<0: break
        print kw,"file begins at byte",x - off
```

Note that the image file begins a few bytes before the "PNG" or "JFIF" string.

## DISPLAYING THE IMAGES

Now that we know where each image is likely to start how do we display them? ImageMagick's *display* utility can help here. Suppose our proprietary file contains a JPEG image beginning at byte 1000. Using *tail* to remove all the bytes that precede it and pipe the rest to *display*.

```
tail -c +1001 proprietary.file | display -
```

Note that *tail -c* begins counting bytes at 1. In case we have many dozens of embedded image files we can adapt our previous Python script to automate the process.

### Listing 2

```
#!/usr/bin/python
from string import find
from sys import argv
from os import system

headers = [("GIF8",0), ("PNG",1), ("JFIF",6)]
filepath = "proprietary.file"
if len(argv)>1: filepath = argv[1]

fh = open(filepath )
dat = fh.read()
fh.close()

for kw,off in headers:
    x = 0
    while 1:
        x = find(dat,kw,x+1)
        if x<0: break
        system("tail -c +%d %s | display -" % (x -
off + 1, filepath))
```

## EXTRACTING EACH IMAGE FILE

ImageMagick throws away any excess data fed to it after reading to the end of the image segment. If we want to separate the image data completely for storage as individual files, we also need to find the end of each image. One way to do this is to use a modified binary chop algorithm.

### Listing 4

```
#!/usr/bin/python
from string import find
from sys import argv
from commands import getstatusoutput

headers = [("GIF8",0,"giftopnm","gif"),
("PNG",1,"pngtopnm","png"),
("JFIF",6,"djpeg","jpg")]
filepath = "proprietary.file"
if len(argv)>1: filepath = argv[1]

fh = open(filepath )
dat = fh.read()
fh.close()

inum = 0
for kw,off,conv,ext in headers:
    x = -1
    while 1:
        x = find(dat,kw,x+1)
        if x<0: break
        beg = x - off
        #possible image located -- find end by
        binary chop
        s1 = len(dat) - x
        s0 = 1
        sz = s1
        while s0<s1:
            (stat,output) = getstatusoutput("tail
-c +%d %s | head -c %d | %s >/dev/null" % (beg + 1,
filepath, sz, conv))
            if stat:
                #failed -- possibly too small
                if sz == s1:
                    #failed -- probably invalid
                    data
                    print "failed... no image here"
                    break
                elif sz == s0:
                    #we've found the length --
                    write out image
                    imgname = "image%03d.%s" %
(inum, ext)
                    print "writing",imgname
```

```

fh = open( imgname, "w")
fh.write(dat[beg :beg+s1])
fh.close()
inum = inum + 1
break
s0 = sz
else:
#might be too big -- try smaller
s1 = sz
sz = int((s0+s1)/2)

```

One can make use of image decoding utilities *giftopnm*, *djpeg*, and *pngtopnm* to locate the end of the file. Like *display* these tools discard excess input data after the end of the image file and with terminate without error. If however they are given truncated image data they will report an error and terminate unsuccessfully. The Python script feeds image data of varying lengths to the decoding tool and its completion status is used to home into the correct length of the required file.

## CONCLUSION

This article has shown how to write scripts that extract data objects, encoded using platform-independent open standards, from within proprietary files. It should be a simple task to extend these scripts for handling other image formats and even other types of data objects, such as sound and music files. Note that there are many file formats that frustrate the techniques described here via a layer of simple encryption and/or obfuscation.

Even if one has access to the appropriate proprietary application for reading a particular email attachment, the scripts outlined above can be useful for avoiding any possible macro viruses or security exploits specific to that application.

And finally a word of warning. The legislature of some countries have vaguely worded laws that can be interpreted in such a way that these scripts may be considered as illegal copyright circumvention devices. This may or may not be relevant to you depending on the country where you reside. As is always the case when mixing open and closed source systems, your mileage may vary.

*[Editor's note: The [Python Imaging Library \(PIL\)](http://www.pythonware.com/products/pil), <http://www.pythonware.com/products/pil> provides a way to work with images from within a larger program. You can open an image and read its type and dimensions, transform it, create thumbnails, etc. -Iron.]*

*When not teaching undergraduate computing at the University of the West Indies, Trinidad, Adrian is writing system level scripts to manage a network of Linux boxes, and conducts experiments with interfacing various scripting environments with home-brew computer graphics renderers and data visualization libraries*

*This article is re-printed with permission. The originals can be found at:*

<http://www.linuxgazette.com/issue79/chung.html>

# Setting up a Squid-Proxy Server

D.S. Oberoi <[ds\\_oberoi@yahoo.com](mailto:ds_oberoi@yahoo.com)>

## ABSTRACT

Linux has become a synonym for Networking. It is being used both in office and home environments as file, print, e-mail, application server and also it is increasingly being used as Proxy server.

A proxy-server provides Internet access to different users at same time i.e. by sharing a single Internet connection. A good proxy server also provides for caching of the requests, which helps to access data from local resources rather fetching the data from web thus reducing access time and bandwidth. Squid is one such software which supports proxy, caching of HTTP, ftp, gopher, etc.. It also supports SSL, access controls, caching of DNS and maintains a full log of all the requests. Squid is as well available for Windows-NT from Logi Sense.

The focus of this article is to give basic guidelines of setting up a proxy server and ways of providing controlled access to users.

## IS SQUID INSTALLED ?

Squid's rpm comes bundled with the RedHat 7.1 and is installed automatically with the Network OS installation option. One can check whether it is installed or not with the following rpm command:

```
rpm -q squid
```

The latest version of Squid can always be obtained from the Squid Homepage and other mirror sites. Squid can be installed on the desired system by using the following rpm command:

```
rpm -ivh squid-2.3.STABLE4-10.i386.rpm
```

## CONFIGURING SQUID

The working and behavior of the Squid is controlled by the configuration details given in it's configuration file i.e. *squid.conf*; this file is usually found in directory the */etc/squid*. The configuration file *squid.conf* is a mile long affair, it just keeps on going for pages after pages, but the good point is that it has all options listed out clearly with explanation.

The first thing that has to be edited is the *http\_port*, which specifies the socket address where the Squid will listen to the client's request; by default this is set to 3128, but can be changed to a user defined value also. Along with the port value, one can also give the IP address of the machine on which Squid is running ; this can be changed to:

```
http_port 192.168.0.1:8080
```

With above declaration Squid is bounded to the IP address of 192.168.0.1 and port address of 8080. Any port address can be given; but make sure that no other application is running at set port value. With similar configuration lines other service's request ports can also be set.

## ACCESS CONTROL

Through access control features the access to Internet can be controlled in terms of access during particular time interval, caching, access to particular or group of sites, etc.. Squid access control has two different components i.e. ACL elements and access list. An access list infact allows or deny the access to the service.

A few important type of ACL elements are listed below

- **src** : Source i.e. client's IP addresses
- **dst** : Destination i.e. server's IP addresses
- **srcdomain** : Source i.e. client's domain name
- **dstdomain** : Destination i.e. server's domain name
- **time** : Time of day and day of week
- **url\_regex** : URL regular expression pattern matching
- **urlpath\_regex**: URL-path regular expression pattern matching, leaves out the protocol and hostname
- **proxy\_auth** : User authentication through external processes
- **maxconn** : Maximum number of connections limit from a single client IP address

To apply the controls, one has to first define set of ACL and then apply rules on them. The format of an ACL statement is

```
acl acl_element_name type_of_acl_element
values_to_acl
```

Note :

1. **acl\_element\_name** can be any user defined name given to an ACL element.
2. No two ACL elements can have the same name.
3. Each ACL consists of list of values. When checking for a match, the multiple values use OR logic. In other words, an ACL element is matched when any one of its values matches.
4. Not all of the ACL elements can be used with all types of access lists.
5. Different ACL elements are given on different lines and Squid combines them together into one list.

A number of different access lists are available. The ones which we are going to use here are listed below

- **http\_access**: Allows HTTP clients to access the HTTP port. This is the primary access control list.
- **no\_cache**: Defines the caching of request's responses

An access list rule consists of keywords like allow or deny ; which allows or denies the service to a particular ACL element or to a group of them.

Note:

1. The rules are checked in the order in which they are written and it terminates as soon as rule is matched.
2. An access list can consists of multiple rules.
3. If none of the rules is matched, then the default action is opposite to the last rule in the list; thus it is good to be explicit with the default action.
4. All elements of an access entry are AND'ed together and executed in following manner  

```
http_access Action statement1 AND
statement2 AND statement3
http_access Action statement3
```

Multiple **http\_access** statements are OR'ed whereas elements of an access entry are AND'ed together
5. Do remember that rules are always read from top to bottom.

## BACK TO CONFIGURATION

By default, Squid will not give any access to clients and access controls have to modified for this purpose. One has to list out one's own rules to allow the access. Scroll down in the squid.conf and enter the following lines just above the **http\_access deny all** line

```
acl mynetwork 192.168.0.1/255.255.255.0
http_access allow mynetwork
```

**mynetwork** is the acl name and the next line is the rule applicable to a particular acl i.e. **mynetwork**. 192.168.0.1 refers to the address of the network whose netmask is 255.255.255.0.. **mynetwork** basically gives a name to group of machines in the network and the following rule allows the access to clients. The above changes along with **http\_port** is good enough to put Squid into gear. After the changes Squid can be started by the following command

```
service squid start
```

Note :

Squid can also be started automatically at boot time by enabling it in ntsysv or setup (System Service Menu). After each and every change in the configuration file, the present Squid process has to be stopped and for new configuration changes to take effect, Squid has to be started once again. These two steps can be achieved by following commands

1. `service squid restart`
2. `/etc/rc.d/init.d/squid restart`

## CLIENT MACHINE CONFIGURATION

Since the client request will be placed at a particular port of the proxy server, client machine's have to be configured for the same purpose. It is taken at this point that these machines are already connected to LAN ( with valid IP address) and are able to ping the



Linux sever.

For Internet Explorer

1. Go to Tools -> Internet Options
2. Select Connection Tab and click LAN Setting
3. Check Proxy Server box and enter IP address of proxy server and port address where request are being handled (http\_port address).

For Netscape Navigator

1. Go to Edit -> Preference -> Advanced -> Proxies.
2. Select Manual Proxy Configuration radio button.
3. Click on View Button &
4. Enter enter IP address of proxy server and port address where request are being handled (http\_port address).

## USING ACCESS CONTROL

Multiple Access controls and rules offer a very good and flexible way of controlling client's access to Internet. Examples of most commonly used control are given below; this by no means should be taken as the only controls available.

1. Allowing selected machines to have access to the Internet

```
acl allowed_clients src 192.168.0.10
192.168.0.20 192.168.0.30
http_access allow allowed_clients
http_access deny !allowed_clients
```

This allows only machine whose IPs are 192.168.0.10, 192.168.0.20 and 192.168.0.30 to have access to Internet and the rest of IP addresses (not listed) are denied the service.

2. Restrict the access during particular duration only

```
acl allowed_clients src
192.168.0.1/255.255.255.0
acl regular_days time MTWHF 10:00-16:00
http_access allow allowed_clients
regular_days
http_access deny allowed_clients
```

This allows the access to all the clients in network 192.168.0.1 to access the net from Monday to Friday from 10:00am to 4:00 pm.

3. Multipletime access to different clients

```
acl hosts1 src 192.168.0.10
acl hosts2 src 192.168.0.20
acl hosts3 src 192.168.0.30
acl morning time 10:00-13:00
acl lunch time 13:30-14:30
acl evening time 15:00-18:00
http_access allow host1 morning
http_access allow host1 evening
http_access allow host2 lunch
http_access allow host3 evening
http_access deny all
```

The above rule will allow host1 access during both morning as well as evening hours; where

as host2 and host3 will be allowed access only during lunch and evening hours respectively.

Note:

All elements of an access entry are AND'ed together and executed in following manner

```
http_access Action statement1 AND staement2
AND statement OR.
```

multiple http\_access statements are OR'ed whereas elements of an access entries are AND'ed together; due to this reason the

```
http_access allow host1 morning evening
```

would have never worked as time morning and evening (morning AND evening) would never ever be TRUE and hence no action would have taken place.

4. Blocking sites

Squid can prevent the access to a particular site or to sites which contain a particular word. This can be implemented in the following way

```
acl allowed_clients src
192.168.0.1/255.255.255.0
acl banned_sites url_regex abc.com
*()(*.com
http_access deny banned_sites
http_access allow allowed_clients
```

The same can also be used to prevent access to sites containing a particular word i.e. dummy, fake

```
acl allowed_clients src
192.168.0.1/255.255.255.0
acl banned_sites url_regex dummy fake
http_access deny banned_sites
http_access allow allowed_machines
```

It is not practical to list all the words list or sites names to whom the access is to be prevented; these can be listed out in the file (say banned.list in /etc directory) and ACL can pick up this information from this file and prevent the access to the banned sites.

```
acl allowed_clients src
192.168.0.1/255.255.255.0
acl banned_sites url_regex
"/etc/banned.list"
http_access deny banned_sites
http_access allow allowed_clients
```

5. To optimize the use

Squid can limit number the of connections from the client machine and this is possible through the maxconn element. To use this option, client\_db feature should be enabled first.

```
acl mynetwork 192.168.0.1/255.255.255.0
acl numconn maxconn 5
http_access deny mynetwork numconn
```

Note:

maxconn ACL uses less-than comparison.

This ACL is matched when the number of connections is greater than the specified value. This is the main reason for which this ACL is not used with the http\_access allow rule.

#### 6. Caching the data

Response of the request are cached immediately, this is quite good for static pages. There is no need to cache cgi-bin or Servlet and this can be prevented by using the no\_cache ACL element.

```
acl cache_prevent1 url_regex cgi-bin /?
acl cache_prevent2 url_regex Servlet
no_cache deny cache_prevent1
no_cache deny cache_prevent2
```

#### 7. Creating Your Own Error Messages

It is possible to create your own error message with a deny rule and this is possible with the deny\_info option. All the Squid error messages by default are placed in the /etc/squid/errors directory. The error directory can be configured with the error\_directory option. You can even customize the existing error messages.

```
acl allowed_clients src
192.168.0.1/255.255.255.0
acl banned_sites url_regex abc.com
*()(*.com
http_access deny banned_sites
deny_info ERR_BANNED_SITE banned_sites
http_access allow allowed_clients
```

In the above example, a special message will be displayed when ever users try to access the sites with above banned words. The file name in the option i.e. ERR\_BANNED\_SITE must exist in the above error directory. This error message file should be in HTML format. The above listed out examples are just a few of the options, facilities and capabilities of ACL. One can read through the [FAQ section](#) at the Squid Home Page for more extensive usage and explanation of other ACL elements and access elements.

## LOG FILES

All log files of Squid are contained in directory /var/log/squid; these contain cache log, access logs and store.log. File access.log maintains the information about the clients request, activity and maintains entry for each HTTP & ICP queries received by the proxy server, clients IP, request method, requested URL, etc.. The data of this file can be used to analyze the access information. Many programs like [sarg](#), [calamaris](#), [Squid-Log-Analyzer](#) are available which can analyze this data and generate reports (in HTML format). The reports can be generated in terms of users, IP numbers, site visited, etc..

The destination of these log files can also be changed by following options

```
cache_access_log For access.log
```

```
cache_log For cache.log
cache_store_log For store.log (Store manager)
pid_filename Squid process ID file name
```

## AUTHENTICATION METHODS

Squid in the default configuration allows any user to have access without any authentication process. To authenticate the users i.e. to allow only valid users (from any machine in the network) to access the Internet, Squid provides for authentication process but via an external program, for this a valid username and password is required. This is achieved by using proxy\_auth ACL and authenticate\_program; which forces a user to verify the username and password before the access is given. Several authentication programs are available which Squid can use and these are

1. LDAP : Uses Linux Lightweight Directory Access Protocol
2. NCSA : Uses NCSA style username and password file
3. SMB : Uses SMB server like SAMBA or Windows NT
4. MSNT : Uses Windows NT authentication domain
5. PAM : Uses Linux Pluggable Authentication Modules
6. getpwnam : Uses Linux password file.

One needs to specify the authentication program being used and this can be specified by using the authenticate\_program option. Make sure that the authentication program being used for the purpose is installed and working.

The changes in the squid.conf file now should also reflect the same authenticate\_program

