

AUUGN

The Journal of AUUG Inc.

Volume 25 • Number 3
September 2004

Features:

<i>mychart</i> Charting System for Recreational Boats	8
Lions Commentary, part 1	17
Managing Debian	24
SEQUENT: Asynchronous Distributed Data Exchange Framework	51

News:

Minutes to AUUG board meeting of 5 May 2004	13
Liberal license for ancient UNIX sources	16
First Australian UNIX Developer's Symposium: CFP	60
First Digital Pest Symposium	61

Regulars:

Editorial	1
President's Column	3
About AUUGN	4
My Home Network	5
AUUG Corporate Members	23
A Hacker's Diary	29
Letters to AUUG	58
Chapter Meetings and Contact Details	62
AUUG Membership Application Form	63



AUUGN

The journal of AUUG Inc.

Volume 25, Number 3
September 2004

Editorial

Greg Lehey <Greg.Lehy@auug.org.au>

After last quarter's spectacularly late delivery of AUUGN, things are gradually getting back to normal. I had hoped to have this on your desk by the end of September, but it wasn't to be. Given that that was only a couple of weeks after the "July" edition, this doesn't seem to be such a problem. I'm fully expecting to get the December issue out in time to keep you from boredom over the Christmas break.

First the good news: last quarter I asked for volunteers for an editorial team. I'm pleased to announce that Frank Crawford has risen to the challenge and is now part of the team. As a result, I'm pleased to say that I'll be staying on the team too.

Only one problem with Frank: he's the most reliable person I know in AUUG. When things go wrong, you can rely on Frank to help out. There should be more like him: but there aren't. And that's my worry. We can do with more people on the team; just because there are now two of us doesn't mean that we can't do with more.

What feedback we've had about the change of subject matter has generally been well received. In this issue, I'm continuing the idea of reprinting AUUG's own content. It hasn't yet been possible to include any of the papers from this year's conference in this issue, but we're confident to get some of the better ones in the December issue. In the meantime, I've dug up some "oldies but goodies" for this issue.

A little over ten years ago, when people still read USENET news, I came into my office (in Germany) one day to find two cryptic messages sent to `alt.folklore.computers`. They were uuencoded and bore the subject line "Leo's notes". Even while I was uudecoding the messages, I thought "Let it be the Lions book!". And it was, somewhat incongruously reformatted in \TeX .

For those newcomers who don't recall the "Lions Book", this is the "Commentary on the Sixth Edition UNIX Operating System" that John Lions wrote for classes at UNSW back in 1977. Supposedly they were the most-photocopied of all UNIX-related documents. I had mislaid my photocopy, poor as it was (weren't they all?) some time earlier, so I was delighted to have an easy to read version.

As Eric Raymond says, the thing that withheld the publication of the notes at the time was the trade secret status of the UNIX kernel. That changed with Caldera's release of "Ancient UNIX" on 20 January 2002, and Peter Salus, who has full rights to reprint, has agreed to let us reprint them. As a result, we can now print the notes (and convert them back into John Lions' beloved *troff*; see page 20). Due to the size, we'll be spreading it over the next few issues.

The posting on USENET didn't include the source code which was an integral part of the notes. I don't intend to print them here either; nowadays it's a lot easier to have that sort of thing online. I'll be including them in the next AUUG CD-R.

About CD-Rs: as I stated last month, due to a comedy (for want of a more appropriate word) of errors, we ended up including one more CD-R than intended in the last issue of AUUGN. To make up for that, this issue doesn't contain any. We'll resume the usual CD-R per quarter next quarter, and after that there will be *only* the CD-R: this is the second-last AUUGN on paper. From next year, the content will be on CD-R instead. If this is your first issue, and you'd like a CD-R, please contact Liz Carroll, our Executive Director (see page 2), and she will get you one.