```
/usr/local/bin/pam_auth
acl pass proxy_auth REQUIRED
acl mynetwork src 192.168.0.1/255.255.255.0
http_access deny !mynetwork
http_access allow pass
http_access deny all
```

This uses the PAM authentication program and all users need to authenticate before accessing the Internet.

Options like authenticate\_ttl and authenticate\_ip\_ttl can also be used to change the behavior of the authentication process i.e. revalidation of username and password.

## REFERENCES

This article just touches the tip of the Squid iceberg; for further reference visit the following Web sites

- Squid Home, [www.squid-cache.org](http://www.squid-cache.org)
- Squid Documentation Project, [squid-docs.sourceforge.net](http://squid-docs.sourceforge.net)
- [visolve.com](http://visolve.com)
- For Proxy Authentication,

*This article is re-printed with permission. The originals can be found at:*

<http://www.linuxfocus.org/English/March2002/article235.shtml>

# Building The Lo-Fat Linux Desktop

John Murray <[pursang@netwit.net.au](mailto:pursang@netwit.net.au)>

## INTRODUCTION

I first started playing with Linux a few years ago, after reading several Introduction-To-Linux type articles in computer magazines and on the web. In almost all these articles, low hardware requirements are listed as one of Linux's advantages. Usually the authors then go on to show how easy it is to use Linux on the desktop with the Gnome or KDE desktop environments.

So I set up my machine to dual-boot Win95 and Linux, and experimented with several different distros. Initially I was disappointed with the performance of Linux, and it took me a while to discover the performance gains made possible by running leaner software. The fact that most of the newbie-oriented documentation emphasised Gnome/KDE while ignoring everything else only made things harder. That's what this page is all about - a newbie's guide to good, lightweight software that runs well on boxes that are less than state-of-the-art. While a lot of us simply can't afford (or justify) the cost of current hardware, Windows 2000/XP's high hardware requirements could be a blessing for Linux users on a tight budget; there should be more secondhand machines becoming available as Windows users upgrade their hardware.

Gnome and KDE are good-looking, feature-packed environments that are as easy to use as the desktop on that other OS, but they aren't the best choice for an older machine. Later versions especially can actually be quite sluggish unless you have some fairly recent hardware to run them. That doesn't mean you're stuck with a text-only console though, as it's easy to set up a nice looking Linux desktop that has plenty of speed on something like an early Pentium with 32megs of RAM. And with RAM being so cheap at the moment, I'd go for 64megs if you can afford it.

So a speedy desktop is largely just a matter of using a window manager and applications that suits your hardware. And by the way, just because you don't use the KDE or Gnome desktop environments doesn't mean you shouldn't install them. KDE and Gnome apps will run quite well under a lightweight window

manager, so if you have the disk space, I recommend installing both. Listed below are some suggestions for the type of apps. that most people use everyday, all of which work nicely on my 233/64 box (and most of this stuff should be fine with just 32megs of RAM). Keep in mind that these suggestions are only my own personal preferences; they certainly aren't the only way to do things.

## THE SELECTION CRITERIA:

- Performance - It should be acceptably fast and stable on older hardware
- Graphical Interface - most newbies and non-geeks prefer this to the command line
- Functionality - It should do everything that normal users (whatever they are) expect of that type of app.
- Ease of Installation - It should be reasonably simple to install, without needing kernel recompilation and without too many obscure dependencies.
- Ease of Configuration - You shouldn't need to be a vi or scripting guru to knock it into shape
- Ease of Use - It should be reasonably easy to learn the usage.

The ease of use bit was simple to test - my wife and kids share my computer but are definitely not geeks. If they were able to use a newly installed program without swearing at it or calling for assistance it was deemed to have passed the ease-of-use test :-)

## WHERE TO GET PACKAGES

You'll find a lot of this stuff is included on the installation cd's of most distro's, or you can follow the links. Wherever possible, these point to the project's homepage, or else to rpmfind's download site. If you're using something other than a RedHat style distro, you may have to backtrack a bit from the rpmfind sites to get the right version.

## THE WINDOW MANAGER

There are several good, lightweight window managers available, my favourite being IceWm (<http://www.icewm.org/>). As well as having a small memory footprint, IceWm can be made to look quite good with wallpapers and themes (<http://www.icewm.org/themes/>). It also has that familiar Win95 layout with the corner start button, menus, toolbar and so on.

Configuring IceWm is extremely easy, and while there are graphical tools available for this, it's just as easy to edit its configuration files manually. The global configuration files are usually in /usr/X11R6/lib/X11/icewm/ and are named preferences, menu and toolbar. Make a hidden folder called .icewm in your home directory and copy these three files into it. Then it's just a matter of editing them to suit your own needs and tastes.



IceWm is included with many recent distros, and includes very good documentation in /usr/doc/icewm.

Another lightweight WM that is very popular is xfce (<http://www.xfce.org/>), an exceptionally good looking and fast window manager that is worth a look.

## THE FILE MANAGER

Of the file managers I have tried I prefer XWC (<http://rpmfind.net/linux/RPM/mandrake/8.1/contrib/RPMS/xwc-0.91.4patch1-10mdk.i586.html>) (X Windows Commander) because of its speed and again for its familiar interface. XWC is a clone of the Win95 style Explorer that supports drag'n'drop and file associations etc. Although it lacks many of the features of say, Nautilus or Konqueror, its got everything I need, without the bloat. Like IceWm, it is very easy to configure using the built in options menu or by editing the ~/.foxrc/XWC file. While I'd prefer something that doesn't look quite so Windows-like, XWC works very well and is pretty speedy. One thing to watch out for is the fact that XWC will always open at the last location it was used. If you last used XWC to browse a removable media (like /mnt/cdrom for example), and you are using supermount, there can be a delay starting XWC if there is no device currently mounted. XWC requires the fox (<http://www.ifg.uni-kiel.de/doc-clients/libfox0.99/html/fox.html>) libraries.

It appears XWC is no longer actively maintained, and is only available in RPM format. Its successor, foXcommander (<http://sourceforge.net/projects/foxdesktop>), is similar and is part of the foXdesktop project. It is available as source.

Another fast, good looking filer that is highly recommended is rox (<http://rox.sourceforge.net/>)

## TEXT EDITORS

While XWC comes with its own basic editor, I much prefer Nedit (<http://rpmfind.net/linux/RPM/cooker/cooker/i586/Mandrake/RPMS/nedit-5.2-1mdk.i586.html>). Nedit is fairly small, fast and has lot's of useful features built in, including: syntax highlighting, search and replace, macro support, shell access and much more. The built in help is very good as well. I know some people get passionate about their editors ( especially the vi crowd ), but if you want a good WYSIWYG style editor, Nedit is very nice indeed.

Internet Stuff Manually configuring PPP is a pain, especially compared to kppp. Setting up kppp can be done in seconds, and this app. alone makes installing KDE worthwhile.

Hopefully, Linux users will soon have browsers that beat the performance of those on other platforms. In the meantime, Netscape 4.7x is probably the best all-

round graphical browser for use on a 32meg machine. While it can be a bit wobbly at times, it handles java/javascript reasonably well, and also works with the more common plug-ins. You'll need to click on Edit > Preferences to play with the font settings (and set your fonts to override the document-specified font) to make it look good. If you have 64meg or more, you might want to try Mozilla or one of its descendants (Galeon seems popular). These sometimes have more features and are more stable than Netscape 4.7x, but are probably no faster. Don't let the vomitous Netscape 6.0 put you off trying later versions like 6.1 or 6.2 that are generally very good, stable browsers. Lots of people like opera, though its interface takes some getting used to.

There is also a browser called Dillo that is worth installing. Dillo (<http://dillo.sourceforge.net/>) is extremely fast, and quite good looking as well. Still under development, it doesn't yet handle frames, java or javascript, so you probably won't be able to do your online banking with it. It's brilliant for reading local html files (like helpfiles and /usr/doc/\*html stuff). I use Netscape for internet work, and Dillo for local files.

Anyone know of a good HTML tool? I'm actually writing this in Netscape Composer instead of the usual Nedit, and while it's certainly easy, it's also making the the most god-awful HTML I've ever seen...

As for email, Netscape and Mozilla both have reasonable email clients built in, though it's a pain waiting for them to load just to read your email. A lot of people recommended Sylpheed, and it is now what I use. Sylpheed (<http://sylpheed.good-day.net/>) is very fast, and has a nice clear interface. It is also a basic newsreader. Netscape 4.7x's newsreader is pretty ordinary, so you might want to try Pan (<http://pan.rebelbase.com/>), a Gnome news app. capable of handling binary attachments.

Another useful utility is tnef (<http://users.netwit.net.au/~pursang/tnef.html>). It was designed to unpack those annoying "ms-tnef" MIME attachments that are commonly sent from Outlook and Exchange mail servers. Although it's a command line tool, it's easy to use and works well.

I know there are several graphical ftp clients, and I did play briefly with gFTP (<http://gftp.seul.org/>) which ran fine), but I can't really recommend anything else as I still prefer the command-line ncftp.

## GRAPHICS APPS

I use xli (formerly xloadimage) as my default image viewer. It's quick, and I like the way I can directly scroll big images with the mouse, though ee (Electric Eyes) is nice as well. Both ee and xv allow browsing through thumbnails of images, as well as simple manipulations. While the GIMP couldn't really be described as lightweight, its feature set make it a must on any Linux desktop, and it runs OK on a 32meg box.

## MUSIC AND VIDEO

XMMS (<http://www.xmms.org/>) is a very popular WinAmp clone that can play mp3,wav and cdr files etc. It also supports skins, including WinAmp skins. As for video mpegs, I use mtvp as the default player. It's a free player that's part of the mtv (<http://www.mpegtv.com/>) package and works very well on lower end machines. Xanim (<http://xanim.va.pubnix.com/home.html>) plays .mov and .avi files, among other things, but isn't very good at mpegs. And if you are reading this, you probably don't have enough computing horsepower to play DVD's. Lots of people have recommended Mplayer (<http://www.mplayerhq.hu/>) to me, and it really is an impressive piece of work. It plays many different formats well, and is quite quick. The only disadvantage is that it must be compiled from source, and this might discourage some newbies from trying it, though on my box at least, it built easily.

There are also plenty of graphical front ends around for cd recording software. I have played around with the very popular xcdroast (<http://www.xcdroast.org/>), but mainly I still use command line tools like cdrecord, mpg123, bladeenc etc. Again, let me know if you have recommendations.

## OFFICE TYPE STUFF

**Word Processing** -There are plenty to choose from here. If all you need is a basic word processor, go with AbiWord (<http://www.abisource.com/>). While it can import simple .doc files OK, it is limited to producing basic documents that don't contain tables etc. Despite the limitations, AbiWord is a fast and useful program. Kword (<http://www.koffice.org/kword>) is the KDE project's word processor, and it looks and works very well, however it has limited compatibility with MS .doc files at present. I use ApplixWords, see the section on Office Suites for more. And Corel's WordPerfect seems to have disappeared from the face of the earth...

**Spreadsheets** -It's hard to recommend a particular spreadsheet as different user's needs vary so widely. While I use the ApplixWare (<http://www.vistasource.com/products/axware/>) spreadsheet, Gnumeric is another fairly mature app. that meets my admittedly modest needs easily, and seems to handle Excel files well. Kspread, like KWord, also runs well enough but doesn't completely work with Microsoft formats just yet. Read the section below for more...

**Office Suites**- These usually include a word processor, spreadsheet, presentation builder, graphics/drawing tools etc. Despite the fact that it's a non-free, commercial app. ApplixWare gets my vote as favourite office suite. Native to Linux, Applix runs well, and has more than enough features to meet my needs. Both the word processor and the spreadsheet seem to handle most MSOffice formats, and the

documentation is very good. Worth paying for in my opinion. The KOffice suite is a good looking KDE2 suite that is only let down by its incompatibility with MSOffice files, however for some people this won't be a problem, and hopefully this issue will be soon overcome by the KOffice developers. StarOffice is probably the most popular Linux suite, but frankly I can't stand it. I especially dislike the monolithic desktop design, and even on a powerful machine it takes forever to load. However it does have lots of features, it's free for personal use, and MSOffice compatibility is very good, so if you have heavy-duty requirements, you might be stuck with it. Upcoming versions, as well as close relative OpenOffice, do away with the irritating integral desktop, but don't seem to be any quicker.

## PERFORMANCE

The table below shows the approximate startup times for some of the software mentioned above. These times were measured on a 233 MHz AMD with 64meg of RAM and Linux 2.2, using the highly unscientific method of clicking on the button and then counting the delay using the toolbar clock. The figures are obviously only rough approximations in view of the measurement technique, but they do give a good indication of just how responsive an old Linux box can be.

Program	First Startup	Subsequent Starts
XWindowsCommander	1 sec	0.5 sec
Nedit	2 secs	1.5 sec
Netscape 4.77	9 secs	4 secs
Dillo	1 sec	0.5 sec
Sylpheed	1.5 sec	1 sec
xli (XLoadImage)	<1 sec	0.5 sec
XMMS	3 sec	2.5 sec
mtvp	1 sec	0.5 sec
ApplixWords	6 secs	4 secs
AbiWord	2.5 secs	2 secs

## MISCELLANEOUS

Terminal Emulators- rxvt has a combination of features and speed that make it my favourite. Plus you can customise its appearance if you are into that sort of thing.

Screen Savers are probably more of a nicety than a necessity. Xscreensaver works very well with lightweight window managers and is easy to set up. It

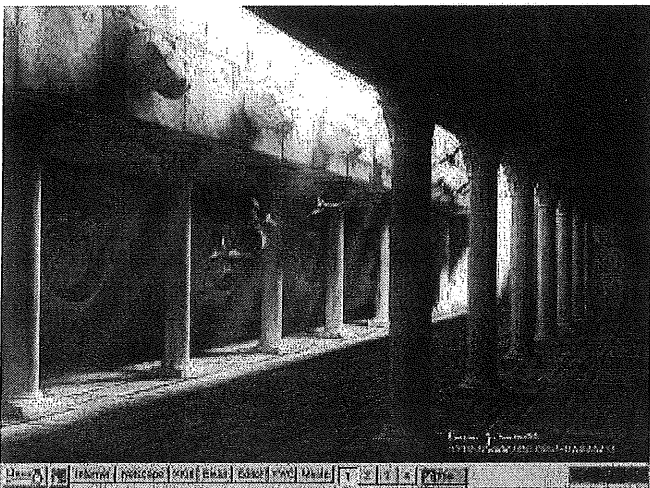
runs a randomly picked screensaver after a user-set period, and continues to change it at pre-set intervals. Run xscreensaver-demo to set the preferences, or see the man pages for more details. The easiest way to start xscreensaver automatically at login is by adding thexscreensaver & command to your window manager's startup script, eg. /usr/X11R6/bin/icewm.

TrueType Fonts are no longer a big deal to set up. Some distros (such as Mandrake 7.2 and later) include a tool for utilising TrueType fonts, even those installed on a Windows partition. This can make a big difference to the appearance to of apps; Netscape in particular. Mandrake's tool is called Drakfont, and is extremely easy to use.

Unnecessary Services or daemons can slow your machine down and lengthen bootup times. Default installations often run all sorts of servers and other stuff that you don't need. As well as using resources, these things can increase your security risk. You can use a graphical tool like tksysv , or you can manually yank the unneeded stuff (usually from /etc/rc.d/rc5.d ), but be sure to make a backup first.

## SCREENSHOTS

Here are some screenshots of some of the things mentioned above.

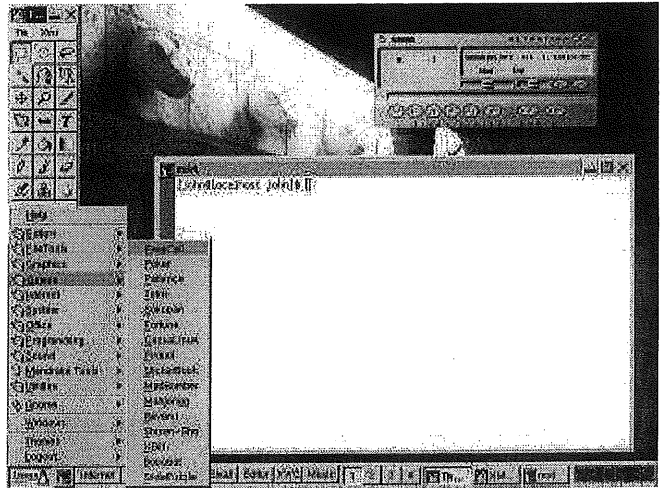


Here we can see the bare IceWm desktop. As you can see I don't use any desktop icons - these are usually covered by windows anyway. The buttons on the toolbar on the bottom edge can use icons but I prefer text labels as these are instantly identifiable even to a casual user. The Linux button in the corner brings up a menu. You can use an image file like this as a background or just select a colour.

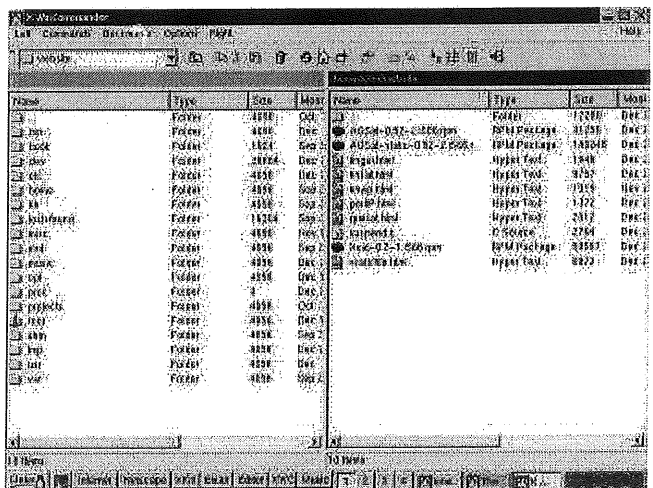
This is the XWinCommander file manager in the two-pane mode. Pretty straightforward if you have used Windows Explorer...

XWC again, this time in tree mode. Don't be put off by the colours - you can choose whatever you like...

This is Nedit, showing the Preferences drop-down menu..



This shot shows the desktop with an IceWm menu



up, also the GIMP, XMMS and an rxvt terminal window are showing.

## LINKS

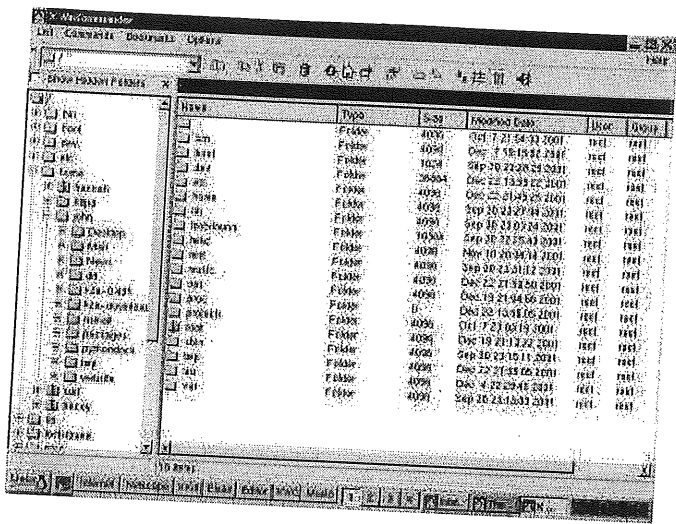
Not much here just yet, mail me if you know of any useful sites to include here.

The Linux Newbie Administrators Guide - some good info in here (<http://sunsite.dk/linux-newbie/>)

Linux For Old PCs - has some great stuff for older computers, eg. 486S (<http://homepages.ihug.co.nz/%7Eichi>)

Linux For Kids - games and educational software (<http://www.linuxforkids.com/>)

RPMFind - A huge, searchable repository of RPM packages (<http://rufus.w3.org/linux/RPM>)

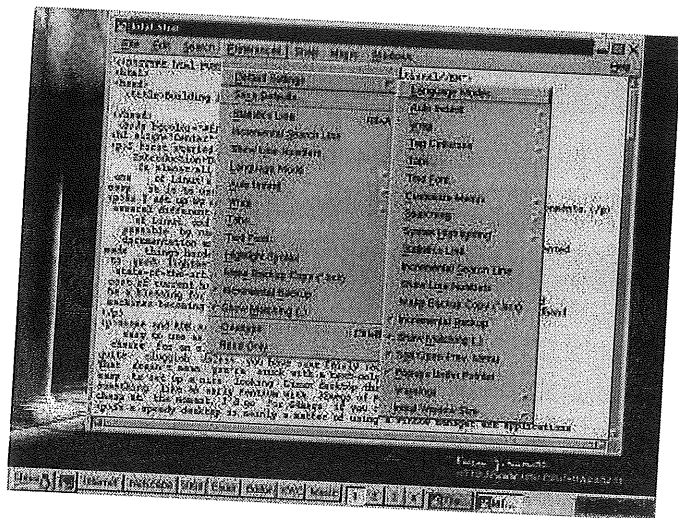


# Writing Documentation - Part 1: POD

Christoph Spiel <[cspiel@hammersmith-consulting.com](mailto:cspiel@hammersmith-consulting.com)>

The title, Writing Documentation, sounds somewhat formal. However, in this article I refer to documentation a broad sense, not only to documentation accompanying a particular piece of software, but to any related textual pieces of information. This textual information could be as short as a few lines and, for example, describe how to start a program with all of its command line options and environment variables set correctly. On the other hand, the text could be several tens of thousands lines long, elaborating all the tricks a group of users has learned over the years while using a large software system.

Choosing a Distro - A Newbies Guide  
<http://users.netwit.net.au/~pursang/distro.html>



## INTRODUCTION

With today's GNU/Linux distributions, the aspiring documentation writer immediately finds herself in fat city: there are several systems to choose from! Three documentation systems will be introduced in this article series. Here, I start with POD. Next month I'll address LaTeX in conjunction with latex2html, and in part 3 DocBook.

The systems cater different documentation needs and all have their highs and lows. But before assessing the pros and cons of the different systems, let me put up some requirements, which I want to impose on the documentation systems.

The sources of the documentation should be:

### Portable

As of December 2001, writing a "portable" file almost implies using 7-bit ASCII to encode characters. Today, 7-bit ASCII is the only encoding that works on a huge number of computers. In the future, hopefully, it will mean Unicode. Unicode can represent many more characters than ASCII does and will be as portable.

Requiring portability ensures that the texts' sources can be read and modified on a wide variety of computer systems, thereby making the documentation accessible to other programmers, which is what Open Source Software is all about.

### Searchable

Data on a computer is only as good as the access to it. -- Sounds like a commonplace, but we easily forget that documentation ought to be maintained just like source code is. We want to be able to search existing documentation for, say, a particular term or identifier. Therefore, the source of the documentation should at least be amenable to standard searching

The links below point to some smaller, simpler Linux distros.

CruX Linux (<http://www.fukt.bth.se/~per/cruX/>)

Peanut Linux (<http://www.ibiblio.org/peanut/>)

Vector Linux (<http://www.ibiblio.org/vectorlinux/>)

Sincere thanks to all to those who emailed with suggestions and feedback, even the Punctuation Police ;-). I'll try all those apps you've recommended and update this page accordingly.

This article is re-printed with permission. The originals can be found at:

<http://users.netwit.net.au/~pursang/lofat.html>

tools like, for example, the grep family (grep, agrep, rgrep, sgrep) of tools.

### **Modular**

Unless we write a short note, it is desirable that the document sources can be split into logical parts, for example sections, and the collection of all the source files is still processed as a whole by the documentation system.

### **Easy to Read**

For documentation to be "open" (as in "Open Source"), the source should be easy to read, and the system to generate the final output should be simple to learn. A documentation system will be better accepted if the writer -- and later possibly the maintainer -- can concentrate on contents rather than syntactic quirks or obscure tool chains. Just as I require certain features in the documentation's source format, so I do with the output.

### **Multiple Output Formats**

The documentation system must be capable of producing different output formats -- the more formats the better. At least HTML and PostScript (some users prefer PDF) must be supported, one for on-screen reading, the other for print outs.

HTML support in turn requires "hyperlinks", this is, references between documents or parts thereof that can be followed in a convenient way. References also help to implement the Modular Requirement in my list of source format features.

### **Automatic Reference Generation**

All references should be resolved automatically as far as this is possible. For example, the system should resolve cross references within in the document and should allow for footnotes or sidenotes to be placed and numbered without the help of the writer. The index and bibliography also should be generated automatically.

Let us now look at a particularly easy to use format: Perl's Plain Old Documentation.

## **PERL'S PLAIN OLD DOCUMENTATION (POD)**

The "Plain Old Documentation" system that ships with every Perl distribution is simplest documentation system in my selection. It is simple to learn, simple to use, but -- and I hesitate to write therefore -- also the most limited of the three. Anyhow, the article you are currently reading (yes this one!) has been prepared with POD. If it is good for the goose, it can't be bad for the gander...

The big advantages of POD are

It comes with Perl. So you probably already have it on your Linux box. Try

```
pod2man --help
```

to see if it is installed.

It offers a small and well-thought-out set of document structuring and markup instructions.

The POD translation tools render at least four different output formats:

HTML, UNIX manual pages, LaTeX (which serves as base for a further translation into PostScript), and plain ASCII text.

### **Syntax**

The POD format defines three different kinds of paragraphs. Paragraphs are separated from each other by one or more completely (!) empty lines.

### **Ordinary Paragraph.**

Any line that does not start with at least four spaces or an equal sign is considered ordinary text. Paragraphs are separated by one or more empty lines. This means, the documenter simply writes one paragraph after the other with at least one blank line between each pair.

Ordinary paragraphs will be filled and justified (if the output format allows for justification) when output.

### **Verbatim Paragraph.**

Lines indented by four or more spaces are considered verbatim text. They are output exactly as typed. All formatting instructions that we will see later, are disabled in verbatim paragraphs.

### **Command Paragraph.**

Command paragraphs start with an equal sign "=" in column zero, immediately followed by an identifier. Usually, command paragraphs consist of single lines. Yet they are syntactically paragraphs, because they are separated by blank lines before and after them.

### **Sectioning**

Text is sectioned by =headN commands, like

```
=head1 primary_heading
=head2 secondary_heading
=head3 tertiary_heading
```

which also define the section headings *primary\_heading*, etc.. How many heading levels (this is largest N permitted) actually are accepted, depends on the POD-to-something converter. For example, *pod2man* allows only two levels, *pod2html* allows up to six levels.

I have added line and column numbers to the source of the examples. The line numbers do not appear in the real source. They are included here to point out the empty lines that must separate the command paragraphs, this is, those starting with an equal sign in column 0. Additionally, I have added a column-number ruler at the beginning of the next example to clarify where column 0 starts.

Example:



```

1      2      3      4      5
0123456789012345678901234567890123456789
001234567890
1      =head1 Hardware
2
3      The physical parts of your computer are
called "hardware".
4
5      =head1 CPU
6
7      The CPU is the most important part of
your computer.
8
9      =head1 Mass Storage
10
11     Mass storage devices store data
permanently.
12
13     =head2 Hard Disk Drives
14
15     Hard disk drives provide fast random
access to data.
16
17     =head2 Magnetic Tapes
18
19     Magnetic tapes provide slow sequential
access to data.
20
21     =head1 Software
22
23     This is where the trouble starts ...

```

### Lists

Itemized, enumerated or description lists are produced with

```

=over N
=item label
=item label
...
=back

```

where =over N starts a list that is indented at least N spaces, and extends until =back. Depending on the first label the POD-to-something translators generate an itemized list (label = \*), a numbered list (label = 1) or a description list (label starts with a letter).

Example: itemized list

Again, I have added line numbers to alert the reader of the (many) empty lines used for separating the command paragraphs.

Source

```

1      =over 4
2
3      =item *
4
5      Fruit, particularly non-imported fruit
like ...
6
7      =item *
8
9      Though not tasty, vegetables should make
up a large part of your
10     daily diet.
11
12     =item *
13
14     Fish is much easier digestible than
meat. Therefore, ...
15
16     =back

```

Result

- Fruit, particularly non-imported fruit like ...
- Though not tasty, vegetables should make up a large part of your daily diet.
- Fish is much easier digestible than meat. Therefore, ...

Example: enumerated list

Source

```

1      =over 4
2
3      =item 1.
4
5      Ensure that the power switch is in
position "OFF".
6
7      =item 2.
8
9      Plug in the power cord.
10
11     =item 3.
12
13     Switch the power switch in position "ON".
14
15     =back

```

Result

1. Ensure that the power switch is in position "OFF".
2. Plug in the power cord.
3. Switch the power switch in position "ON".

Example: description list

Source

```

1      =over 8
2
3      =item Robert
4
5      Lead singer
6
7      =item Jimmy
8
9      Lead guitar
10
11     =item John-Paul
12
13     Bass guitar
14
15     =item John
16
17     Drums and percussion
18
19     =back

```

Result

```

Robert
  Lead singer
Jimmy
  Lead guitar
John-Paul
  Base guitar
John
  Drums and percussion

```

### Inline Markup Commands

Within Ordinary Text, several markup commands are recognized. All markup commands start with a single capital letter and enclose their argument within angle

brackets: LETTER<argument>. The argument can consist of multiple words, which can span more than one line.

**I<argument>**

Render argument in italics. I corresponds to the HTML tags em and var, thus it is primarily used for emphasizing words or marking up variables.

Examples:

- Do *not* remove your Linux kernel!  
is produced by

```
Do I<not> remove your Linux kernel!
```

- Use **cd** *directory* to change your working directory to *directory*.

is generated with

```
Use B<cd> I<directory> to change your working directory to I<directory>.
```

**B<argument>**

Render argument bold. B corresponds to the HTML tag b. It is used to emphasize in text and to mark up program names or switches.

Examples:

- Always shut down your machine before switching it off.

comes from

```
B<Always> shut down your machine before switching it off.
```

- podchecker accepts the options **-warnings** and **-nowarnings**.

is the result of

```
B<podchecker> accepts the options B<-warnings> and B<-nowarnings>.
```

**C<argument>**

C marks up code or anything else which is to be taken literally. The corresponding HTML tags is are code, samp, and tt.

Examples:

- Every C-program must have a function called *main*.

is generated by

```
Every C-program must have a function called C<main>.
```

Boolean false is represented by [1 1 0], and boolean true by [1 1 1].

is produced by

```
Boolean false is represented by C<[1 1 0]>, and boolean true by C<[1 1 1]>.
```

**L<reference> or L<description|reference>**

Link to an existing reference. If description is omitted, the link's text is reference, otherwise it is description. Using L is a bit tricky. Therefore, I have devoted the next section to it.

**Cross References**

The L-command is distantly related to HTML's <a href = "reference">description</a>, however, in POD, reference is not a general unified resource locator (URL). reference can only refer to (automatically by the POD-to-something translator) generated labels. These labels are inserted for every =head and =item. The label associated with =head heading is heading downcased, but otherwise unchanged, e.g.

```
=head1 A Multi-Word Heading (MWH)
```

automatically gets assigned the label

```
a multi-word heading (mwh)
```

The labels of =items are prefixed by item\_, spaces are replaced by underscores, and non-alphanumeric characters are replaced by their hexadecimal ASCII code prefixed by a percent sign. Anybody expected an easy rule? So, one of the items in this article,

```
=item Automatic Reference Generation.
```

has the label

```
item_Automatic_Reference_Generation%2E
```

because the ASCII number of the period is 46 in decimal or 2e in hexadecimal.

Example:

Source

```
=head1 Introduction
Section L<"concepts"> introduces the basics of the field.
=head1 Concepts
...
=head1 Synchronization
=over 4
=item Deadlocks
=item Race Conditions
=item Recovering from Deadlocks
=back
```

How to cope with deadlocks was already discussed in L<Deadlocks|"item\_Deadlocks">, and L<Recovering from Deadlocks|"item\_Recovering\_from\_Deadlocks">.

Result

**INTRODUCTION**

Section concepts introduces the basics of the field.

**CONCEPTS**

...

## SYNCHRONIZATION

### Deadlocks

### Race Conditions

### Recovering from Deadlocks

How to cope with deadlocks was discussed in [Deadlocks](#), and [Recovering from Deadlocks](#).

The L-command is very limited in its use, for the writer cannot insert places to refer to with an L-command; HTML-like "anchors" are missing.

A second limiting factor are some POD translators trying to be smart and decorate link with additional text. For example, *pod2latex* mangles both references to items in the above example:

```
How to cope with deadlocks was discussed in the
\textsf{Deadlocks$|}$"item\_Deadlocks")      entry
elsewhere in this document, and the
\textsf{Recovering
from
Deadlocks$|}$"item\_Recovering\_from\_Deadlocks")
entry elsewhere in this document.
```

where I have underlined the words added by *pod2latex*. Clearly, we want a better mechanism. The mechanism exists in format-specific paragraphs.

## FORMAT-SPECIFIC PARAGRAPHS

We have just seen that the L-command is somewhat difficult to control. Why can't we simply use a HTML-reference? The terse answer, "because POD is not HTML", leads to the solution. If we had a way to say "this text is for HTML, this line is for LaTeX, and this paragraph is for "SnaFoo", we could use the specific markup provided by these formats.

The special command

```
=for format paragraph_of_text
```

tells a translator to look at `format` before processing `paragraph_of_text`. If the translator feels responsible for handling `format`, it transforms `paragraph_of_text` according to its own rules, otherwise it completely ignores the paragraph. The second part of the translator's name usually specifies which format it takes care of. For example, *pod2man* transforms `=for man` paragraphs, *pod2html* processes `=for html` paragraphs, and so on.

As all command paragraphs, a `=for format` paragraph ends at the first completely empty line that follows the introducing `=for`.

A consistent document structure will show "forks" whenever specific formats are used, because a `=for format` clause ought to appear for each desired output format, otherwise we punch a logical holes into the document.

```
This is an ordinary paragraph, which is
```

```
processed by all translators.
=for html <p>This paragraph only appears if the
file is processed
with <b>pod2html</b>.</p>
=for latex This very paragraph is only treated
by {\bf pod2latex}.
=for text I am a paragraph for the *pod2text*
formatter.
We now continue with the ordinary text for all
formatters.
```

The translators ignore unknown formats, which means we can invent special paragraphs for our own purposes! For example, to "comment out" a paragraph, write

```
=for comment Can someone clarify the next
section?
```

Another popular use is the emacs format :) To switch emacs into text-mode when preparing a POD-file, start the file with

```
=for emacs -*- text -*-
```

or end it with

```
=for emacs
Local Variables:
mode: text
End:
```

The emacs-users who are using the hyperbole add-on can convert their "dumb" POD-files into hyper-linked collections (well -- hyperbole can do a lot more than that, but hyperlinks are a beginning) of files with

```
=for hyperbole <(std-reference)>
```

where `<(std-reference)>` is a hyperbole button taking you to another file which holds the reference documentation of `std` when you click the button in emacs.

## PROGRAMS THAT WORK WITH POD

*pod2html*, *pod2man*, *pod2latex*, *pod2text*

### Translators from POD

to HTML, UNIX manual pages, LaTeX and plain text respectively.

### podchecker

Simple syntax checker for POD files.

## PROS AND CONS OF POD

### Pro

Simplicity  
Conversion speed

### Cons

Lack of tables  
No program to generate an index supplied by default

## FURTHER READING

Manual pages of *perlpod(1)*, *pod2man(1)*,



pod2html(1), pod2latex(1),  
pod2text(1), and podchecker(1).

This article is re-printed with permission. The originals  
can be found at:

<http://www.linuxgazette.com/issue73/spiel.html>

# Bookreview: The qmail Handbook

Bruno Sousa <bruno@linuxfocus.org>

## ABSTRACT

"qmail Handbook" is a book written by Dave Sill, the author is well known for the "Life With Qmail". The book was published by Apress. The price is something about AU\$80, not too expensive and I think it is worth the money.

## INTRODUCTION

- What's qmail ?
- What's a MTA ?
- What's a MUA ?
- qmail was written by Whom ?

Can you answer all the questions above? If you can I believe you are an experienced user (probably an sysadmin or a consultant). If not don't get worry, because this book will explain these things and a lot more.

The book is suitable for both kind of users, the experienced qmail user and the newbie (like me). The book focuses on readers who want to learn how to administrate a qmail system. The author mentions this in the introduction chapter (XXI)

The language used in the book is very accessible, and in my opinion does not confuse the reader with "difficult technical terms", they are well explained and the images support that explanation.

Another good thing, before I forget it: The author does not assume that you have a specific distribution like Slackware or SuSE, but e.g. in the installation he details the differences between different versions of Unix (Linux, FreeBSD).

## THE BOOK'S ORGANIZATION

Here I could detail all the chapters of the book and tell my opinion about them, but I won't do it for two reasons:

My experience can't be compared against the Dave's experience (something like 2 years compared to 15

years).

At the time of writing I could not talk about all the chapters since I haven't read them all. Forgive me for this!

I write about the chapters that I have read and describe them in my own words.

### Chapter 1, "Introducing qmail"

This chapter will help you to find answer for the question above. If you know what qmail is and you are not sure if it is suitable for your needs, then the answer is in this chapter, since it presents the features of qmail.

### Chapter 2, "Installing qmail"

As the name says it is a step-by-step configuration guide for the installation, the creation of users, directories, the assignment of the required permissions...

### Chapter 3, "Configuring qmail: The Basics"

Here you find the steps needed to put the system into operation (communicating with other MTAs). It explains the qmail control files, the aliases, and other things, like the mechanism of the qmail-users.

### Chapter 4, "Using qmail"

Covers how the users can send and receive messages and some utilities to control the mailboxes.

### Chapter 5, "Managing qmail"

Gives an explanation about the management commands of qmail. How to manage a queue.

### Chapter 6, "Troubleshooting qmail"

Helps you to understand the log files, and to manage them correctly. And talks about common problems.

### Chapter 7, "Configuring qmail: Advanced Options"

This chapter might be interesting for those who have sendmail and decided to migrate to qmail...

### Chapter 8, "Controlling Junk Mail"

This chapter is something interesting nowadays, the days of spam. This helps to deal with it.

### Chapter 9, "Managing Mailing Lists"

As the name says it talks about the mailing lists and the more known programs to manage that, like Majordomo, ezmlm.

### Chapter 10, "Serving Mailboxes"

If you need to provide remote access to the mailboxes then the questions can be answered here. It talks about POP3 and IMAP protocols.

### Chapter 11, "Hosting Virtual Domains and Users"

If you need to implement virtual mail accounts then all the processes needed for that are here described using Vpopmail and/or VmailMgr.

### Chapter 12, "Understanding Advanced Topics"

Some advanced Topics like the combination of qmail and SQL or LDAP. And if you have windows machines you can protect them with anti-virus software in the mail system.

## APPENDICES

Here you find more detailed explanations, if after reading the book you have any doubts, or still open questions then read this chapter.

## CONCLUSION

The book is what the title says, "qmail handbook". It really is the handbook of qmail. So if you are an administrator of the mail system and you use qmail then this is for you. No matter if you are an experienced user or not. The book can be on your desk all the time, if any problem occurs, search in the index. It's easy to use. Some blank pages are included for (who knows) any annotation.

Probably the reader, when sending an email to anyone, e.g. a person in Dallas sends an email to his French friend in Paris, may wonder what happens to the email? What route does it take? ...

Well, the book can help you answering these questions, since it gives a technical description of qmail but also, gives theoretical explanations about the processes that qmail handles, and so on...

You can find other peoples opinions if you search for the book at amazon.

The final decision is always up to you...

## REFERENCES

The Official qmail site, <http://cr.yip.to/qmail.html>  
The Unofficial qmail site, <http://www.qmail.org>

*This article is re-printed with permission. The originals can be found at:*

<http://www.linuxfocus.org/English/March2002/article232.shtml>

# Compiler Design with Python

Dinil Divakaran <[dinildivakaran@rediff.com](mailto:dinildivakaran@rediff.com)>

## INTRODUCTION

### Purpose

C is obviously the first choice for anybody interested in designing a production quality compiler or interpreter. But what about designing a 'little

language' just for the fun of it (or maybe, for more serious purposes)? Why worry when you have Python - and some really smart tools to go with it!

### The toolkit

We will be using Python Lex-Yacc (PLY) for recognizing tokens and parser construction. These tools are very closely modeled after traditional lex/yacc. If you know how to use these tools in C, you will find PLY to be similar. You can download PLY from the site [system.cs.uchicago.edu/ply](http://system.cs.uchicago.edu/ply).

We will need the modules `lex.py` and `yacc.py` in our working directory. Also we require Python version 2.1 or higher.

## GETTING STARTED

Before going into the details of implementation, let us get down to the basics.

### Tokens

What are tokens ? Tokens are symbols like +, -, \* or /; or words such as `begin`, `end`, `if` or `while` - which we would like to identify as operands, reserved words, keywords etc. Tokens must be defined as regular expressions.

### Defining the Language

Since we are writing a compiler for a particular language with constructs that we would like to include, the first thing to do is to define the language. This is done by writing a set of rules or grammar for the particular language. For example, if you want your language to provide the 'if-then-else-endif' construct, then one simple way to write a rule for it is :

```
if_statement : IF LPAREN statement
              RPAREN multiple-statements ELSE
              multiple-statements ENDIF
```

where (1) IF, LPAREN, RPAREN, ELSE and ENDIF are tokens for recognizing `if`, `(`, `)`, `else` and `endif` respectively. (2) 'statement' and 'multiple-statements' are again different constructs for which rules are written.

### Parsing

In simple terms, parsing is the method of verifying whether the input program does match the rules given to the parser. There are different types of parsing methods. But we needn't go into the details involved. It is only sufficient to know that, given a set of rules (as seen in the example above) the parser sees, if the input constructs corresponds to the rules defined.

## IMPLEMENTATION

Well, we are now ready to implement a compiler. There are different phases of a compiler like token recognition, parsing, taking semantic actions, producing intermediate code, optimizing it, and finally producing the required output assembly code. The steps that we are taking will also be quite similar.

## DEFINING THE LANGUAGE

As said earlier, the first step is to define the language which you want your compiler to accept. You should be certain which constructs and operators you want to provide. Constructs such as 'while', 'if', 'assignment statements' etc are common. So are operands such as +, -, \*, / etc. You should write down the rules for your language. A set of rules for a language accepting assignment statements are given below.

```
assign_statement : VAR EQUALS statement
statement       : statement ADDOP term
                | statement SUBOP term
                | term
term            : term MULOP factor
                | term DIVOP factor
                | factor
factor          : VAR
                | NUM
                | LPAREN statement RPAREN
```

Throughout our discussion, we adopt the convention that words in upper cases (NUM, VAR, EQUALS, ADDOP, SUBOP, MULOP, DIVOP, LPAREN, RPAREN) are tokens and those in lower cases (assign\_statement, statement, term, factor) are rules.

## TOKEN DEFINITION AND RECOGNITION

Next, we have to define the tokens that we are using. In our example, we have used nine tokens - NUM, VAR, EQUALS, ADDOP, SUBOP, MULOP, DIVOP, LPAREN and RPAREN. The following program implements a simple lexer for tokenizing our language. [text version]

```
import lex

# List of token names. This is compulsory.
tokens = (
    'NUM',
    'VAR',
    'EQUALS',
    'ADDOP',
    'SUBOP',
    'MULOP',
    'DIVOP',
    'LPAREN',
    'RPAREN'
)

# Regular statement rules for tokens.
t_VAR      = r'[a-zA-Z][\w]*'
t_EQUALS   = r'='
t_ADDOP    = r'\+'
```

```
t_SUBOP    = r'\-'
t_MULOP    = r'\*'
t_DIVOP    = r'\/'
t_LPAREN   = r'\('
t_RPAREN   = r'\)'

# A regular statement rule with some action code.
def t_NUM(t) :
    r'\d+'
    try:
        t.value = int(t.value)
    except ValueError:
        print "Line %d: Number %s is too large!" %
            (t.lineno, t.value)
        t.value = 0
    return t

# Define a rule so that we can track line numbers.
def t_newline(t):
    r'\n+'
    t.lineno += len(t.value)

# A string containing ignored characters (spaces and tabs).
t_ignore = ' \t'

# Error handling rule
def t_error(t):
    print "Illegal character '%s'" % t.value[0]
    t.skip(1)

# Build the lexer
lex.lex()

# Get the input
data = raw_input()

lex.input(data)

# Tokenize
while 1 :
    tok = lex.token()
    if not tok :
        break
    print tok

# If you want to include
# reserved words, it is usually
# easier to just match a variable name
# (identifier) and do a special
# name lookup in a function like this:
reserved = {
    'if' : 'IF',
    'then' : 'THEN',
    'else' : 'ELSE',
    'while' : 'WHILE',
    ...
}

def t_VAR(t):
    r'[a-zA-Z][\w]*'
    t.type = reserved.get(t.value, 'ID') # Check
    for reserved words
    return t
```

## PARSING

Parsing is quite easy when we use yacc.py. The parser invokes the lexer for getting tokens. So we have to import the lex module that we had written earlier. Now, corresponding to each rule, we define a function and write the rule itself as a document. Within the function we can write the semantic actions needed to be taken. For our example language, the parsing can be done as shown below.