AUUG Membership and General Correspondence

The AUUG Secretariat

AUUG Inc
PO Box 7071
Baulkham Hills BC NSW 2153
Telephone: 02 8824 9511
or 1800 625 655 (Toll-Free)
Facsimile: 02 8824 9522
Email: auug@auug.org.au

AUUG Executive Director

Elizabeth Carroll
AUUG Inc
PO Box 7071
Baulkham Hills BC NSW 2153
liz@auug.org.au

AUUG Board of Directors

Email: auugexec@auug.org.au

President

David Purdue
Sun Microsystems
Level 6, 476 St Kilda Road
Melbourne, VIC 3004
Phone: +61 3 9869 6412,
Fax: +61 3 9869 6288
David.Purdue@auug.org.au

Vice-president

Steve Landers
Digital Smarties
PO Box 717 Willetton WA 6155
Business phone: +61 8 9313 6868
Business fax: +61 8 9313 6077
Steve.Landers@auug.org.au

Secretary

Jonathon Coombes
Cybersite Consulting Pty Ltd
34 Newcastle Road, Wallsend NSW 2287
Business Telephone: +61 2 4965 6989
Jonathon.Coombes@auug.org.au

Treasurer

Gordon Hubbard
Custom Technology Australia Pty Ltd
3 Spring Street,
Bus Tel: 02 9659 9590,
Bus Fax: 02 9659 9510
Gordon.Hubbard@auug.org.au

Ordinary board members

Grant Allen

Adrian Close
Cybersource Pty.Ltd.
4, 10 Queen Street Melbourne VIC 3000
Business Telephone: +61 3 9621 2377
Business Fax: +61 3 9621 2377
Mobile: +61 417 346 094
adrian@auug.org.au

Andrew Frederick Cowie
Operational Dynamics
GPO Box 4339
Sydney, NSW, 2001
Telephone: +61-2-9977-6866
Andrew.Cowie@auug.org.au

Enno Davids

Michael Still
Phone: +61 414 382 568
mikal@auug.org.au

Immediate past president:

Greg Lehey
PO Box 460
Echunga, SA, 5153
Bus. Tel (08) 8388 8286,
Mobile 0418 838 708,
Fax (08) 8388 8725
Greg.Lehey@auug.org.au

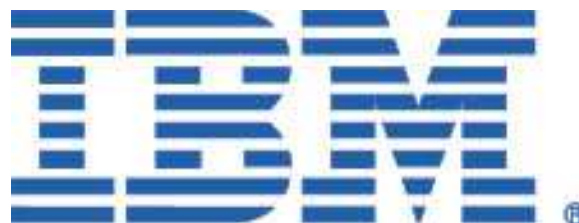
Returning officer:

Jason Ozolins

Assistant Returning officer:

David Baldwin

AUUG Incorporated gratefully acknowledges the support of its corporate sponsor:



President's Column

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

“The reports of my death are greatly exaggerated”

The text of a cable sent by Mark Twain from London to the press in the United States after his obituary had been mistakenly published.

*—The New Dictionary of Cultural Literacy,
Third Edition*

You may have read recently some dire predictions regarding the future of AUUG. However, following the success of our recent conference, AUUG'2004, I can safely say that those predictions were misguided at best, misleading at worst.

Interest in AUUG is high—we have over 800 people subscribed to auug-announce, and there has been a lot of press coverage for the conference, especially the presentations of the four main parties on their approaches to open computing in government—the culmination of a process AUUG kicked off two years ago.

The conference was successful in every respect. Using the three main criteria we use for measuring success—attendance, revenue and profit—it was our most successful conference for about 5 years. AUUG has also received a lot of very positive feedback on the quality of the conference—from attendees, speakers and sponsors. This includes a couple of attendees who have given presents to Liz for making the conference experience easier—of course, Liz did not tell me about this until after she had gone home, so she didn't have to share the chocolates!

This is not to say that AUUG does not face some challenges. But we have a board in place that is actively addressing these challenges and coming up with workable solutions.

We are focussing on what AUUG is. Who are our members and what can AUUG, as a volunteer organisation, deliver to support our members in their professional activities? By the time this AUUGN appears, the board will have voted on a new tag line and statement of purpose, and that should be on the front page of the web site (<http://www.auug.org.au/>). This statement will be the launching point for ideas for the development and expansion of AUUG, and a way of assessing whether new ideas are worth pursuing.