```

# Yacc example

import yacc

# Get the token map from the lexer that we defined
# earlier. This is required.

from ourexlex import tokens

__var_names = {}

def p_assign_statement(t) :
    'assign_statement : VAR EQUALS statement'
    __var_names[t[1]] = t[3]

def p_statement_plus(t) :
    'statement : statement ADDOP term'
    t[0] = t[1] + t[3]

def p_statement_minus(t) :
    'statement : statement SUBOP term'
    t[0] = t[1] - t[3]

def p_statement_term(t) :
    'statement : term'
    t[0] = t[1]

def p_term_times(t) :
    'term : term MULOP factor'
    t[0] = t[1] * t[3]

def p_term_div(t) :
    'term : term DIVOP factor'
    t[0] = t[1] / t[3]

def p_term_factor(t) :
    'term : factor'
    t[0] = t[1]

def p_factor_num(t) :
    'factor : NUM'
    t[0] = t[1]

def p_factor_var(t) :
    'factor : VAR'
    if __var_names.has_key(t[1]) :
        t[0] = __var_names[t[1]]
    else :
        print "Undefined Variable", t[1], "in line
no.", t.lineno(1)

def p_factor_expr(t) :
    'factor : LPAREN statement RPAREN'
    t[0] = t[2]

# Error rule for syntax errors
def p_error(t):
    print "Syntax error in input!"

# Build the parser
yacc.yacc()

while 1:
    try:
        s = raw_input('enter > ')
    except EOFError:
        break
    if not s: continue
    yacc.parse(s)

```

Here each function accepts a single argument, *t*, which is a tuple. The values of *t*[*i*] are mapped to grammar symbols as shown here:

```

def p_statement_plus(t):
    'statement : statement ADDOP term'
    #   ^           ^           ^   ^
    #   t[0]         t[1]         t[2] t[3]
    t[0] = t[1] + t[3]

```

## SEMANTIC ACTIONS

The semantic actions are the steps that the parser takes when it reduces the input to a particular rule. In our example above, the actions correspond to that of an interpreter. For a simple compiler, the semantic action may be to produce the assembly code corresponding to a rule.

Suppose you want to produce 8086 assembly instructions as output. Let us assume that 'bx' is a temporary register. Now, whenever we see an operand, we store the contents of the accumulator in the temporary register, and store the operand itself in the accumulator. Thus, the last seen operand (or the result of an evaluation) will always be in the accumulator.

```

def p_factor_num(t) :
    'factor : NUM'
    __output_fp.write("\tmov bx,ax\n"%f) # bx
<-- [ax]
    __output_fp.write("\tmov ax,0x%x\n"%t[1]) # ax
<-- t[1]

```

where, '*\_\_output\_fp*' is a file pointer to an output file

Since the operands of an operation (be it binary or unary) is now available, we can write the semantic action for adding as :

```

def p_statement_plus(t) :
    'statement : statement ADDOP term'
    __output_fp.write("\tadd ax,bx\n")
    # ax <-- [ax] + [bx]

```

Similarly, whenever we see a new variable, we can allocate a register for the variable (a stack location is a better choice to store local variables), and remember the register allocated by using a dictionary. The variable name is the key and the register name is the value. Every time a variable name is referenced, the dictionary is searched using the name of the variable as key, to get the corresponding register name.

## OPTIMIZATION

For a C compiler, the assembly instructions are not produced so early as we have depicted here. Actually, it is the intermediate code that is produced. Then the intermediate code is optimized and finally the assembly code is generated.

Since, optimization is itself a vast topic, we will only discuss a simple optimization technique, namely peephole optimization. The easiest way to implement peephole optimization is to hand-code a particular assembly program and compare it with the code your compiler produces.

For example, if your assembly instruction set does not have an instruction for multiplying, then you can

make your compiler produce code for multiplication by repetitive addition. A simple optimization that you can make here is this : if you have one operand as 1, then you can store the other operand as the result; instead of going for the repetitive addition, which will obviously be a loop. Again, since the multiplier determines the loop count, you can choose the lower of the two operands as the multiplier.

Another example for peephole optimization is in the use of jump's. Look at the following example :

```

        jmp .L1
        .
        .
.L1     jmp .L2
        .
        .
.L2     add ax,bx

```

Here, the first jump statement can be changed to reduce the number of jumps, as is shown below.

```

        jmp .L2
        .
        .
.L1     jmp .L2
        .
        .
.L2     add ax,bx

```

There are various algorithms for producing optimized codes. The methods discussed above are only the beginning steps towards complex time and space saving optimization techniques.

## WHAT NEXT ?

The illustration that we have gone through is not a full-fledged compiler. To complete it, we need to implement more and more common constructs. It's only a matter of writing rules for the constructs, defining regular statement for every new token, writing parser functions for the grammar, and finally taking semantic actions in those functions.

*This article is re-printed with permission. The originals can be found at:*

<http://www.linuxgazette.com/issue79/divakaran.html>

# Configuring GDM 2.2

Mark Nielsen <articles@gnujobs.com>

## INTRODUCTION

GDM, or GNOME Display Manager, is a graphical login service for your computer when it boots up. Basically, it makes a nice pretty screen to look at before you log in. With the standard installation of RedHat, and I assume other distributions, GDM is really cool. As a user, you can run the program "gdmphotosetup" to set the picture of you that will show up in GDM when your computer starts. As the "root" user, you can configure GDM with "gdmconfig" which lets you set a lot of cool options. So why am I writing this article when you can do all this yourself? I will show you some bad things you are not suppose to do.

Configuring GDM naughtily. In the gdm.conf file, I changed these options (which you can also probably do in the gui setup program).

```

TitleBar=true
Browser=true

LockPosition=false
SetPosition=true
PositionX=0
PositionY=700

```

Here is my /etc/X11/gdm/Init/Default script.

```

#!/bin/sh

/usr/X11R6/bin/xsetroot -solid "#363047"

### This next item is a huge security risk.
### It basically sets up an xterm with the user
"mark".
xterm -r -fn 6x12 -geometry +0+25 -e
'/etc/X11/gdm/mark.sh' &
#xterm -r -fn 6x12 -geometry +0+25 -e
'/etc/X11/gdm/dummy.sh' &
### This puts a picture on the background.
/usr/bin/xsri -geometry 500x500+600+300
/etc/X11/gdm/im000048.jpg
### This puts xeyes on the screen to watch your
mouse pointer.
xeyes -geometry +800+650 -bg white -fg green
-outline blue &
### Christmas all year round.
xsnow -santaspeed 10 -santa 2 -snowflakes 1000
-whirl 4 -windtimer 30 &
### A clock down to the second.
xclock -digital -geometry +600+650 -update 1 &

### The popular mine game. Sorry, doesn't seem
like you can position it.
### it just pops up in the middle of the
screen.
gnomine &

### Maelstrom is pretty cool.
Maelstrom &
### xboard pops up behind GDM, so we can't use
it.
#xboard &
### Chromium is a cool arcade-like old fashioned
game.
chromium-setup &
### Look at the light shining on the earth.
kworldclock -geometry +750+0 &
### A silly creature for your desktop.

```



```
amor &
```

```
Now the contents of '/etc/X11/gdm/mark.sh'.  
#!/bin/bash
```

```
trap "" HUP  
trap "" INT  
trap "" QUIT  
trap "" KILL  
trap "" TSTP  
  
su -l mark  
exit  
exit
```

Why is the above script dangerous? Well, people can do stuff without logging in. That is why I put a bunch of traps in the script and made it exit as soon as someone quits as the user 'mark'. I don't want anyone to execute root commands, so you have to make it so root exits as soon as the user 'mark' quits and you have to trap the script so that someone doesn't cancel the quit -- which would leave them logged in as root. Still, the whole thing is bad and you shouldn't do it, even though I do. As an alternative to my xterm session, you could use "chroot", which I did successfully. It can be a little tricky to setup a chroot environment, but you can do it. Here is a sample of an account I called "dummy". Remember, the /chroot/named environment has to look like the root directory with a /bin, /sbin, /lib, and all the other directories if you want the user to be able to do anything at all, and of course /etc/passwd.

```
Now the contents of '/etc/X11/gdm/dummy.sh'.  
#!/bin/bash
```

```
trap "" HUP  
trap "" INT  
trap "" QUIT  
trap "" KILL  
trap "" TSTP  
  
chroot /chroot/dummy su -l dummy  
exit  
exit
```

## CONCLUSION

GDM is really cool, and I assume KDM is just as cool. I just like to configure GDM to be nice to look at when I or someone else sits down at them.

You might want to have other games playing in the background of your gdm session. I tested various games, some work and some don't. Remember, every program you run is a potential security hole if someone can somehow execute commands through the program or know how to screw it up causing your computer to get messed up. Obviously, doing something like this on a client computer should get you fired.

## REFERENCES

Configuring GDM 2.2  
(<http://www.tcu-inc.com/Articles/19/GDM.html>)

Gnome Display Manager  
(<http://www.socsci.auc.dk/~mkp/gdm/>)

6-24-1999 Setting up xdm.  
(<http://www.tcu-inc.com/mark/articles/Xdm.html>)

if this article changes, it will be available  
<http://www.gnujobs.com/Articles/19/GDM.html>

Mark works at [AudioBoomerang.com](http://AudioBoomerang.com) which creates, delivers, and tracks personalized multimedia email, web, and newsletter campaigns. He works as a consultant delivering end products to [AudioBoomerang.com](http://AudioBoomerang.com) clients, such as advanced customized statistical reports used for demographic or psychological profiles for future campaigns. In his spare time, he writes articles relating to Free Software (GPL) or Free Literature (FDL) and is involved with the non-profit learning center [eastmont.net](http://eastmont.net).

*This article is re-printed with permission. The originals can be found at:*

<http://www.tldp.org/LDP/LG/current/nielsen3.html>

# Improving Hard Disk Performance with *hdparm*

By Piter Punk <[piterpk@terra.com.br](mailto:piterpk@terra.com.br)>

Translated from the Portuguese by William N. Zanatta  
<[wzanatta@uol.com.br](mailto:wzanatta@uol.com.br)>

Nowadays the IDE devices already have a high transfer rate (by UltraDMA technology), but there are still other ways to improve your hard-disks performance and we'll show how to do it with the *hdparm* utility.

## INTRODUCTION

*hdparm* is an utility which provides us a powerful tuning control over HDs (HD PARaMeters) and this is what we'll be discussing in this document. Sometimes your HD is set not to use its maximum power as it could and that's why you may get annoyed with its performance. With *hdparm* we can magically change this to reach its maximum performance using all of its features.

## LOOKING THE HARD DISK

The first thing to do, is to gather all information about your hard drive and the current settings. These information will be used as a base for us while configuring the hard disks. Be extremely careful in all the steps you take because any misconfiguration may damage your disk partially (data) or entirely (hardware).

By now, lets assume /dev/hda as our disk. Take the command:

```
darkstar:~$ hdparm -i /dev/hda
```

You should get some info like:

```
/dev/hda:
Model=QUANTUM FIREBALLct20 20,
FwRev=APL.0900, SerialNo=552114732078
Config={ HardSect NotMFM HdSw>15uSec Fixed
DTR>10Mbs )
RawCHS=16383/16/63, TrkSize=32256,
SectSize=21298, ECCbytes=4
BuffType=DualPortCache, BuffSize=418kB,
MaxMultSect=8, MultSect=off
CurCHS=16383/16/63, CurSects=-66060037,
LBA=yes, LBASects=39876480
IORDY=on/off, tPIO={min:120,w/IORDY:120},
tDMA={min:120,rec:120}
PIO modes: pio0 pio1 pio2 pio3 pio4
DMA modes: mdma0 mdma1 mdma2 udma0 udma1 udma2
udma3 udma4 *udma5
AdvancedPM=no
Drive Supports : ATA/ATAPI-5 T13 1321D
revision 1 : ATA-1 ATA-2 ATA-3
ATA-4 ATA-5
```

But, you may ask yourself "What the hell is this?". Heah, don't be afraid this information will make you happy soon. Here we have many important and useful information...let's look at some:

- **MaxMultSect:** This field tell us what is the maximum number of sectors your hard disk can read at a time.
- **MultSect:** This one says the current number of sectors being read at a time.
- **PIO and DMA modes:** These are the modes supported by your hard drive. The one marked with an asterisk (\*) is the currently set.
- **AdvancedPM:** If 'yes', means that your drive supports APM (Advanced Power Management).

Another command issued to get other information is:

```
darkstar:~$ hdparm /dev/hda
```

This one brings:

```
/dev/hda:
multcount = 0 (on)
I/O support = 0 (16-bit)
unmaskirq = 0 (off)
using_dma = 0 (off)
keepsettings = 0 (off)
nowerr = 0 (off)
readonly = 0 (off)
readahead = 8 (on)
geometry = 2482/255/63, sectors = 39876480,
start = 0
```

In a brief description...

- **multcount** is the number of sectors being read at a time;
- **I/O support** indicates the operating mode of your hard disk (16/32/32sync);
- **using\_dma** tell us whether the drive is using the DMA feature or not;
- **keepsettings** keeps the settings after a soft reset (don't touch unless you know what it is);

- **readonly** is normally set to 1 only for CD-ROMs, this setting tells the system whether the device is read-only or not;
- **readahead** shows how many sectors ahead will be read when you access the hard drive;

If you didn't understand some of these don't get bored, you are not a dumb, and we will discuss them as you read this document. Some of these parameters are related to your hard drive hardware physically and not logically, so you cannot change them unless you change the hardware (and if you do it, you will probably cry for damaging your hard disk and destroy all your data, =] ).

## DEVICE SETUP

And now...the show! We are going to setup our HD. **REMEMBER:** Mistakes during the setup process may damage your hard disk and all of its data. The information provided by 'hdparm -i ' now, is your driver. Follow them and you must not get any problems.

### I/O Support

Well, unless you have a (E)ISA IDE interface card, the rest (PCI/VLB), all support 32bits mode. If your box is newer than a 486, probably you have a PCI IDE controller. If it's not, check for it...

```
hdparm -c0 // Set operating mode to 16-bits.
hdparm -c1 // Set operating mode to 32-bits.
hdparm -c3 // Set operating mode to 32-bits
synchronized.
```

The mode '3' only is needed for some chipsets. People often use mode '1' for best performance. We didn't find any info about mode '2' (supposed to be 16-bit synchronized).

### MultSect or Multcount

This one is simple. Check your HD's MaxMultSect info for what you can do. We set our MultSect to 8 since our HD supports that, so...

```
# hdparm -m 8 /dev/hda
```

Remember to change /dev/hda to YOUR device and '8' to the MaxMultSect supported by your hard disk as provided by 'hdparm -i '.

### Activating DMA

The most simple of all. Simply type:

```
# hdparm -d 1 /dev/hda
```

to set your DMA mode to ON. Your card must supportd the DMA mode.

### PIO and DMA modes

You can set both of these using the same flag '-X'. This one, if not used with EXTREME care may eject

your hard drive (BELIEVE IT!) and make it the first HD to arrive the Moon by itself. Set just the modes supported by your hard disk.

Hmm, it works like this...for normal DMA modes (multiword DMA or mdma), use -X32 + (DMA identifier number). For mdma2 it would be:

```
# hdparm -X34 /dev/hda // 32 + 2 (from mdma2)
```

For the PIO and UltraDMA modes the process is almost the same. The difference is that the base number for the PIO modes is 8 and for the UltraDMA modes it is 64. The hard disk used while writing this document supports ATA100, so it was put in udma5 mode using:

```
# hdparm -X69 /dev/hda
```

Keep in mind that the highest DMA modes are available just for some chipsets.

The ATA66 and ATA100 modes requires a 80-way IDE cable. Think that put you disk in ATA100 without these cables will not work.

#### Readahead

The option readahead IS NOT the same as multcount. The multcount refers to the possibility of the hardware to read more than one sector at a time while the readahead option is the number of sectors ahead your computer should read. The readahead feature is great when reading big-size files but it brings down the performance for short-size files. A good idea is to leave the value of readahead the same as the multcount so it will not be needed to make more than an access per time to read more sectors ahead.

If you are going to access big files you can set the readahead to a greater value. The default value is 8 sectors/read access (something like 4kb).

Now the syntax:

```
# hdparm -a N /dev/hda
```

N is the number of sectors for readahead.

#### FINAL COMMENTS

There are many other features you can set using hdparm. Most of them are covered in the hdparm manpage. These ones we covered are just the most common.

The configuration will be reset when you reboot your machine (the keep settings will not work as it covers just soft reboots). Put the commands in your rc.local (maybe, for large configuration, it would be a good idea to have a rc.hdparm or something like this).

*This article is re-printed with permission. The originals*

*can be found at:*

<http://tldp.org/LDP/LG/issue79/punk.html>

## Discover the universe: Celestia & Open Universe

By Katja Socher <[katja@linuxfocus.org](mailto:katja@linuxfocus.org)>

### ABSTRACT

Celestia and Open Universe are programs that let you travel through the universe and explore all the planets and stars. If you ever looked upon the sky at night dreaming of flying through space visiting all those bright shining stars and planets you will love them! Both are real time programs, that means that you can view all the planets and stars move along their paths, trace them and orbit them.

### WHAT IS CELESTIA?

With Celestia you can go on a space travel and explore our universe. When you start the program you will first see Jupiter's moon Io. The voyage can begin.

But when you run the program for the first time you should first make a guided tour and go on a demo flight by pressing d-key. You will leave Earth and see some very nice pictures of our blue planet. Next is the moon, followed by pictures of the sun. Now you see the planets on their orbits. After this you travel to see Saturn, some star constellations and the milky way before going home again.

Now you have an impression of the program it's time to go on your own exploration:

### HOW TO USE IT

There are several ways to navigate through space. You can press the return key and enter the name of the planet, star or constellation. Then choose a travel speed (e.g. F2, F3) and press g-key. Off you go!

You can also travel through the universe by clicking and dragging with the mouse and selecting an object with a left mouse click. If its name is then shown on the top left of the program window the object is selected. This is really a cool feature as you can select almost every point that you can see on your screen. Press c-key to get the selected object in the center of your window. Choose a travel speed if you haven't already done so and press g-key. You are now traveling to your selected object. By clicking g-key again you can get closer to it.

With t-key you can track an object. If you press n-key you get the names of the planets and moons, b-key



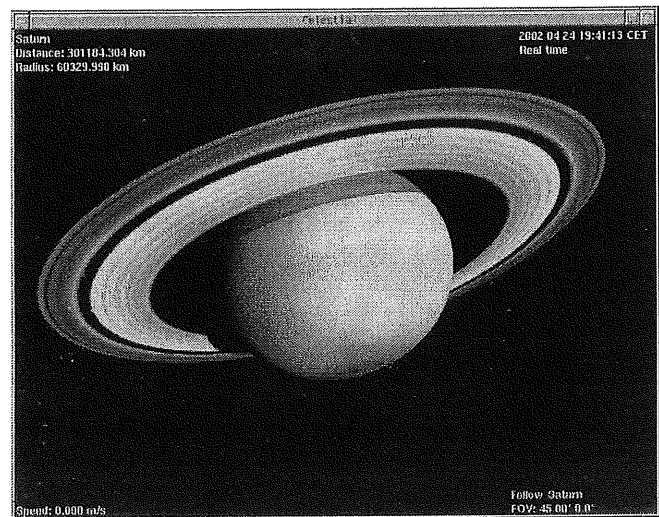
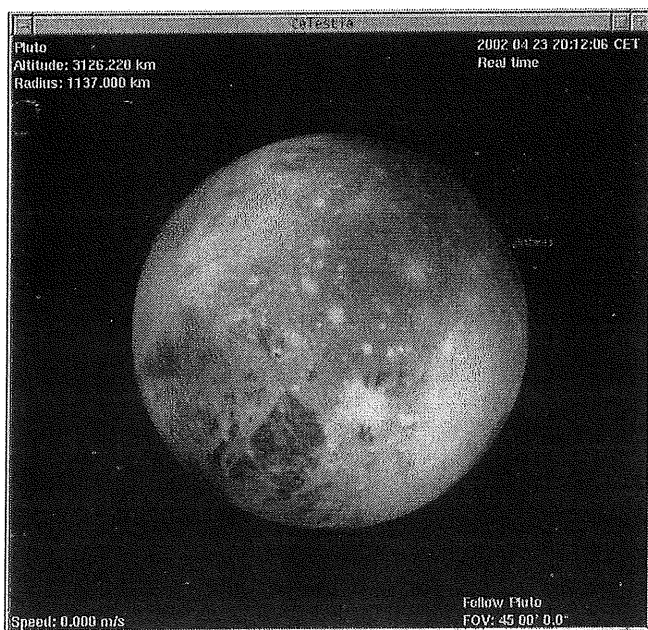
gives you the names of the stars, = the constellation names and with v-key you get some information about your target. Pressing any of this buttons again lets disappear the names and information again. This information really is very useful for your orientation. A click on "h" (followed by "g" of course) brings you back to our sun which I find very helpful when I am lost in space once again ;-).

You can select different travel speeds with F2 to F6 (F2 being the slowest). Pressing F1 stops everything.

To get closer you have to press g-key again until you are as close as you want to. You can read "Traveling" written on the left bottom of the screen in addition to the moving stars and planets. With ESC you stop everything.

To find out more read the Readme of the program which is included in the top level directory of the source code. If you prefer to read about the keybindings online then take a look at *the keybindings page*.

Here are a few screenshots:



The installation should be straight forward.

```
./configure --prefix=/usr/local/celestia
make
make install
```

This will install Celestia to /usr/local/celestia/bin

## OPEN UNIVERSE

Open Universe is a program similar to Celestia. It doesn't have that many stars and planets because it focuses on our solar system. It hasn't been updated for a while now as the people of OpenUniverse are busy helping with Celestia, but it has a nice navigation bar where you can choose your target from a list of planets, stars etc. so that you don't get lost that easily. I really think it is worth looking at, too.

## HOW TO USE IT

If you start it you will see some beautiful pictures of the earth.

When using it for the first time you might also want to see a demo first. Click on Options (on the bottom of the menu) and an options menu pops up. Here you can choose demo mode. If you want to know the names of the stars and planets you are passing by make sure that you also have the options "info", "star labels" and "body labels" ticked.

Now lean back and enjoy watching for a while.

Okay, now it's time to go on a space exploration by ourselves! In OpenUniverse you are a bit more restricted than in Celestia but are also less likely to get lost in space that way. To navigate through space you choose an object from the source list and another from the target list. You can also set the camera mode. If you choose "body to body" you get a view from the target as seen from the source. If you choose "orbit" you orbit around the target. Now click "go there" and your voyage begins!

## INSTALLATION

The version used for this article was celestia-1.2.2. You can download it from the Celestia webpage (<http://www.shatters.net/celestia/>). The package, celestia-1.2.2.tar.gz, is about 10Mb big. To use it you need a 3D graphic card and the Mesa 3D graphics libraries. Packages, headerfiles and libraries should already be included on the CDs of your Linux distribution.

You can read the manual to get more information on how to use OpenUniverse. If you need help while traveling pressing h will also give you some clues.

## INSTALLATION

The version used in this article was openuniverse-1.0beta3. You can download it from the OpenUniverse webpage (<http://www.openuniverse.org/>). The package, openuniverse-1.0beta3.tar.gz, is about 4Mb.

It requires a bit of manual code change to get it compiled but it is really worth it.

It is said on the installation page that the glui libs are optional but I could not get it to work without them. You get the glui\_v2\_1\_beta sources at <http://www.cs.unc.edu/~rademach/glui>.

To compile the glui libraries:

unpack:

```
tar zxvf glui_v2_1_beta.tar.gz
```

Edit the makefile and set the GLUT\_ variables to fit your Linux system:

```
GLUT_LIB_LOCATION=/usr/X11R6/lib
GLUT_INC_LOCATION=/usr/X11R6/include/GL
```

Set the CC variable:

```
CC=g++ -O3
```

Compile:

```
make
```

Copy the resulting library lib/libglui.a to the place where your other open GL libs are:

```
cp lib/libglui.a /usr/X11R6/lib
```

Copy the header files:

```
cp algebra3.h arcball.h glui.h quaternion.h
stdinc.h viewmodel.h /usr/X11R6/include/GL/
```

To install OpenUniverse:

```
tar zxvf openuniverse-1.0beta3.tar.gz
./configure --with-gl-libs=/usr/X11R6/lib --with-
glui-inc=/usr/X11R6/include/GL --
prefix=/usr/local/openuniverse
```

To get the whole thing to compile under Mandrake I had to add

```
#include <string.h>
```

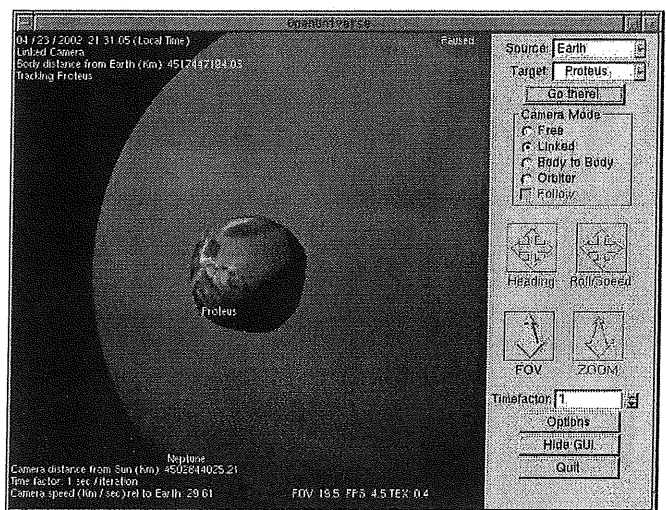
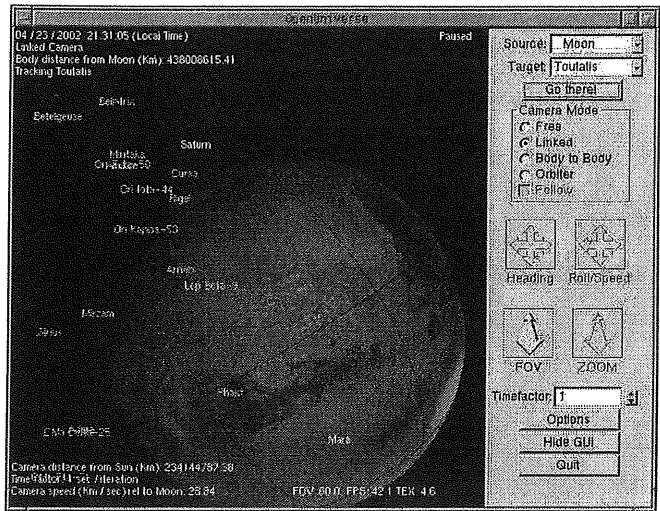
in the files src/cfglex.l src/cfgparse.y  
src/milkyway.cpp src/stars.cpp  
and add

```
#include <GL/gl.h> and #include <string.h>
```

in the file src/ou.h

```
make
make install
```

A few screenshots of OpenUniverse:



Have fun!

This article is re-printed with permission. The originals can be found at:

<http://www.linuxfocus.org/English/May2002/article244.shtml>

## Mandrake 8.2 First Impressions

Anthony Barker <[@xminc.com](mailto:@xminc.com)>

Yesterday, I eagerly rushed home with three freshly minted mandrake 8.2 CDs. From what I'd read this is a 'stability' release. Not so many new features - but they all work. Besides that, I wanted to try the new linux kernel and virtual memory manager, the updated supermount (for my wife), avoid any '.so hell' (libpng version 3 vs version 2) and finally I wanted to fix the zlib and openssl security issues.

I always have like Mandrake in the past for their

windows integration, centralized menu system and 'leading edge' versions of open source apps and have been known to give money via their website.

## BACKGROUND

I have been using linux for about 8 years (all major distros) - but have only heavily gotten into it in the past year or two. My home machine is "yesterdays" dream machine. An Athlon 1.33, with 512K RAM, 50 GIG disk, ATI TV Wonder Rage 128, DEC Tulip Network card, SoundBlaster 128, a DVD/CD Drive and CD Burner. The onboard ATA-RAID controller and onboard soundcard are disabled.

## PREPARATION

I first booted into my old install of Mandrake, loaded the configuration panel (diskdrake) and wrote down all the hard drive partitions. My configuration was simple, as this is a workstation.

```
/          hda7      ReiserFS    3 GIG
/home      hda8      ReiserFS    3 GIG
(Where I store my user data)
2 swap partitions
/mnt/win_c hda4      fat32       2 GIG
/mnt/fatdata hda5    fat32       2 GIG
```

At the command prompt I did a 'cat /proc/interrupts' and noted down the interrupts. I then copied the /etc /tmp and /root directories into /home. This way you can always refer to your old configuration. I store downloads in the /tmp/apps - so I can quickly reinstall some of my favorite apps without re-downloading them. Feeling confident, I proceeded without backing up my data.

## INSTALLATION

The installation tries to be slicker than older version. I had problems in the past (8.1) particularly with the disk configuration when mixing JFS, Reiser and EXT2 file systems. First, I tried the upgrade option, but it bombed, telling me I didn't have enough disk space to continue. I wasn't sure if this was caused by the ReiserFS giving back wrong information or that Mandrake needs a lot of free space to do the installation. I rebooted, made sure the First CD was in the drive and started again (Expert Install). This time I selected full install - of everything.

One feature I wish distros would have when you select individual packages is a "sort by size" - so you could easily prune down the size of your install without having to hunt and peck. I chose grub as my bootloader as it is somewhat technically superior to lilo. Then set my security level to standard (msec) and configured my video card with XFree86 4.2 with 3D support. The install prompted me for a root password and my user name (ops... see troubleshooting). I setup CUPS to use my old Brother 720 Laser printer. It tried to configure my networking, but as I didn't have my ADSL settings on hand, I skipped that section. While my daughter crawled all over me, the

install reformatted my Y drive and installed the files in about 11 minutes (1.9 GIG). The mandrake install is intuitive and leaves Windows XP in the dust.

## TROUBLESHOOTING

I booted the fresh install - up came Mandrake with a new pretty splash screen (Aurora) and I was autologged in as my user. You can delete the 'quiet' option from the boot options in the grub or lilo config file to get your messages back. Mandrake had left my /home directory intact - but had overwritten my user directory /home/ant, as I had used the same user name (ARGHH !!#@!!- 1 day of python scripts I wrote gone...). I kicked myself and then went into the control panel to configure the ADSL connection. I noticed the control panel looks much spiffier - but has mostly the same functionality. I spent a 1/2 hour trying to configure the adsl connection using the wizard and then resorted to the command prompt. su'ed to root, ran adsl-setup and then adsl-start without luck. Then I

```
cd /etc/ppp
>options .....# emptied my /etc/ppp/options
file
updatedb&
locate pppoe.conf
locate pap-secrets
```

and then copied the old config files to the /etc/ppp/ directory. I also edited the /etc/resolv.conf for dns resolution.

Next, I had to troubleshoot the sound. I had essentially the same problem with mandrake 8.1 with my SoundBlaster 128 (ES1370). The ALSA Sound system doesn't support it properly. I loaded the configuration panel again, went to the services area, and disabled ALSA. Opened up my /etc/modules.conf with vi and noticed it had added bunch of fancy looking stuff. I copied my backup modules.conf from /home/etc/modules.conf, which included the ES1370.

```
lsmod
modprobe es1370
```

voila - I had sound.

To fix the known kdm insecurity I edited the /etc/X11/xdm/Xaccess file and commented out:

```
* CHOOSER BROADCAST #any indirect host can get a
chooser
to:
** CHOOSER BROADCAST #any indirect host can get a
chooser
```

And then restarted kdm.

## OBSERVATIONS AND TWEAKING

I threw in a Sesame Street DVD for my daughter to watch - xine worked perfectly. There is also seems to be better integrated support for stuff such as

scanners and other peripherals.

I am a heavy mozilla user - and I was a bit disappointed they didn't squeak 0.99 into the final release of Mandrake 8.2. Although Mandrake gives you a default set of bookmarks ( I had lost my old ones) they don't make any effort to do the mime mappings (a.k.a. Helper Applications) - why not include xpdf/ghostview(ps)?

The menu system has more options aimed at easing new users into linux and DiskDrake seems to be improved a lot. Sound configuration is hidden in harddrake, which may confuse new users coming from Windows.

gcc is still at version 2.96 - I guess they are waiting for Redhat to go to 3.0x before upgrading.

They also have not yet included the mosfet theme liquid theme for kde (a little OSX in your linux), which is available at <http://kde-look.org> - I haven't got it working yet in Mandrake as the directory structure differs.

## MORE TWEAKING

I imported my windows fonts, setup verdana anti-aliasing on my desktop without any problems and then proceeded to install:

- fluxbox (my favorite light windows manager)
- xplanet - planet earth with live cloud cover for your background (make sure you follow the instructions for mandrake in kde)
- iKons 0.6 for KDE2/3 ( a nice kde theme)
- cowsay/cowthink - everyone should have /usr/games/fortune | /usr/local/bin/cowsay in their /etc/profile script
- acroread
- openoffice
- flash plugin

Finally, I disabled services in the control panel that I don't need (webmin, proftpd, linuxconf etc) and configured the integrated Mandrake security level to high (what happened to the firewall?). I also created an encrypted filesystem in DiskDrake - the only hassle is you have to type in your 20 key AES 128 encryption key every time you reboot.

## SUMMARY OF FIRST IMPRESSIONS:

Mandrake has done a lot of work cleaning up the user interface and making Linux more intuitive. Moreover, it is supposed to be more stable - the kernel as well as Mandrake's tools ( although I have not experienced that so far). Perhaps I have been a bit harsh because I lost my data directory (my own fault - but of course I internally blame the vendor). Overall, I think mdk 8.2 is the best Mandrake release so far, a candidate for the best linux distribution, and perhaps my favorite desktop operating system.

## Excellent

---

- Linux 2.4.18 kernel
- Integrated menu system - menudrake
- Spiffy New control panel
- Ease of use for newbies (menus, supermount...)
- Windows integration
- DiskDrake is excellent and intuitive (JFS, XFS, ReiserFS, EXT3 & Encrypted Filesystems)
- Multi-media (xine!) and peripheral support
- Integrated file sharing and remote windows framebuffer access (Rfbdrake)- Virtual Network configuration ...(I have yet to test this but it looks cool)
- urpmi - the graphical rpm installer system rocks and is much faster
- zlib and OpenSSH security bugs are fixed

## Good

---

- Lots of games for those who like them
- All the latest versions of linux applications
- Application selection - postfix, not sendmail - proftpd not wu-ftp
- HardDrake/DiskDrake more stable

## Could be better

---

- No Troubleshooting wizards for newbies
- Configuration tools still need some work although they are better (my ES1370 soundcard problem - some users having nvidia problems)
- No mozilla 0.99 - nor is mozilla customized at all, galeon 1.2 isn't included, nor is netscape 4.7 (for when you are stuck). Browsers are key for home machines ....
- KDE 3.0 RC2 is not included in the Mandrake Linux 8.2 I downloaded (they forgot it...?)
- Applications compiled with libpng 2 will not work - eg opera (i believe) -
- you will get the error: "libpng warning: Application is running with png.c from libpng-1.2.1 or libpng error: Incompatible libpng version in application and library"
- Many users have reported font problems although I haven't experienced them
- No abiword... why not? Is it because a mandrake programmer is the main developer for kword?(I read it was due to a font problem actually)
- kde still loads too slowly (fixed in kde 3 ?)
- upgrade issues(mine failed)
- nautalis, the gnome file manager is a pig (not mandrake's fault)
- Where's the firewall gone? Does msec (the security manager) do this? There is no mention in the help.

**Anthony lives in Toronto, Canada and enjoys teaching his daughter and wife the wonders of zsh ;-) He runs a small consulting company (<http://www.xminc.com/>) that advises firms on email security and develops workflow applications.**

*This article is re-printed with permission. The originals can be found at:*

*<http://www.xminc.com/linux/mandrake82.html>*

# GUI Programming in C++ using the Qt Library, part 1

Author: Gaurav Taneja <[tech@gauravtaneja.com](mailto:tech@gauravtaneja.com)>

In the vast world of GUI Development Libraries there stands apart a Library, known as 'Qt' for C++ developed by Trolltech AS. 'Qt' was commercially introduced in 1996 and since then many of the sophisticated user interfaces have been developed using this Library for varied applications.

Qt is cross-platform as it supports MS/Windows, Unix/X11 (Linux, Sun Solaris, HP-UX, Digital Unix, IBM AIX, SGI IRIX and many other flavors), Macintosh (Mac OS X) and Embedded platforms. Apart from this 'Qt' is object oriented, component based and has a rich variety of widgets available at the disposal of a programmer to choose from. 'Qt' is available in its commercial versions as 'Qt Professional' and 'Qt Enterprise Editions'. The free Edition is the non-commercial version of Qt and is freely available for download ([www.trolltech.com](http://www.trolltech.com)).

## GETTING STARTED

First of all you need to download the library, I assume that you have downloaded the Qt/X11 version for Linux as the examples will be taken for the same.

You might require the superuser privileges to install, so make sure you are 'root'.

Let's untar it into /usr/local directory :

```
[root@Linux local]# tar -zxvf qt-x11-free-3.0.1
[root@Linux local]# cd qt-x11-free-3.0.1
```

Next you will need to compile and install the library with the options you require to use. 'Qt' Library can be compiled with custom options suiting your needs. We will compile it so that we get gif reading, threading, STL, remote control, Xinerama, XftFreeType (anti-aliased font) and X Session Management support apart from the basic features.

Before we proceed further, remember to set some environment variables that point to the correct location as follows:

```
QTDIR=/usr/local/qt-x11-free-3.0.1
PATH=$QTDIR/bin:$PATH
MANPATH=$QTDIR/man:$MANPATH
LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
export QTDIR PATH MANPATH LD_LIBRARY_PATH
```

You can include this information in your .profile in your home directory.

```
[root@Linux qt-x11-free-3.0.1]# ./configure -qt-
gif -thread -stl -remote -xinerama -xft -sm
```

```
[root@Linux qt-x11-free-3.0.1]# make install
```

If all goes well, you will have the 'Qt' library installed on your system.

## YOUR FIRST STEPS WITH 'Qt'

In order to start writing programs in C++ using the 'Qt' library you will need to understand some important tools and utilities available with 'Qt' Library to ease your job.

### Qmake

Qmake lets you generate makefiles with the information based on a '.pro' file.

A simple project file looks something like this:

```
SOURCES = hello.cpp
HEADERS = hello.h
CONFIG += qt warn_on_release
TARGET = hello
```

Here, 'SOURCES' can be used to define all the implementation source for the application, if you have more than one source file you can define them like this:

```
SOURCES = hello.cpp newone.cpp
or alternatively by:
SOURCES += hello.cpp
SOURCES += newone.cpp
```

Similarly 'HEADERS' let's you specify the header files belonging to your source. The 'CONFIG' section facilitates to give qmake info about the application configuration. This Project file's name should be the same as the application's executable. Which in our case is 'hello.pro'.

The Makefile can be generated by issuing the command:

```
[root@Linux mydirectory]# qmake -o Makefile
hello.pro
```

### Qt Designer

Qt Designer is a tool that let's you visually design and code user interfaces using the 'Qt' Library. The WYSIWYG interface comes in very handy for minutely tweaking the user interface and experimenting with various widgets. The Designer is capable of generating the entire source for the GUI at any time for you to enhance further. You will be reading more about the 'Qt Designer' in the articles that will follow.

## HELLO WORLD!

Let's begin by understanding a basic 'Hello World' Program. Use any source editor of your choice to write the following code:

```
#include <qapplication.h>
```



```
#include <qpushbutton.h>
int main( int argc, char **argv )
{
    QApplication a( argc, argv );
    QPushButton hello( "Hello world!", 0 );
    hello.resize( 100, 30 );
    a.setMainWidget( &hello );
    hello.show();
    return a.exec();
}
```

Save this code as a plain text file('hello.cpp'). Now let's compile this code by making a project file (.pro) as follows:

```
TEMPLATE = app
CONFIG += qt warn_on_release
HEADERS =
SOURCES = hello.cpp
TARGET = hello
```

Let's save this file as 'hello.pro' in the same directory as that of our source file and continue with the generation of the Makefile.

```
[root@Linux mydirectory]# qmake -o Makefile
hello.pro
```

Compile it using 'make'

```
[root@Linux mydirectory]# make
```

You are now ready to test your first 'Qt' Wonder. Provided you are in 'X', you can launch the program executable.