AUUG's main value is as a forum for professionals interested in standards based computing to exchange ideas—to help each other out. Many of

the activities we are proposing are aimed at fostering community. For example, we are continuing to expand the series of one-day symposia that we have been running for several years. To stay up to date with these developments, make sure you are subscribed to auug-announce (go to <http://www.auug.org.au/mailman/listinfo/auug-announce> to subscribe or check your subscription options).

So now is a good time to pipe up about what you think AUUG is and where it should go. AUUG is a volunteer organisation—I encourage you all to participate, even if it is just turning up to your local chapter meetings. If you have ideas, comments, criticisms or a wish to get involved and just do something, please contact the AUUG board at <auugexec@auug.org.au>

Finally, it is my great pleasure to announce that AUUG has promoted Liz Carroll to the position of Executive Director. In practical terms, this means that we have given Liz the means to delegate away much of the clerical grunt work she used to do, so she can focus on promoting AUUG.

About AUUGN

AUUGN Editorial Committee

The AUUGN Editorial Committee consists of:

- Frank Crawford
<frank@crawford.emu.id.au>,
and
- Greg Lehey <Greg.Lehey@auug.org.au>.

You can always reach the current committee at auugn@auug.org.au.

Send physical mail to the following address:

AUUG Inc
PO Box 7071
Baulkham Hills BC NSW 2153

Contributors:

Thanks to the following people for contributions to this issue: Frank Crawford <frank@crawford.emu.id.au>, Greg Lehey <Greg.Lehey@auug.org.au>, Owen Mace <owen.mace@kaz-group.com>, Martin Michlmayr <tbm@cyrius.com> and Devraj Mukherjee <devraj@eternitytechnologies.com>.

Public Relations and Marketing: Elizabeth Carroll <liz@auug.org.au>

AUUGN Submission Guidelines

Submission guidelines for AUUGN contributions can be obtained from the AUUG Web site at <http://www.auug.org.au/publications/auugn/subguide.html>.

AUUGN Back Issues

A number of back issues of AUUGN are still available. For price and availability please contact the AUUG Secretariat.

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

Direct enquiries regarding the purchase of the AUUGN mailing list 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 © 2004 by AUUG Inc.

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

AUUGN is the journal of AUUG Inc., the organisation of UNIX™ and Open Source Professionals. Its interests include 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.

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

Contribution Deadlines

Vol 25, No 4 (December 2004)	15 November 2004
Vol 26, No 1 (March 2005)	15 February 2005
Vol 26, No 2 (June 2005)	15 May 2005
Vol 26, No 3 (September 2005)	15 August 2005

Note that from January 2005, AUUGN will appear in electronic form only.

My Home Network

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

Well, AUUG2004 has now come and gone, and a good time was had by all (or at least me). Along with the conference came the latest issue of AUUGN, and if you read it, you will realise some changes are coming up. However, one thing that won't change is this column (as long as I can keep coming up with new ideas :-)). And so, now to this month's column.

As a present to my self (and a birthday present from my family) I picked up a new toy, an AMD Athlon 64 processor, and associated hardware. In fact, what I've picked up is a new Gigabyte motherboard (a GA-K8VNXP), an Athlon 64 3200+, and 1GB RAM consisting of 2 x DDR400 512MB DIMMS. To this I added a Matrox G550 graphics card (with 4xAGP support), a 40GB IDE disk, a floppy drive and a DVD ROM, from the old system, and voila, I had a new high-end system.

For those interested in what the model 3200+ means, it indicates that its performance is equivalent to an Intel Pentium 4 running at 3.2GHz. This does not mean that it runs at 3.2GHz, in fact it runs at 2.2GHz, and hence is cooler than the equivalent Pentium. An interesting fact is that even Intel are moving away from the idea of specifying performance via clock speeds, and is looking at a new naming scheme.

In line with previous motherboards, this one includes more and more built-in facilities. For example, aside from the standard ports (i.e. serial, parallel, mouse and keyboard), the GA-K8VNXP has an 800MHz FSB (Front-Side Bus), allowing very high internal data rates, 8 USB ports, 3 IEEE1394 ports (c.f. Firewire), 2 RJ45 LAN ports (1 GigE the other Fast Ethernet), ALC658 CODEC and audio support, 4 IDE and 2 SATA ports. In addition, 2 of the IDE and both the SATA ports can be configured in a hardware RAID (0 - striped or 1 - mirroring). Given all these features, it is amazing to find almost all of them are supported within the current Linux kernel (but more on that later).