```
[root@Linux mydirectory]# ./hello
```

You should see something like this:



Let's understand the individual chunks of the code we've written.

The First two lines in our code include the `QApplication` and `QPushButton` class definitions.

Always remember that there has to be just one `QApplication` object in your entire Application.

As with other c++ programs, the `main()` function is the entry point to your program and `argc` is the number of command-line arguments while `argv` is the array of command-line arguments.

Next you pass these arguments received by `Qt` as under:

```
QApplication a(argc, argv)
```

Next we create a `QPushButton` object and initialize it's constructor with two arguments, the label of the

button and it's parent window (0 i.e., in it's own window in this case).

We resize our button with the following code:

```
hello.resize(100,30);
```

`Qt` Applications can optionally have a main widget associated with it. On closure of the main widget the Application terminates.

We set our main widget as:

```
a.setMainWidget( &hello );
```

Next, we set our main widget to be visible. You have to always call `show()` in order to make your widget visible.

```
hello.show();
```

Next we will finally pass the control to `Qt`. An important point to be noted here is that `exec()` keeps running till the application is alive and returns when the application exits.

*This article is re-printed with permission. The originals can be found at:*

<http://www.linuxgazette.com/issue78/taneja.html>

## A Linux Fax Server for a Windows Network

Author: Pedro Fraile <pedro.fraile@solvay.com>

### INTRODUCTION

The firm I work for had a fax system integrated in the corporate e-mail platform, Microsoft Exchange, for sending and receiving. One day after a software upgrade, the system broke. We needed to find something with the equivalent functionality but with the following conditions:

- Minimum cost, or better still, no cost at all, especially regarding software licenses.
- Transparent integration with the end user's software tools (basically Microsoft Office).
- No need to install any software on the client side, even free software, in order to minimize the work load of the network administrators.

This article describes how the integration of several open source applications on a Linux platform has fulfilled all of these conditions.

### ACKNOWLEDGMENTS

I want to express my gratitude to various persons or organizations without whose assistant I would have never written this article. First, to the members of the IT department in the Solvay Química S. L. plant at Torrelavega, Spain. Second, to my firm's hierarchy for their approval and support of this article's writing.



Last and most especially, to all the contributors to the Open Source projects mentioned throughout this text, to the participants in the HylaFAX mailing list (which have given me essential information) and to Craig Kelly, developer of the `smbfax` client tool.

## SYSTEM OVERVIEW

To clarify which computer I'm talking about where, I'll refer to the PC where the fax software is installed as `TOSERFAX`.

The applied solution involves the HylaFAX software. This application controls the installed modems, distributes the incoming faxes and sends the outgoing ones.

The incoming faxes are converted to PDF format and forwarded via SMTP e-mail to their respective destinations. PDF was chosen because Acrobat Reader is part of the standard software platform at the site. The destination is ascertained via certain rules as will be later explained.

If someone wants to send a fax, he prints the document in a printer queue on `TOSERFAX`, which Samba makes visible to all the other computers. The print job will cause an e-mail to be sent to the user that has spooled the job. This e-mail includes the URL of a web form created on-the-fly in the Apache web server. The web form allows the user to fill in the fax details, particularly the destination phone number. Once the user has completed the form, upon clicking on the "Send" button, the fax is finally put on the outgoing queue.

## HARDWARE AND SOFTWARE

`TOSERFAX`'s hardware is the following:

- PC Dell Optiplex GX150, running a 1 Ghz Pentium III processor, with 256 MB of RAM and a hard disk of 20 GB. The modems are 3Com US Robotics 56K Faxmodem.

As far as software is concerned:

- The base system is the SuSE Linux 7.2 distribution. It includes HylaFAX version 4.1beta2, the Apache web server version 1.3.19 and the SMTP server sendmail version 8.11.3.
- Samba version 2.2.3a.
- Fax sending from the clients is implemented using the package `smbfax`, version 1.4.

## HYLAFAX INSTALLATION AND CONFIGURATION

The installation of HylaFAX was carried out following the standard procedures, clearly explained in the documentation. The most delicate part is the modem configuration. HylaFAX does not include a template for the US Robotics 56K Faxmodem. However, a search in its mailing list provided the needed information, which resulted in the file

`/var/spool/fax/etc/config.ttyS0` (and `config.ttyS1` for the second modem). The first of these files can be found here.

## RECEIVING FAXES

Our plant has several telephone numbers that are connected to fax machines. The telephone exchange can divert phone calls originally made to one extension to a different one. This feature makes it possible to centralize the reception of all faxes in `TOSERFAX` without any change in the phone numbers that are accessible to the public.

For example, suppose the Purchasing Department has 5550001 as fax number, while Logistics has 5550002. One of `TOSERFAX`'s modems is connected to the internal extension 1700. The PBX diverts all incoming calls to 5550001 and 5550002 to the extension 1700, where `TOSERFAX` receives the fax.

But of course, the person that should receive the faxes to Purchasing is not the same one that should get the ones to Logistics. HylaFAX manages incoming faxes by way of the scripts `faxrcvd` and `FaxDispatch`, placed in `/var/spool/fax/bin`. The discrimination we want requires knowledge of the fax number the fax was originally sent to, which is not known in the standard version of `faxrcvd`. A workaround is to recover that number from the session log, assigning it to a variable, for instance `TOPHONE`.

```
TOPHONE=$(SAWK '/SESSION BEGIN/ (print $NF; exit)'  
log/c$(COMMID))
```

The new versions of `faxrcvd` and `FaxDispatch` can be found here and here.

The standard version of `faxrcvd` sends the fax to the addressee as a postscript attachment in an e-mail. This is not the best option at my plant, as the standard PC does not include a postscript viewer. But it does include a PDF viewer, and postscript files can be converted to PDF.

However, here we run into a small problem, related to the sending of the e-mail message with the attached file. `TOSERFAX` uses as SMTP relay a Windows NT server running IIS version 4. For some reason that I have not been able to discover, this server could not distribute the e-mails with attachments created with `faxrcvd`.

The solution was to use the tool "metasend", included in the packages `metamail` 2.7.19. The scripts `metasend.sh` and `tiff2pdf.sh` succeed in sending the fax, previously transformed into PDF format, in a way that is acceptable for the SMTP relay. It is worth mentioning that these scripts invoke the tools `tiff2ps` and `gs`.

## SENDING FAXES

There are several fax clients written to be used with HylaFAX, for multiple platforms. However, IT

administrators at Torrelavega would rather avoid any software installation on the clients. The only operation at the site's PCs should be, at the most, the configuration of a network printer, and it should be made automatically by the end user himself, if possible.

Using a printer queue has the added advantage that any application that is able to print a document (that is, practically all applications) will be able to fax. In this respect, the fax solution described in this article is clearly superior to other proprietary systems installed in Microsoft Exchange, which only allow to send faxes generated by some applications, for instance those in the Microsoft Office suite.

The package `smbfax`, developed by Craig Kelly, fulfills the above mentioned requirement. The underlying idea is very clever: the client prints the document he wants to fax in a printer queue, configured in TOSERFAX with Samba, and which features a postscript printer. The printing provokes in fact the execution of a perl script, which puts the printed document into a file and sends the client an e-mail with an URL in it. This URL is a link to a web form created on the fly in the web server at TOSERFAX (Apache). The client clicks on the URL, fires the browser and, using the web form, fills in the number or numbers the fax should be sent to, chooses whether a cover page should be added, and other details. Finally, upon clicking on the "Send" button, the fax is put in the outbound queue. In case there is any error processing the job, the client will equally be notified by e-mail. Obviously, this system requires knowing the identity of the user who is faxing (it must be possible to get the authentication credentials he have acquired upon logging in the Windows PC) as well as his e-mail address.

The installation of `smbfax` is straightforward. The package documentation clearly explains the different steps, and repeating them here would just be redundant.

Configuring Samba, on the other hand, does show some interesting tricks. The pertinent file can be seen off a link on the web-site printed at the end of this article. The following lines must be emphasized:

```
[global]
workgroup = DOM
netbios name = TOSERFAX
security = DOMAIN
winbind uid = 10000-20000
winbind gid = 10000-20000
template homedir = /home/win/%D/%U
winbind separator = +
printer admin = @DOM+PRINTADMIN
...

[print$]
path = /etc/samba/printers/
browseable = yes
read only = yes
write list = @DOM+PRINTADMIN,root

# The fax queue is configured in this section
[fax]
comment = Fax queue
path = /tmp
```

```
printable = Yes
writable = no
create mode = 0700
guest ok = no
postscript = Yes
printing = lprng
print command = /usr/local/smbfax/smbfax -r
queue %u %s
lpq command = /usr/local/smbfax/smbfax show
lprm command = /usr/local/smbfax/smbfax
dequeue %j
```

As a Samba server, TOSERFAX is included in a Windows 2000 domain (Active Directory). Samba version 2.2.3 features support for "winbindd", which provides client authentication based on the credentials obtained upon starting a session in the domain. As a consequence, to create the Windows users in the Linux box is no longer needed. Each client that connects for the first time to the Samba server will be identified by the combination <Domain name>+<User name>, and will earn an "uid" in the range 10000 - 20000. Inside the [fax] section, the line

```
print command = /usr/local/smbfax/smbfax -r queue
%u %s
```

invokes the program `smbfax` passing in the parameter `%u` the name of the user, identified as previously explained.

Inside the [global] section, the line

```
printer admin = @DOM+PRINTADMIN
```

gives administrative rights on the printer queues to all members of the PRINTADMIN group in the NT domain DOM. These users will be able to configure printers, install drivers (for different Windows versions) and grant printing rights to the domain users by means of the standard remote administrative tools which are present in an NT or Windows 2000 box, and that use Remote Procedure Calls (RPC). And all this in a transparent way, without being aware that the printer server is not really a Windows box, but a Linux one.

The members of the DOM+PRINTADMIN group must of course have been granted write access to the path `/etc/samba/printers`. This is achieved by establishing the necessary permissions in the Linux filesystem:

```
$ chown -R DOM+PROWNER:DOM+PRINTADMIN
/etc/samba/printers
$ chmod 0775 /etc/samba/printers
```

Driver installation is an especially interesting feature. It is possible to install at TOSERFAX the drivers of a postscript printer for all Windows versions that are used at the site: 95, NT and 2000. Once this work is done, any client that connects to the printer queue for the first time will be able to auto-install the needed drivers. We achieve therefore one of the goals of the network administrators: no configuration work needed on the client side.

Additionally, any member of the PRINTADMIN group may restrict access to the printer queue, using the NT access control lists (ACL).

The only question still unanswered is how to reach by e-mail the users that want to send a fax. Thanks to winbindd the user has been authenticated, but, which is the e-mail account? Lacking a way to read this information from the Active Directory, maybe using OpenLDAP, the solution is to manually add to the "aliases" file the list of possible fax users, with their e-mail addresses

```
DOM+User1: email-1@domain.com
DOM+User2: email-2@other-domain.com
```

and so on. Execute "newaliases" and the system is ready.

## SYSTEM MAINTENANCE

Once each and every component is configured, the last thing to do is to automate some basic housekeeping tasks. This is easily fulfilled adding to /etc/crontab the following lines:

```
0 21 * * * root test -x /usr/sbin/faxqclean
&& /usr/sbin/faxqclean
25 23 * * * root test -e /usr/sbin/faxcron &&
sh /usr/sbin/faxcron | mail faxmaster
```

Beware though that the HylaFAX package included in SuSE 7.2 leaves faxcron in /etc/cron.daily. Therefore, you will have to move it to apply the proposed scheme.

## CONCLUSION

The combination of HylaFAX, Samba, smbfax and other open-source packages on a Linux system has allowed to integrate an efficient centralized fax service in a Windows environment, realizing the expectations of the IT managers, especially the lack of additional software installation on the client side.

*This article is re-printed with permission. The originals can be found at:*

<http://www.brynjarrhauksson.com/LDP/LDP/LG/issue79/fraile.html>

# Securing a Heterogeneous Network with Free Software Tools

Author: Georges Tarbouriech <[georges.t@linuxfocus.org](mailto:georges.t@linuxfocus.org)>

## ABSTRACT

This article was first published in a Linux Magazine France special issue focusing on security. The editor, the authors, the translators kindly allowed LinuxFocus to publish every article from this special issue. Accordingly, LinuxFocus will bring them to you as soon as they are translated to English. Thanks to all the people involved in this work. This abstract will be reproduced for each article having the same origin.

## PREAMBLE

Security in computers networks is probably one of the biggest technology challenges of the 21st century. However, like for many worrying fields, everybody talks about it, but the ones who should feel the most affected do not seem to have detected the scale of the potential disaster. The "most affected" of course, are the main software or system designers. The best example, once again, comes from Redmond, where security seems to be a word, at least much less "under control" than marketing, for instance.

Fortunately, the two last decades of the 20th century have seen the birth of Free Software and the philosophy going with it. If you "wish" to improve the security of your machines, your systems, your networks... this is where you will have to look for. The Free Software community has done much more about security than all the big software companies together. That said, tools don't make it all, and securing a network, for instance, is an almost permanent job: new changes all the time!

This means you will never be able to say that a network is 100% secure. You can only reduce the risks. What we show here, is only a small part of what you can do to limit these risks. After reading this special issue (Author's note: remember, this article was part of a Linux Magazine France special issue focusing on security), you will know a bit more about security, but in no way will you be able to say that your network is secure. You have been warned.

Last but not least: such an article can't be exhaustive. There is a lot of literature on the matter and it is far from having gone round the problem. Accordingly, don't expect from this article to mention everything, as far as OSes, tools, configuration, use... are concerned.

To end with this preamble, let's add that some parts

of this article are borrowed from LinuxFocus, but don't worry, with the author's agreement: it turns out to be one and the same person!

## PRESENTATION

First, we will talk about the structure of a very heterogeneous network, containing systems more or less widespread. The more OSes, the worse the complexity since not all systems are not equal in front of adversity. Furthermore, the machines used as servers should have different functions in a network: we will have a diversified network.

Next, we will go through a range of tools essential to improve security. The choice will be arbitrary: they are far too numerous to mention all of them. Obviously, we will explain how to secure machines and networks with these tools. The following chapter will review the features of different systems during the securing stage.

The conclusion will try to explain the "relativity" of the securing processes, to show why this a long way, without "diving" into futurology.

## EXAMPLE OF AN HETEROGENEOUS NETWORK

As a first advantage, the TCP/IP protocol is "spoken" by every OSes on earth. With it, very different systems are able to communicate with each other. Accordingly in the network we will use as an example, TCP/IP will always be present. In other words, we will not mention proprietary protocols, the least widespread nor the outdated ones. Neither will we talk about the physical structure, that is the type of connection, the category, etc.

So, in this network, we will put a bit of everything. Of course, we will find Unix, proprietary or free: for instance, a drop of Solaris 2.6, or SunOS 5.6, if you prefer, a drop of Irix 6.5, Linux (RH 6.2), MacOS X. We could have added a little bit of QNX or NeXTSTEP, or NetBSD or OpenBSD. On the "conventional" side we will include the one and lonely Not Terminated 4.0 (no, not any other, they are worse). Here too, we could have added OS2 which is less worse. Last, we will add a drop of "unconventional", let's say BeOS and AmigaOS (yes, it does exist... well, not much really!)

Of course, some of you are already complaining: what, no AIX, no HP-UX ? No! If we would like to mention every Unix, it would be a ten volumes article. However, the basic security rules are applicable to all the systems.

Now, what will we ask them ?

For example, let's say Solaris will be an applications server. Irix will manage the backups. NT will be another applications server. Linux will be a gateway. Another Linux box will be an http server or a database server. All the other machines are clients. We will consider that this network contains about 30

machines using password file authentication. We could have selected a more sophisticated authentication: NIS (Yellow Pages) or LDAP or Kerberos... Let's make things simple! Neither will we use NFS. Even if it can be helpful, when security is a concern, you better forget it, despite some improvement. In France, elderly people, used to say "don't put all your eggs in the same basket". Then, the "uncertain" but required, services or protocols will be present only once, on machines doing nothing else. For instance, only one ftp server, one http server, preferably on Unix machines. Some Unix machines will be SSH servers and the other ones will be SSH clients. Back on this later. We will use static IP addresses: no DHCP. In other words, we will stay basic! This of course this can be applied to a 50 machines network: with many more machines, it could become a nightmare.

## TOOLS AND HOW TO USE THEM

As usual, there is more than one way to do it (TIMTOWDI). The ideal case would be to start from scratch, with machines to install and network to setup. But this is only true in films! Accordingly, let's consider a network grown up over time, with machines moving from one place to another, new ones coming, and so on. Due to the Mhz "race", for instance, today Intel machines don't last long. After about 3 years, it becomes quite difficult to find spare parts. Thus, either you recycle the machines to subsidiary tasks or you get rid of them: sad but true! Fortunately, some others last much longer and deserve to be improved. Don't believe this is off topic: an administrator must work with high availability in mind. The basics

We could call "generalities" the first step of the job. It consists in removing everything useless on every machine: not a "light" task! Each OS, Unix included, installs an incredible number of services, protocols, that you will never use. The master word is: throw them away! Under Unix, a simple... and rough way is to comment out everything in /etc/inetd.conf. That makes a few services less. Of course, this is a bit exaggerated, but on many machines it is perfectly acceptable. It depends on your needs. Under Linux and a few others you can also use the chkconfig command to deactivate some services.

Also check the SUID/SGID files and don't hesitate in removing the "faulty" bit or consider deactivating the program. A command like: `find / -user root -a \( -perm -4000 -o -perm -2000 \) -print` will give you the list of those files. To remove the "s" bit, type `chmod a-s programname` (note: of course you loose some functionality by removing the "s" bit. It has its purpose after all).

Remove "dangerous" programs or the ones known as "risky": the remote commands such as rsh, rlogin, rcp... for instance. SSH will very well replace them.

Check the permissions for directories such as /etc, /var... The more restrictive the better. For instance, a

command such as `chmod -R 700` on the directory containing the startup files (`/etc/rc.d/init.d` on many Unixes) is not a bad idea. The same rule applies to all the systems being part of the network: remove what you don't use or, at least, deactivate. For NT, feel free to stop a maximum of services from the configuration panel. There are many basic "things" to do and there is a lot of literature on the subject out there. The tools

Let's begin with Unix, since it is the only one to really take security problems into account. Next, there is a huge quantity of free tools and most of them work on (almost) every Unix flavors.

For now, we will work on the individual machines since securing a network means, before everything, to secure its elements. Installing these tools is quite simple, that is why we will not spend time on the matter. Their parameters also depend on the systems, the needs... Up to you on how to apply this to your own case. The first required tool is called `shadow` utils. It is a means to do password encryption. Fortunately, it is part of many Unix distributions. The `/etc/shadow` file is then "created" from `/etc/passwd`.

Even better, PAM (Pluggable Authentication Modules) allows to restrict user access by service. Everything is managed from the directory containing the configuration files for each concerned service, usually `/etc/pam.d`. Many services can be PAM "driven", such as `ftp`, `login`, `xdm`, etc, allowing the administrator to choose who has right to do what.

The next tool is a must have: `TCPWrapper`. It also works on every Unix flavor or almost every. To make it short, it allows to restrict the access to services to some hosts. These hosts are allowed or denied using two files: `/etc/hosts.allow` and `/etc/hosts.deny`. `TCPWrapper` can be configured in two ways: either moving the daemons or changing the `/etc/inetd.conf` file. Later, we will see that `TCPWrapper` works fine in conjunction with other tools. You will find `TCPWrapper` at <ftp://ftp.porcupine.org/pub/security>

Another interesting tool is `xinetd`. Again, to make things short, `xinetd` is a replacement for `inetd` with much more features. According to what we above said about `inetd`, we will not insist. If you are interested, you will find it at <http://www.xinetd.org>.

Under Linux, there is one tool you can't live without: it is called Bastille-Linux. You will find it at <http://www.bastille-linux.org>. This tool, written in Perl, is not only didactic but also very efficient. After running a script, you answer many questions and Bastille-Linux acts accordingly. Every question is explained and default answers are provided. You can undo the changes, start a new configuration, check what has been done... Everything is there! It also offers a firewall configuration: back on this later. At the time of this writing, Bastille-Linux is at version 1.1.1, but the version 1.2.0 is already available as release candidate. It is much improved, and provides a GUI based on Tk and its Perl module. (Author's note: this article was written many months ago. As a

matter of fact, the present version of Bastille-Linux is 1.3.0).

Intrusion detection systems are also essential. The two "heavyweight" are called `snort` and `portsentry`. The first one can be downloaded from <http://www.snort.org> and the second one from the Abacus website, <http://www.psionic.com>. Those tools should not be compared: the first one is an NIDS (Network Intrusion Detection System) mainly providing with information, while the second one can be considered host oriented and more active. `snort` has a lot of options to supervise the network traffic. You can listen to everything you want: incoming, outgoing, inside the firewall, outside the firewall. Of course, it then can create huge logs, but you must know what you want! A Win 32 version is available, it is important if we consider the number of free tools available on these "systems".

`portsentry` has a very interesting feature: it can block the scanned ports according to your choice. Either you redirect the attacker to an unused address or you redirect to the firewall. Of course, you can select who to block and who not to block. Now we can go back to `TCPWrapper`: `portsentry` is able to write into the `/etc/hosts.deny` file if you want to. Thus, `portsentry` becomes quite efficient. We will not get into the debate about `portsentry` philosophy using port binding. It's up to you: make your choice after going deeper into the subject. Also be advised that `portsentry` can make a machine "invisible", what is not bad! Last, `portsentry` can use different operating modes, the most advanced being "reserved" for Linux (at least for now).

We cannot talk about security without mentioning encryption. However, the law about it, is different from one country to another and sometimes it is completely forbidden to use encryption.

Author's note: the following section has been removed from the English version of this article since it only concerns French law.

Conclusion: if your country allows encryption, install `ssh` clients and servers on your Unix machines (well, according to the needs!).

To finish with Unix tools, let's mention the ones belonging to proprietary Unixes. Under Solaris, you have `ndd`, `aset`; under Irix, you can use `ipfilterd`. MacOS X provides you with some free tools: `ssh`, `ipfwadm`...

Back on this later.

Now, let's talk about the one and lonely (fortunately!) Not Terminated 4.0. Here we cannot speak about free tools... however, the man from Redmond provides us with "free" stuff to improve the system features (it has nothing to do with bug corrections since there are no bugs!). Concerning security, NT 4.0 is a model... of absurdity. It's a bit like a sieve! Never mind. Accordingly, you just have to download the latest

service pack (6 at the time of this writing) and the HotFixes... which are security patches. Next... you can get some free tools (in the meaning of freely available and without the source code). That's all.

For other systems you will have to search. For AmigaOS, development doesn't seem to motivate much people and the TCP/IP layer is a bit old. However, Public Domain is still there to keep you busy. Concerning BeOS, things are not better: this great OS seems to have a very compromised future and the network layer called Bone is still in the works.

(Author's note: unfortunately, now BeOS is dead. A few people try to keep it alive as a free software product... and they do a very good job.) But there too, you will find some tools from the Unix world to improve things. Securing the hosts

Now, you will have to configure all this! Again, let's consider that every Unix machine is "equipped" with shadow-utils, PAM, TCPWrapper, that every useless service has been stopped or removed, that permissions have been hardened on the "sensitive" directories, etc.

On the Linux machines, it's time to launch Bastille-Linux. (This tool should work on most of the Linux distros, however, originally it has been designed for RedHat and Mandrake). Feel free to answer the questions in a very restrictive way.

On the Linux machine used as a gateway, the system must be "minimalist". You can remove most of the servers: http, ftp, etc. Remove X11 : you don't need it! Remove the not needed software... that is, almost everything. Stop the useless daemons. You should get a system where a ps ax command won't even fill the console screen. If you use IP Masquerading, the lsof -i command should display one line: the one concerning the listening server (we suppose that it is not a permanent connection).

Arbitrarily, we will install portsnentry on the Linux machines and it will be launched at startup time, using the "advanced" mode (reserved for Linux, that is with -atcp and -audp options). This implies that TCPWrapper and a firewall have been installed. Back on this later.

For Solaris, we will use the aset and ndd commands. More on this later too. portsnentry will be installed as well. We could add IP Filter and replace the standard version of RPCbind with version 2.1 available from porcupine.org. For Irix, we will choose ipfilterd for packet filtering as the name says. It is part of Irix distributions but it is not installed by default.

Concerning NT, things get a bit more complicated... The "fascist" solution consists in blocking ports 137 and 139, that is the famous NetBIOS (or even better removing NetBIOS)... but then no network is left (that is Windos network) it can be a small problem when it concerns an applications server! You can also install snort but it will not prevent those machines from being like sieves. Accordingly, you will have to be very restrictive about partition access, directory access...

as soon as you work with NTFS partitions, of course. There is a freely available program to get rid of the guest account but the source code is not available. Then, install all the security patches you can find! Last but not least, roll-up your sleeves and try to make that thing less vulnerable. It is a bit like to go round an assault course but it is compulsory.

For the "exotic" OSes, you will have to search and choose. As usual, and before all, the basic rules should be applied: the less active services, the better.

## PROTECTING THE NETWORK

If the hosts have been properly "prepared", you are half the way. But you will need to go further. Since we are talking about free software, we will choose a free firewall for the gateway: well, it is the machine allowing you to access the "wild" world. Arbitrarily (again!) we use a Linux box: so we can use the Bastille-Linux firewall. It works with ipchains or ipfwadm according your kernel version. If you use a 2.4 kernel, it will work with iptables.

A small digression: it is not a good idea to have all the initial problems to put up with, when security is a concern. The "race" to the latest kernel version may lead to a very negative situation. This does not mean that the work on new kernel is not a good, however, the "marriage" with existing tools, not designed to work that way can be a big mistake. An advice: be patient! The new firewall tool, part of the 2.4 kernel is very promising but probably a bit "young". That said, it is up to you...

So, the Bastille-Linux firewall is both simple and efficient. However, there is a much more elaborated tool, a bit like a "gas factory", called T.REX. It is available from <http://www.opensourcefirewall.com>. If you look for a very sophisticated free tool, here it is.

Other solutions exist, such as proxys, however they are not always better. Another digression: proxys are often called "firewalls". Nevertheless, they are two very different things. The firewalls we are talking about use packet filtering and do not provide authentication method. There are two types of proxy servers: applications or socks. In short, an application proxy does the job for you managing the entire communication and it allows for user authentication. This is why it needs much more resources than a firewall. But, again and again, this sort of tool only protects for a short lapse of time. A firewall can be "cracked" in about 15 minutes. Good to know, isn't it ? Hence the need to properly secure the hosts in your network: deciding to secure a network only relying on a firewall or a proxy is an heresy!

Another method to reduce the risks in a network is encryption. For example, using telnet is like making crackers walk on a red carpet. It is a way to give them the keys of the shop. Not only can they see the circulating data, but even better, that get the password in clear text: nice, isn't it ? Accordingly, feel



free to use ssh with the "uncertain" protocols (or instead of). If you MUST (?) use telnet, send the data through a secure connection. In other words, redirect the telnet port to a secure one. You will find more on this in the article titled "Through the tunnel" (LinuxFocus, May2001, article 202). (Free ads!)

OK, we tried to improve security, but now we should check our work. To do this, let's become "crackers", sort of: we will use their tools. Ugly, isn't it? In this area too, there is a nice collection of programs, then, again arbitrarily, we will choose two of them: nmap and nessus. There is no redundancy, since, for instance, the second one requires the first one. These tools are port scanners, even if nessus is much more than this. Nessus informs you about system vulnerabilities, comparing the scan results to its vulnerabilities database. Running these tools in a network will allow you to discover each host's weaknesses, whatever the OS is. The results are quite revealing thus making these tools a must have. You will find nmap at <http://www.insecure.org> and nessus at <http://www.nessus.org>

From the beginning of this article we are talking about securing a local network in which some machines are opened to the external world. An Internet Service Provider case obviously would be quite different and we will not get into the many details of the subject. Let's say that all we mentioned is still available but you will have to use much more elaborated methods, such as VPN (Virtual Private Network), LDAP for authentication (for example), etc. It is almost another subject since constraints are much more numerous according to the case. Let's not talk about e-business sites, where things are reckless. Secured sites they say! Don't tell me... Do you send your credit card number through the Internet? If yes, you are very courageous. Suggestion: if you can read French have a look at this website <http://www.kitetoa.com>, it is worth it.

## SYSTEMS PARTICULARITY

As already mentioned, systems are not equal when in front of the enemy. Some have very good abilities while others are sieves. Paradoxically (well, not really!), free OSes are among the better. The different BSD's (OpenBSD, NetBSD, FreeBSD...), the different Linuxes are quite ahead when security is a concern. Again, it is the result of the great work from the free software community. The others, even Unix labeled, are a bit less advanced. When they are not Unix, it is much worse!

All the tools mentioned in this article have been developed for free OSes. Most of the proprietary Unix systems can benefit from them. However, these proprietaries OSes often have their own tools. For instance, concerning Solaris, we mentioned ndd and aset. Despite a widespread idea, Sun systems are not security models. A tool such as aset, allows to improve things as far as access rights are concerned. aset offers three protection levels: low, medium and high. You can run it from a shell or from a cron task.

In a running network the situation changes, what was true at 5pm may become false at 5.30pm. Hence the interest to run commands periodically to keep some homogeneity. This is why aset has the ability to be cron managed. Thus, it will check every 30 minutes or every hour, or whatever you want, the permissions of directories, files...

ndd, allows to change the IP-stack parameters. For instance, it can be used to hide the system fingerprints. An identified system is a more vulnerable one, since the crackers know better where to "strike". With ndd, you can change the TCP Maximum Segment Size (MSS). By default, this size is 536 under Solaris 2.6. The `ndd -set /dev/tcp tcp_mss_def 546` command changes it to 546. The higher MSS is, the better (not too much!). Nmap, for instance is able to find this weakness. Using ndd, you cut the ground from under its feet. If you have machines running Solaris, feel free to use ndd. There are many options: check the man page.

You can also use IP Filter, a packet filtering tool. It is available from <ftp://coombs.anu.edu/pub/net/ip-filter>.

Concerning Irix, the situation is again different. SGI (ex Silicon Graphics), as the name says, designed its systems for graphics. Security was not the main concern. Necessity knowing no law, it became compulsory to provide ways to reduce the risks. ipfilterd was then provided in Irix distributions, but it is not installed by default: you will have to look for it! ipfilterd, is of course used for packet filtering, thus allowing to deny access to who you want. It relies on a configuration file called ipfilterd.conf and this is where things become a bit tricky. The syntax of this file is rather peculiar and does not like unexpected spaces or empty lines. Thus, to allow the machine called "mars" to talk to the machine called "jupiter" (which is the SGI workstation), you will have to type a line looking like:

```
accept -i ec0 between jupiter mars
```

The machines not listed in this file will not be able to access jupiter. Even worse: if you do not change the ipfilterd\_inactive\_behavior parameter using systune, nobody will access the machine! Efficient, isn't it? This parameter defaults to 1, and you will need to change it to 0 using the `systune -i ipfilterd_inactive_behavior 0` command.

Another well known thing, better to remind, Irix has a "great" vulnerability, called fam (File Alteration Monitor). This program is in charge of a very nice feature, the communication between various daemons. For example, it is the one allowing to get beautiful icons in the file manager. Nevertheless, there is only one thing to do: deactivate it! Sad, but it is like that.

To end with Unix systems, let's mention that QNX is very vulnerable but it can of course benefit from free tools. Mac OS X already provides some of these tools.

We must talk a bit of the absolute reference among network systems: the one and lonely NT 4.0. Securing that thing is an utopian view, despite what the King of Redmond (and many others) says. Simulating an attack with nessus, for instance, will be a nightmare. As far as NetBIOS is active, nessus will provide you with the names of every machine in the domain with their corresponding users, including administrators. The answer is: get rid of NetBIOS! Right, but as already mentioned, no NetBIOS, no network... You will have to choose your side.

nessus will kindly inform you that it can login as the guest user with a NULL session (that is with a NULL username and a NULL password). Remove it, then ! Yes, but how ? And it is all like that!

So, reduce the access to partitions (NTFS), to directories. For FAT partition... no solution. However, according to the software you use you may need FAT partitions: some software will not work on NTFS. To end with it, avoid the great IIS, especially as ftp server. In fact, don't install it. If today, so many ISP are mad enough to use that thing, we just can suggest them to use Apache instead, but... Don't we spend too much time on IIS, there is a lot of literature on the subject.

As a matter of fact, there is a way to make the sieve become a filter (holes are smaller!). The problem is that it is rather a long way and the whole magazine would not be enough. Let's only mention the most important. Obviously, the point is not to secure with free software: we are talking about the Microsoft world! The first suggestion is to use MSCE (Microsoft Security Configuration Editor) available from ServicePack 4 with MMC (Microsoft Management Console). However, be extremely careful! If you make a mistake, you have won. Of course, this software is an English version. If you use a foreign (not English) version of the system, be advised that the mixup of languages never gave very good results in the Redmond world. You have been warned. Next, among the required measures, you must "secure" the administrator account, or even deactivate it. Have a look at passprop available from the SP 3. You can also harden the passwords using the passfilt dll through the registry (I always thought that people who invented that thing were under LSD influence...). Deactivate the famous guest account. It is not very useful (see above), but it makes things less worse. But, you can restrict its access to the logs from the registry. In "HKEY\_LOCAL\_MACHINE", create the keys  
System\CurrentControlSet\Services\EventLog\Application, Security and System (these two last should replace Application). Their name is "RestrictGuestAccess", the type is REG\_SZ and the value is 1. You can encrypt the passwords with syskey. Careful, it is an irreversible operation! At least, some good news: you can restrict the guest access. Again, let's play with the registry, still in "HKEY\_LOCAL\_MACHINE". This time the key is called System\CurrentControlSet\Control\Lsa. The name is

"RestrictAnonymous", the type is "REG\_DWORD" and the value is 1. However, Microsoft world is a teaser: be advised that this change may alter some network services... Among the important things, you can restrict the access to some ports, using the Network application in the configuration panel. From the TCP/IP properties, select "Advanced" and check the "Activate security" box (I believe that this its name, but I don't have this kind of thing at home to be able to check). From the "Security" window, check "Allow only" and select the ports you want to activate. Here too, be careful. You should know what you are doing, otherwise some services will not work anymore.

A lot more can be done, but these are the essential. To learn more, you can visit sans.org: tons of documents are available.

## THE UNBEARABLE LIGHTNESS OF THINGS

Well, you have done all this. You run nessus to scan the whole network and you still get security holes. We will not say where they come from... we already know! Try to delude these system substitutes. It will not remove the holes "provided" by NetBIOS, but it will limit the damage. Create subdomains. Don't login as an administrator. Apply patches. Last, try to hide all this behind Unix machines used as gateways. Unfortunately, the relativity of security doesn't only come from products made in Redmond. A network is alive: there is always something going on. A good administrator is a "paranoid" one, accordingly, often check the "inventory of fixtures". Write scripts to automate the checks. For instance to control on a regular basis the SUID/SGID programs, the critical files, the logs... To get a few more friends, lock the users floppy or CDROM devices. Don't accept that users download software without your agreement, especially when this software is executable like always in the Microsoft world. Prevent your users from opening attached documents like those in Word or Excel format using a mail filtering system. Yes, I know it is like fascism, but what can you do against macro-viruses ? Do not use products such as Outlook. Once again, you must know what you want! I know, what I say is useless, but can you talk about security with such products ? The famous "I love you" did not teach any lesson.

Concerning Unix, downloads must be controlled as well. Checksums have not been provided by accident.

Get the habit of controlling your network on a regular basis with logs, scripts, scans... You will notice: things change quite fast and not only in the good way. Last, we did not say a word about it, but don't forget backups. The strategy is unchanging: daily, weekly and monthly. An Unix machine can also have problems, even if it is unusual. And, sometimes the users make mistakes... but not very often. It is well known that the problems come from the machines or from the department in charge of them:-)

## AT LEAST, IT IS OVER!

If you reached this section is that you are courageous. The problem is that we only skimmed over the subject! Security has no end and doesn't only concern networks. Vulnerable applications can compromise a network. A badly configured firewall is far more dangerous than no firewall at all. An Unix machine often holds thousands of files. Who can be sure that none of them is vulnerable ? Who thinks a cracker will try to break a 128 bits key ? Don't be fooled: he will try to find a door behind the house. Again and again, you can install all the security tools available, if you leave a very small hole, this is where the "bad" will go through.

Security is also a behavior: follow what is going on. For example, visit the security websites on a regular basis, same for the websites of your OSes editors... For example, Sun publishes recommended patches every month. SGI releases a new Irix version every three months. Microsoft frequently provides ServicePacks or HotFixes. Linux distributors publish erratas for each newly discovered vulnerability. Same for the different BSD's. If you don't use the products corresponding to a patch remove them from your hard disk. And so on: the list of things to be done is a very, very long one. In short, this job should not know lay-offs. Last, let's say it again, all this will only contribute to make your network a bit less vulnerable. Don't expect you will get a 100% secure network, even at a given time (well, may be, if all the machines are stopped). That said, it is not a requirement to be paranoid to do this job... but it helps ! But don't be like like that in your everyday life, it will be much nicer for the people around you...

## REFERENCES

<http://www.linuxsecurity.com>  
<http://www.sans.org>  
<http://www.infosyssec.org>  
<http://www.securityfocus.com>  
<http://www.cs.purdue.edu/coast/hotlist/>

Life is sad: let's have some fun!

*This article is re-printed with permission. The originals can be found at:*

<http://www.linuxfocus.org/English/July2002/article245.shtml>

# Graphics Programming with libtiff

Author: Michael Still <[mikal@stillhq.com](mailto:mikal@stillhq.com)>

## ABSTRACT

TIFF is an extremely common, but quite complex raster image format. Libtiff is a standard implementation of the TIFF specification, which is free and works on many operating systems. This article discusses some of the pitfalls of TIFF, and guides the reader through use of the libtiff library. This article provides examples on how to use libtiff for your black and white imaging needs.

TIFF (Tagged Image File Format) is a raster image format which was originally produced by Adobe. Raster image formats are those which store the picture as a bitmap describing the state of pixels, as opposed to recording the length and locations of primitives such as lines and curves. Libtiff is one of the standard implementations of the TIFF specification, and is in wide use today because of its speed, power and easy source availability.

This article focuses on black and white TIFF images, there isn't enough space in the article to cover color images as well. These will be covered in another article in a later edition of DeveloperWorks.

## CODING FOR TIFF CAN BE HARD

Most file format specifications define some basic rules for the representation of the file. For instance, PNG (a compeditor to TIFF) documents are always big endian. TIFF doesn't mandate things like this though, here is a list of some of the seemingly basic things that it doesn't define:

1. The byte order -- big endian, or little endian
2. The fill order of the bit within the image bytes -- most significant bit first, or least significant
3. The meaning of a given pixel value for black and white -- is 0 black, or white?
4. ...and so on

This means that creating a TIFF can be very easy, because it is rare to have to do any conversion of the data that you already have. It does mean, on the other hand, that being able to read in random tiffs created by other applications can be very hard -- you have to code for all these possible combinations in order to be reasonably certain of having a reliable product.

So how do you write an application which can read in all these different possible permutations of the TIFF format? The most important thing to remember is to never make assumptions about the format of the image data you are reading in.

## WRITING TIFF FILES

The first thing I want to do is show you how to write a TIFF file out. We'll then get onto how to read a TIFF file back into your program.

### Infrastructure for writing

It is traditional for bitmaps to be represented inside your code with an array of chars. This is because on most operating systems, a char maps well to one byte. In the block of code below, we will setup libtiff, and create a simple buffer which contains an image which we can then write out to disc.

```
#include <stdio.h>
#include <tiffio.h>

int
main (int argc, char *argv[])
{
    char buffer[32 * 9];
}
```

The code above is pretty simple. All you need to use libtiff is to include the tiffio.h header file. To compile this, use the command `gcc foo.c -o foo -ltiff -lm`. The `-ltiff` is a command which will include the library named libtiff, which needs to be in your library path. Once you have started specifying libraries explicitly, you also need to add `-lm`, which is the mathematics library. The char buffer that we have defined here is going to be our black and white image, so we should define one of those next...

### Writing the image

To make up for how boring that example was, I am now pleased to present you with possibly the worst picture of the Sydney Harbour Bridge ever drawn. In the example below, the image is already in the image buffer, and all we have to do is save it to the file on disc. The example first opens a tiff image in write mode, and then places the image into that file.

Please note, that for clarity I have omitted the actual hex for the image, this is available in the download version of this code for those who are interested.