For those who are unaware of AMD's Athlon 64, it is the desktop version of their Opteron, which has put Intel in a bit of a spin. It is a 64-bit extension to the standard Intel x86 architecture, among other things allowing access to larger virtual address space (16EB (Exabyte) vs 4GB, although, don't expect to see that much RAM in a system for a while) (okay, at present you can only access 40-bits of physical memory, and ultimately,

it is planned to extend this to 52-bits, however, this is only an implementation limitation, not a physical issue.), 64-bit general-purpose integer registers and 128-bit XMM registers. All this is while keeping compatibility with current x86 programs. (You might be interested to know that Intel are now planning to release their own 64-bit extensions, called 'EM64T', which they claim to have been working on for some time. However, nothing was mentioned about it, until after AMD released the Opteron.)

However, while the 64-bit architecture is important, it has been available in a number of other architectures for years (e.g. Sun's Sparc, SGI's MIPS and IBM's PowerPC), it isn't the only issue. One of the other big items is a revamp of the internal CPU bus to move to an open standard called HyperTransport. Rather than being a tradition bus structure HyperTransport is a very high-speed point-to-point link for connecting integrated circuits on the motherboard. It simplifies the connection of multiprocessor systems, as well as other devices, such as memory and other external devices.

Far more information can be found about this at AMD's website, <http://www.amd.com>.

So, what to do with this new hardware? Well, firstly, since it was an "upgrade" to an existing system, I just let it boot up. As with all my other Linux installations, it was running Fedora Core 2, optimised for an Intel Pentium 4. After putting all the hardware together and powering on, as usual, the Linux kernel booted. In fact, after running 'kudzu', the "stupid" utility (at least I've called it that a few times) to detect and configure hardware changes, the system was up and running fine with almost all of the new hardware correctly identified.

But, one of my aims with purchasing an Athlon 64 was to experiment with the features of the new CPU, in particular the 64bit code. A secondary issue was to look at how good the compatibility mode for i386 32bit code is. So, the first issue was to get native 64bit code onto the system. Fedora Core 2 (FC2) comes with two versions, i386 and x86_64 distributions, so my first thought was that an upgrade should perform the conversion. As this was an existing working installation, I wanted to keep everything working.

Unfortunately, this was a no-go.

Booting from the FC2 DVD and requesting an upgrade installed only a few packages, but didn't even install the x86_64 kernel! Thinking more rationally about this, it's not surprising it didn't

work. While there is a compatibility mode, as far as the installation is concerned, the i386 and x86_64 are two totally different architectures and unrelated.

Okay, so a simple upgrade wouldn't work, so next idea, lets individually upgrade the various packages. To do this I first obtained a list of all the packages currently installed with:

```
rpm -qa --qf '%{name}-%{version}-%{release}. \
  %{arch}.rpm\n' 2>&1 | sort > rpmpkgs
```

This is pretty much straight from the script `/etc/cron.daily/rpm`, and includes details of the architecture as well as the name, release, etc. Following this I just grepped the i386 packages from this list and located the equivalent packages in the FC2 DVD and updates. Now, while this scheme looked like one that would succeed, a quick test kicked up a lot of issues with the versions of RPM. In particular, the installation, because of a new feature called "multiarchitecture support", both the old i386 and the new x86_64 rpms were both installed and there were a number of ensuing conflicts.

Oh, well, after this, there weren't really too many other options. I could restore from the backup I'd taken, or do a fresh installation from the x86_64 FC2 DVD. Since my aim for this system is to learn about the x86_64 architecture, performing a fresh install was not a big issue. Now I won't go through the entire procedure for the installation, but I will mention that prior to everything else, I saved anything I wanted to keep on `/home`, and then ensured that I did not reformat the `/home` partition. Anyway, suffice to say that the fresh installation went fine and after about an hour I had a new x86_64 installation.

However, while having a base system was fine, and in terms of FC2 there are a lot of included features, there were a few things that aren't there. In fact, the two biggest areas needing other packages are Mozilla plugins and movie players. Since Mozilla is a 64bit application, most plugins are not compatible, or at least need to be recompiled before installation. The one exception to this is that `mozplugger`, which comes with FC2. In this case, 'mozplugger' acts as the Mozilla plugin, but then execs the appropriate program.

The first "plugin" I needed to find was Java. While normally this can be downloaded from <http://java.sun.com>, at present Sun doesn't support x86_64 versions. So off to <http://www.blackdown.org/java-linux.html>, which is based on Sun's 1.4.2_03 code. In fact, this is the original site for obtaining Java for Linux, be-

fore Sun started distributing Linux RPMs, and is still a good location for Java for unusual systems.

Over and above this, other plugins are a bit harder to find, and this reflects a general problem with x86_64 packages at present. While it isn't too hard to make these packages, many developers are not doing so yet. Even worse, sometime there are word size issues, etc, which need to be addressed, and haven't been yet, or don't appear until execution.

At the present time, there are only a few repositories that support FC2 x86_64 packages, and the two major ones I use are DAG (<http://dag.wieers.com/home-made/apt/packages.php>), FreshRPMs (<http://freshrpms.net>) (of which DAG is a part), Livna (<http://rpm.livna.org>) and Fedora.us (<http://www.fedora.us>), although Fedora.us really has very few x86_64 packages.

Rather than trying to find the appropriate packages, and the dependencies for some of these packages does start to get complicated, I decided to look at *yum*, the Yellowdog Updater, Modifier from Duke University (<http://www.linux.duke.edu/projects/yum>). *yum* is an automatic update and package install/remover for rpm system, automatically computing dependencies and ordering for correct installation. While you can download this package from the Yum site, I picked it up from DAG's site.

With *yum* you need to set up the configuration file with the locations of suitable repositories (some parts of which can be system variables, e.g. `$releaseversion` and `$basearch`) and then request packages which match certain patterns, e.g. `yum search videolan*`. The configuration file, `/etc/yum.conf`, I have is:

```
---
[main]
cachedir=/var/cache/yum
debuglevel=2
logfile=/var/log/yum.log
pkgpolicy=newest
distroverpkg=redhat-release
tolerant=1
exactarch=1
retries=20
pgpcheck=1

[base]
name=Fedora Core $releasever - $basearch - Base
baseurl=http://download.fedora.redhat.com/pub/fedora/linux/core/$releasever/$basearch/os/

[updates-released]
name=Fedora Core $releasever - $basearch - Released Updates
baseurl=file:///home/mirror/Fedora/updates/$releasever/$basearch/
```

```
[updates-released-i386]
name=Fedora Core $releasever - i386 - R
eleased Updates
baseurl=file:///home/mirror/Fedora/upda
tes/$releasever/i386/

[updates-testing]
name=Fedora Core $releasever - $basearc
h - Unreleased Updates
baseurl=http://planetmirror.com/pub/fed
ora/linux/core/updates/testing/$releas
e/$basearch/

[Fedora.us]
name=Fedora.us - $basearch - Extras
baseurl=http://fedora.linux.duke.edu/fe
dorax86_64/fedora.us/$releasever/$basea
rch/RPMS.stable

[dag]
name=Dag RPM Repository for Fedora Core
baseurl=http://apt.sw.be/fedora/$releas
e/en/$basearch/dag

[freshrpms]
name=FreshRPMs
baseurl=http://ayo.freshrpms.net/fedora
/linux/$releasever/$basearch/freshrpms/
http://ftp.us2.freshrpms.net/li
nux/freshrpms/ayofedora/linux/$releas
e/$basearch/freshrpms/

[livna-stable]
name=Livna.org Fedora Compatible Packag
es (stable)
baseurl=http://rpm.livna.org/fedora/$r
eleasever/$basearch/RPMS.stable

[livna-unstable]
name=Livna.org Fedora Compatible Packag
es (unstable)
baseurl=http://rpm.livna.org/fedora/$re
leasever/$basearch/RPMS.unstable

[livna-testing]
name=Livna.org Fedora Compatible Packag
es (testing)
baseurl=http://rpm.livna.org/fedora/$re
leasever/$basearch/RPMS.testing
```

You will see from this that I search various levels of the sites I mentioned earlier. If you go to the home page for each of these sites you will find the yum configurations for each site, as well as the GPG keys, which each of these packages is signed with.