```
#include <stdio.h>
#include <tiffio.h>

int main(int argc, char *argv[]){
    // Define an image
    char buffer[25 * 144] = { /* boring hex omitted
    */ };
    TIFF *image;

    // Open the TIFF file
    if((image = TIFFOpen("output.tif", "w")) ==
    NULL){
        printf("Could not open output.tif for
        writing\n");
        exit(42);
    }

    // We need to set some values for basic tags
    before we can add any data
    TIFFSetField(image, TIFFTAG_IMAGEWIDTH, 25 * 8);
    TIFFSetField(image, TIFFTAG_IMAGELENGTH, 144);
    TIFFSetField(image, TIFFTAG_BITSPERSAMPLE, 1);
```

```
    TIFFSetField(image, TIFFTAG_SAMPLESPERPIXEL, 1);
    TIFFSetField(image, TIFFTAG_ROWSPERSTRIP, 144);

    TIFFSetField(image, TIFFTAG_COMPRESSION,
    COMPRESSION_CCITTFAX4);
    TIFFSetField(image, TIFFTAG_PHOTOMETRIC,
    PHOTOMETRIC_MINISWHITE);
    TIFFSetField(image, TIFFTAG_FILLOORDER,
    FILLOORDER_MSB2LSB);
    TIFFSetField(image, TIFFTAG_PLANARCONFIG,
    PLANARCONFIG_CONTIG);

    TIFFSetField(image, TIFFTAG_XRESOLUTION, 150.0);
    TIFFSetField(image, TIFFTAG_YRESOLUTION, 150.0);
    TIFFSetField(image, TIFFTAG_RESOLUTIONUNIT,
    RESUNIT_INCH);

    // Write the information to the file
    TIFFWriteEncodedStrip(image, 0, buffer, 25 *
    144);

    // Close the file
    TIFFClose(image);
}
```

There are some interesting things to note in this example. The most interesting of these is that the output image will not display using the `xview` command on my linux machine. In fact, I couldn't find an example of a group 4 fax compressed black and white image which would display using that program. See the sidebar for more detail.

### Problems with xview

Xview is part of the `xloadimage` package written by Jim Frost, which comes with X windows.

It's a good example of how hard it can be to handle TIFF images well. I am currently working on a patch to submit to Jim which will resolve this problem. If you have trouble viewing the output of the sample code, then try using some other program, like the `gimp`.

The sample code shows the basics of using the libtiff API. The following interesting points should be noted...

1. The buffers presented to and returned from libtiff each contain 8 pixels in a single byte. This means that you have to be able to extract the pixels you are interested in. The use of masks, and the right and left shift operators come in handy here.
2. The `TIFFOpen` function is very similar to the `fopen` function we are all familiar with.
3. We need to set the value for quite a few fields before we can start writing the image out. These fields give libtiff information about the size and shape of the image, as well as the way that data will be compressed within the image. These fields need to be set before you can start handing image data to libtiff. There are many more fields for which a value could be set, I have used close to the bar minimum in this example.
4. `TIFFWriteEncodedStrip` is the function call which actually inserts the image into the file. This call inserts uncompressed image data into the file. This means that libtiff will compress the image data for you before writing it to the file. If you have already compressed data, then have a look at the

TIFFWriteRawStrip instead.

5. Finally, we close the file with TIFFClose.

---

**More information about the libtiff function calls**

---

If you need more information about any of the libtiff function calls mentioned in this article, then checkout the extensive man pages which come with the library. Remember that case is important with man pages, so you need to get the case in the function names right -- it's TIFFOpen, not tiffopen.

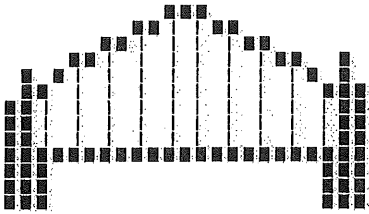


Figure 1. Figure 1. The Sydney Harbour Bridge, by Michael Still

## READING TIFF FILES

Reading TIFF files reliably is much harder than writing them. Unfortunately, I don't have enough space in this article to discuss all of the important issues. Some of them will need to be left to later articles. There are also plenty of pages on the web which discuss the issues involved. Some of my favourites are included in the references section at the end of this article.

The issue that complicates reading black and white TIFF images the most is the several different storage schemes which are possible within the TIFF file itself. libtiff doesn't hold your hand much with these schemes, so you have to be able to handle them yourself. The three schemes TIFF supports are single striped images, stripped images, and tiled images.

1. A single strip image is as the name suggests -- a special case of a stripped image. In this case, all of the bitmap is stored in one large block. I have experienced reliability issues with images which are single strip on Windows machines. The general recommendation is that no one strip should take more than 8 kilobytes uncompressed which with black and white images limits us to 65,536 pixels in a single strip.
2. A multiple strip image is where horizontal blocks of the image are stored together. More than one strip is joined vertically to make the entire bitmap. Figure 2 shows this concept.
3. A tiled image is like your bathroom wall, it is composed of tiles. This representation is show in Figure 3, and is useful for extremely large images -- this is especially true when you might only want to manipulate a small portion of the image at any

one time.

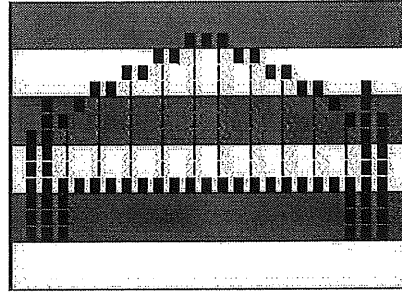


Figure 2. Figure 2. The Sydney Harbour Bridge, in strips

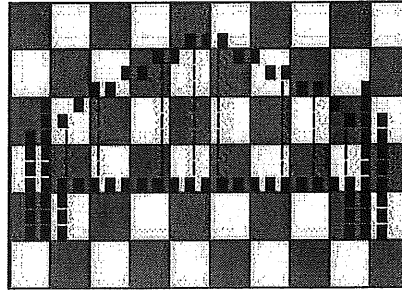


Figure 3. Figure 3. The Sydney Harbour Bridge, in tiles

Tiled images are comparatively uncommon, so I will focus on striped images in this article. Remember as we go along, that the single striped case is merely a subset of a multiple strip images.

---

**Infrastructure for reading**

---

The most important thing to remember when reading in TIFF images is to be flexible. The example below has the same basic concepts as the writing example above, with the major difference being that it needs to deal with many possible input images. Apart from stripping and tiling, the most important thing to remember to be flexible about is photometric interpretation. Luckily, with black and white images there are only two photometric interpretations to worry about (with colour and to a certain extent grayscale images there are many more).

What is photometric interpretation? Well, the representation of the image in the buffer is really a very arbitrary thing. I might code my bitmaps so that 0 means black (TIFFTAG\_MINISBLACK), whilst you might find black being 1 (TIFFTAG\_MINISWHITE) more convenient. TIFF allows both, so our code has to be able to handle both cases. In the example below, I have assumed that the internal buffers need to be in MINISWHITE, so we will convert images which are in MINISBLACK.

The other big thing to bear in mind is fillorder (whether the first bit in the byte is the highest value, or the lowest). The example below also handles both of these correctly. I have assumed that we want the buffer to have the most significant bit first. TIFF images can be either big endian or little endian, but libtiff handles this for us. Thankfully, libtiff also supports the various compression algorithms without



you having to worry about those. These are by far the scariest area of TIFF, so it is still worth your time to use libtiff.

```
#include <stdio.h>
#include <tiffio.h>

int main(int argc, char *argv[]){
    TIFF *image;
    uint16 photo, bps, spp, fillorder;
    uint32 width;
    tsize_t stripSize;
    unsigned long imageOffset, result;
    int stripMax, stripCount;
    char *buffer, tempbyte;
    unsigned long bufferSize, count;

    // Open the TIFF image
    if((image = TIFFOpen(argv[1], "r")) == NULL){
        fprintf(stderr, "Could not open incoming
image\n");
        exit(42);
    }

    // Check that it is of a type that we support
    if((TIFFGetField(image, TIFFTAG_BITSPERSAMPLE,
&bps) == 0) || (bps != 1)){
        fprintf(stderr, "Either undefined or
unsupported number of bits per sample\n");
        exit(42);
    }

    if((TIFFGetField(image, TIFFTAG_SAMPLESPERPIXEL,
&spp) == 0) || (spp != 1)){
        fprintf(stderr, "Either undefined or
unsupported number of samples per pixel\n");
        exit(42);
    }

    // Read in the possibly multile strips
    stripSize = TIFFStripSize (image);
    stripMax = TIFFNumberOfStrips (image);
    imageOffset = 0;

    bufferSize = TIFFNumberOfStrips (image) *
stripSize;
    if((buffer = (char *) malloc(bufferSize)) ==
NULL){
        fprintf(stderr, "Could not allocate enough
memory for the uncompressed image\n");
        exit(42);
    }

    for (stripCount = 0; stripCount < stripMax;
stripCount++){
        if((result = TIFFReadEncodedStrip (image,
stripCount,
                                buffer +
imageOffset,
                                stripSize))
== -1){
            fprintf(stderr, "Read error on input strip
number %d\n", stripCount);
            exit(42);
        }

        imageOffset += result;
    }

    // Deal with photometric interpretations
    if(TIFFGetField(image, TIFFTAG_PHOTOMETRIC,
&photo) == 0){
        fprintf(stderr, "Image has an undefined
photometric interpretation\n");
        exit(42);
    }

    if(photo != PHOTOMETRIC_MINISWHITE){
        // Flip bits
        printf("Fixing the photometric
interpretation\n");

        for(count = 0; count < bufferSize; count++)
```

```
        buffer[count] = ~buffer[count];
    }

    // Deal with fillorder
    if(TIFFGetField(image, TIFFTAG_FILLORDER,
&fillorder) == 0){
        fprintf(stderr, "Image has an undefined
fillorder\n");
        exit(42);
    }

    if(fillorder != FILLORDER_MSB2LSB){
        // We need to swap bits -- ABCDEFGH becomes
HGFEDCBA
        printf("Fixing the fillorder\n");

        for(count = 0; count < bufferSize; count++){
            tempbyte = 0;
            if(buffer[count] & 128) tempbyte += 1;
            if(buffer[count] & 64) tempbyte += 2;
            if(buffer[count] & 32) tempbyte += 4;
            if(buffer[count] & 16) tempbyte += 8;
            if(buffer[count] & 8) tempbyte += 16;
            if(buffer[count] & 4) tempbyte += 32;
            if(buffer[count] & 2) tempbyte += 64;
            if(buffer[count] & 1) tempbyte += 128;
            buffer[count] = tempbyte;
        }

        // Do whatever it is we do with the buffer -- we
dump it in hex
        if(TIFFGetField(image, TIFFTAG_IMAGEWIDTH,
&width) == 0){
            fprintf(stderr, "Image does not define its
width\n");
            exit(42);
        }

        for(count = 0; count < bufferSize; count++){
            printf("%02x", (unsigned char) buffer[count]);
            if((count + 1) % (width / 8) == 0)
printf("\n");
            else printf(" ");
        }

        TIFFClose(image);
    }
}
```

This code works by first opening the image and checking that it is one that we can handle. It then reads in all the strip for the image, and appends them together in one large memory block. If required, it also flips bits until the photometric interpretation the one we handle, and deals with having to swap bits if the fillorder is wrong. Finally, our sample outputs the image as a series of lines composed of hex values. Remember that each of the values represents 8 pixels in the actual image.

## CONCLUSION

In this article I have shown you how to write and read some simple black and white images using libtiff. There are of course more issues that can be dealt with to have the perfect code, but being aware of the issues is the first step. Finally, before you leap off and start coding with libtiff, remember to put some thought into what compression algorithm you should be using for your images -- group 4 fax is great for black and white, but what you use for color really depends on your needs.

## RESOURCE LIST



1. The libtiff website (<http://www.libtiff.org>) is a good place to download the libtiff source. It is also quite likely there is a binary package for your chosen operating system.
2. If all else fails, then the Adobe TIFF Specification (<http://partners.adobe.com/asn/developer/pdfs/tiff/TIFF6.pdf>) can be useful.
3. The xloadimage web page (<http://gopher.std.com/homepages/jimf/xloadimage.html>) might be of interest.
4. The Cooper Union for the Advancement of Science and Art has some notes ([http://www.ee.cooper.edu/courses/course\\_pages/past\\_courses/EE458/TIFF/](http://www.ee.cooper.edu/courses/course_pages/past_courses/EE458/TIFF/)) from a previous course dealing with libtiff online.

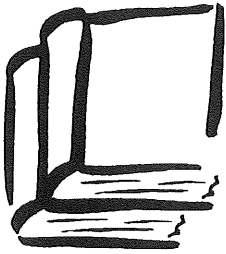
#### ABOUT THE AUTHOR

Michael has been working in the image processing field for several years, including a couple of years

managing and developing large image databases for an Australian government department. He currently works for TOWER Software, who manufacture a world leading EDMS and Records Management package named TRIM. Michael is also the developer of Panda, an open source PDF generation API, as well as being the maintainer of the `comp.text.pdf` USENET frequently asked questions document. You can contact Michael at [mikal@stillhq.com](mailto:mikal@stillhq.com).

*First published by IBM developerWorks at <http://www.ibm.com/developerworks/>*

---



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

**AMERICAN  
BOOK STORE**

173 Elizabeth St, Brisbane Queensland 4000

Ph: (07) 3229 4677 Fax: (07) 3221 2171 Qld Country Freecall: 1800 177 395

american\_bookstore@compuserve.com

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

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

Post Code: \_\_\_\_\_

Phone Number: \_\_\_\_\_

Payment Method:     Cheque         Money Order     Amex         Bankcard  
                          Diners         Mastercard     Visa

Card Number: \_\_\_\_\_

Expiry Date: \_\_\_\_\_ Signature: \_\_\_\_\_

This is a:     Special Order     Mail Order     Book on Hold

QUANTITY	TITLE	PRICE
.....	.....	.....
.....	.....	.....
.....	.....	.....
.....	.....	.....
.....	.....	.....
.....	.....	.....
.....	.....	.....
.....	.....	.....
.....	.....	.....
.....	.....	.....

SUBTOTAL	\$ .....
LESS 10% DISCOUNT	\$ .....
POST & PACK	\$ .....
TOTAL	\$ .....

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

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

# AUUG Chapter Meetings and Contact Details

CITY	LOCATION	OTHER
<b>ADELAIDE</b>	We meet at IBM in 180 Greenhill Road, Parkside, at 7 pm on the second Wednesday of each month.	Contact sa-exec@auug.org.au for further details.
<b>BRISBANE</b>	Inn on the Park 507 Coronation Drive Toowong	For further information, contact the QAUUG Executive Committee via email (qauug-exec@auug.org.au). The technologically 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.
<b>CANBERRA</b>	Australian National University	
<b>HOBART</b>	University of Tasmania	
<b>MELBOURNE</b>	Various. For updated information See:  <a href="http://www.vic.auug.org.au/auugvic/av_meetings.html">http://www.vic.auug.org.au/auugvic/av_meetings.html</a>	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.
<b>PERTH</b>	The Victoria League 276 Onslow Road Shenton Park	
<b>SYDNEY</b>	TBA	

FOR UP-TO-DATE DETAILS ON CHAPTERS AND MEETINGS, INCLUDING THOSE IN ALL OTHER AUSTRALIAN CITIES, PLEASE CHECK THE AUUG WEBSITE AT [HTTP://WWW.AUUG.ORG.AU](http://www.auug.org.au) OR CALL THE AUUG OFFICE ON 1-800-625655.



# Application for Institutional Membership

## Section A: MEMBER DETAILS

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

NAME OF ORGANISATION: \_\_\_\_\_

### Primary Contact

Surname \_\_\_\_\_ First Name \_\_\_\_\_  
 Title: \_\_\_\_\_ Position \_\_\_\_\_  
 Address \_\_\_\_\_  
 Suburb \_\_\_\_\_ State \_\_\_\_\_ Postcode \_\_\_\_\_  
 Telephone: Business \_\_\_\_\_ Facsimile \_\_\_\_\_  
 Email \_\_\_\_\_ Local Chapter Preference \_\_\_\_\_

## Section B: MEMBERSHIP INFORMATION.

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

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

## Section C: PAYMENT

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

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

-OR-

Please debit my credit card for AS \_\_\_\_\_  
 Bankcard  Visa  Mastercard

Name on Card \_\_\_\_\_  
 Card Number \_\_\_\_\_  
 Expiry Date \_\_\_\_\_  
 Signature \_\_\_\_\_

Please mail completed form with payment to: \_\_\_\_\_ Or Fax to: \_\_\_\_\_

Reply Paid 66 AUUG Inc  
 AUUG Membership Secretary (02) 8824 9522  
 PO Box 366  
 KENSINGTON NSW 2033

## Section D: MAILING LISTS

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

Yes  No

## Section E: AGREEMENT

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

*I/we understand that I/we will receive two copies of the AUUG newsletter, and may send two representatives to AUUG sponsored events at member rates, though I/we will have only one vote in AUUG elections, and other ballots as required.*

Signed: \_\_\_\_\_  
 Title: \_\_\_\_\_  
 Date: \_\_\_\_\_

### AUUG Secretariat Use

Chq: bank \_\_\_\_\_ bsb \_\_\_\_\_  
 A/C: \_\_\_\_\_ # \_\_\_\_\_  
 Date: \_\_\_\_\_ \$ \_\_\_\_\_  
 Initial: \_\_\_\_\_ Date Processed: \_\_\_\_\_  
 Membership#: \_\_\_\_\_



UNIX® AND OPEN SYSTEMS USERS

# Membership Application

**AUUG Inc**  
 PO Box 366, Kensington NSW 2033, Australia  
 Tel: (02) 8824 9511  
 Free Call: 1 800 625 655  
 Fax: (02) 8824 9522  
 email: auug@auug.org.au  
 ACN A00 166 36N (incorporated in Victoria)

<http://www.auug.org.au>

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

**AUUG OFFERS SOMETHING FOR YOU!!**

**Education**

Tutorials  
Workshops

**AUUGN**

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

**Events.....Events.....Events**

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

**DISCOUNTS**

to all AUUG events and education.  
Reciprocal arrangements with overseas affiliates.  
Discounts with various internet service providers, software, publications and more...!!

**Connections**

- Newsgroup [aus.org.auug](http://aus.org.auug)

# Application for Individual or Student Membership

**Section A: PERSONAL DETAILS**

Surname \_\_\_\_\_ First Name \_\_\_\_\_  
 Title: \_\_\_\_\_ Position \_\_\_\_\_  
 Organisation \_\_\_\_\_  
 Address \_\_\_\_\_  
 Suburb \_\_\_\_\_ State \_\_\_\_\_ Postcode \_\_\_\_\_  
 Telephone: Business \_\_\_\_\_ Private \_\_\_\_\_  
 Facsimile: \_\_\_\_\_ E-mail \_\_\_\_\_

**Section B: MEMBERSHIP INFORMATION**

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

**RENEWAL/NEW INDIVIDUAL MEMBERSHIP**  
 Renewal/New Membership of AUUG  \$110.00

**RENEWAL/NEW STUDENT MEMBERSHIP**  
 Renewal/New Membership of AUUG  \$27.50  
*(Please complete Section C)*

SURCHARGE FOR INTERNATIONAL AIR MAIL  \$66.00

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

**Section C: STUDENT MEMBER CERTIFICATION**

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

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

NAME OF STUDENT: \_\_\_\_\_  
 INSTITUTION: \_\_\_\_\_  
 STUDENT NUMBER: \_\_\_\_\_  
 SIGNED: \_\_\_\_\_  
 NAME: \_\_\_\_\_  
 TITLE: \_\_\_\_\_  
 DATE: \_\_\_\_\_

**Section D: LOCAL CHAPTER PREFERENCE**

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

**Section E: MAILING LISTS**

AUUG mailing lists are sometimes made available to vendors. Please indicate whether you wish your name to be included on these lists:  
 Yes  No

**Section F: PAYMENT**

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

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

-OR-

Please debit my credit card for AS \_\_\_\_\_  
 Bankcard  Visa  Mastercard

Name on Card \_\_\_\_\_  
 Card Number \_\_\_\_\_  
 Expiry Date \_\_\_\_\_  
 Signature \_\_\_\_\_

Please mail completed form with payment to:      Or Fax to:

Reply Paid 66      AUUG Inc  
 AUUG Membership Secretary      (02) 8824 9522  
 PO Box 366  
 KENSINGTON NSW 2033  
 AUSTRALIA

**Section G: AGREEMENT**

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

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

**AUUG Secretariat Use**

Chq: bank \_\_\_\_\_ bsb \_\_\_\_\_  
 A/C: \_\_\_\_\_ # \_\_\_\_\_  
 Date: \_\_\_\_\_ \$ \_\_\_\_\_  
 Initial: \_\_\_\_\_ Date Processed: \_\_\_\_\_  
 Membership#: \_\_\_\_\_