While *yum* seems to be the update manager that has received the most support from the Fedora community, it isn't the only one, and they all seem to share repositories and information. The two other most common ones are *apt*, which has come from the Debian community, and *up2date*, which was originally developed by Red Hat. Probably the biggest interchange of ideas comes with *up2date*, which in the Fedora incantation uses the *yum* header information to determine updates.

One other important issue, which I haven't really addressed is that all the update managers work

with RPM's multiarchitecture support. In this case the x86_64 installation also support i386 rpms, and in many cases, the same product can be installed from two different architectures on the same system. Probably the only concession to this is that there are a number of 64bit specific directories, i.e. */lib* vs */lib64* and */usr/lib* vs */usr/lib64*. This does cause some issues for certain simple-minded installations, in particular systems that include dynamically loaded libraries (e.g. vlc, the videolan-client). They may be compiled in 64bit, but look in */usr/lib* rather than */usr/lib64* for additional packages.

Getting back to the original discussion, the new Athlon 64 system, and how is it going. Well at present I've managed to install mplayer and vlc as well as some other smaller products, however, while mplayer works and vlc partially works, there is still a fair bit of work to have all the same products available as I could find on a pure-i386 system. Even more importantly, this system is not really intended as a simple desktop, but as a performance workstation, and hopefully soon I'll get some chances to try it out as such.

With all these new power machines coming out, what are you doing with your machine? An important issue these days is that it isn't just CPU speed that is changing, but rather there are improvements through the entire system, including changes in some basic structures, in this case 32bit to 64bit architectures. These changes open up a number new possibilities, which would be interesting to hear about. So, why not write something up for everyone to hear about?

mychart Charting System for Recreational Boats

Owen Mace <owen.mace@kaz-group.com>

First, dear reader, let me introduce myself. I wrote my first computer program as an engineering student in 1966. The program predicted where to look in the sky for satellites. My professional life has been approximately equal parts academia and IT industry and during that time I have never written more than a few hundred lines of code, perhaps a thousand at most, at one time for my work.

I have enjoyed writing code and have, from time to time, written code outside my professional life. This project is my first attempt with a graphical user interface. I am therefore an ultra-newbie in the company of giants.

I am also a keen cruising yachtsman and I wanted a charting system to show the location of my boat when at sea. It's important to know when you are about to run out of sea, because absence of sea almost always means expensive things for boats. I'm sure you understand.



So I decided to build a chart plotter, which I call *mychart*, from non-proprietary, open components—software, hardware, data and standards.

mychart is a charting and navigation system for recreational boats.

The picture on the left is of my boat, called “Limelight”. She is a Duncanson 35 which means she is 35 feet long, about ten and a half metres, and some six tonnes of fibreglass and lead. Limelight is capable of cruising the world. Maybe someday ...

Overview

My decision to build *mychart* from “open” components has a number of implications:

- I use Open Source Software (OSS) throughout, for example the Linux operating system. I intend to team *mychart* with other OSS software such as OpenOffice and an open source multimedia player for the ship's entertainment system,
- Open Interfaces, that is, published, freely available and normally at no cost software and hardware interfaces are specified throughout, for example, NMEA 01831 (National Marine Electronics Association specification 0183 is the most widely used specification for communicating data on recreational vessels. Chances are that your GPS communicate with each other using NMEA 0183, version 2.2 or 3. It's an ASCII-based 4,800 baud protocol of data sentences.), serial and USB. Also, I use the Unix practice of text data files throughout so that users can view and modify data, if necessary,
- Open Data which is freely available coastline data from the US National Geophysical Data Center (<http://www.ngdc.noaa.gov/mgg/shorelines/shorelines.html>). Now this is where it becomes contentious. The World Vector Shoreline (WVS) is not very accurate and is not maintained. Therefore, *mychart* cannot be used as the primary means of navigation and must be used in conjunction with properly maintained paper or electronic charts. (Which cost an arm, a leg and three fingers to buy for a large area. I have plans for reducing the cost of maintained and accurate data, see the Future plans section.) , and
- Marinised PC hardware. I use *mychart* on the laptop that I bring on board each weekend but I have in mind another option, see future plans, later in this article, Because of these factors, the cost is as low as possible. In fact, anyone could build a navigation system and modify it, provided they have the appropriate skills.

I will repeat the caveat. *mychart* cannot be used as the primary means of navigation. The principles of good seamanship will guide you in the use of *mychart*.

Software Overview

When I started on *mychart*, I knew that I wanted a depiction of the sea and land in a format that mimics conventional marine charts. I had a couple of books on X Windows with sample programmes that introduced readers to the delights of X11 and worked them through so that I had some minor proficiency in the subject. That seemed to be OK for me, as I still remembered a little of the few lines of C that I had written while still in academe nearly 20 years ago.

So off I went, working through the examples in the various chapters of the book I selected, getting all the example programs to work, understanding them and learning a little of the joys of make. I even built an early version of *mychart* that displayed the coastlines credibly but I felt that handling mouse events would be a little too tedious, if not demanding, for my nascent knowledge of X11 programming.

What I did learn, however, is that an understanding of some of the principles of X11 helped me enormously as I climb the ladder up of abstraction and found myself on the Glade rung. But I'm ahead of myself.

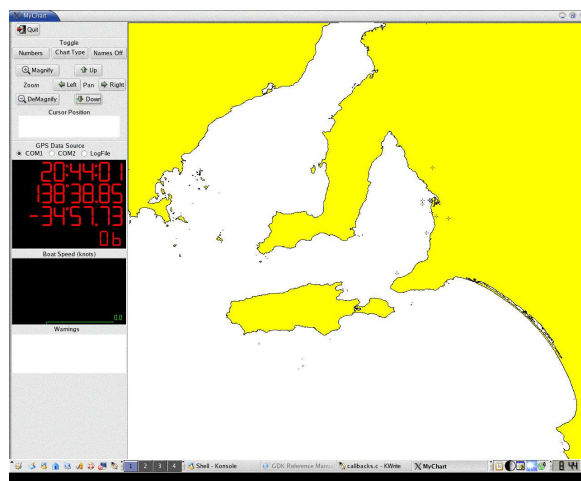
Along the way, I found that there was a real problem with the coastline data and I will describe my wrestling match with it in the Data Overview section. I also decided that I needed to lay down some rules for myself on the interfaces and I will talk about that subject in the Interface Overview section.

I spoke to some of the other open sourcers (or, as I prefer to call them, open sorcerers, for the magic they weave in their systems) in my workplace and received, of course, generous and conflicting advice. qt is definitely the way to go. No, Glade is much better for the Mandrake/KDE that I am running. Nah, stick to X11. A toss of a coin and Glade won. Not really, but I must offer prolific thanks to all who helped, advised, cajoled and laughed quietly out of earshot. So with help, I loaded Glade, Gtk and the other bits and pieces necessary to make a Glade application work. Plus, a sample "hello world" program, some Gtk tutorials, a gdk reference and I was away. Thanks Gary.

I have also disregarded their advice in another

area, too. *mychart* needs a 1280 x 1024 pixel display and does not scale to smaller displays. This resolution is needed to get the most from the chart display.

Here is what the main screen looks like now. You should recognise the Gulfs of St Vincent and Spencer, and Kangaroo Island.



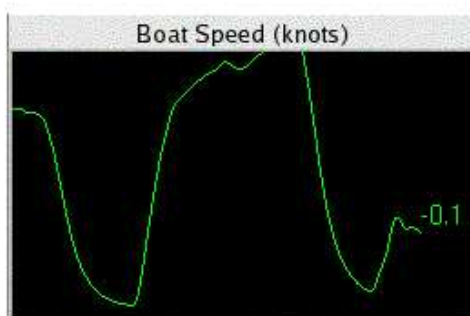
First of all I built a simple chart that I could zoom and pan, then I added options controlled by user buttons. (My, I was getting sophisticated!) The pan and zoom buttons are obvious in the left hand panel of the window.

Next, I wanted to deal with the external events of characters arriving at the serial port from the GPS receiver. Parsing the NMEA messages from my Garmin eTrex GPS receiver was a doddle for me because I had done it many times before. (Don't ask.) So now I could plot GPS positions on the chart—at least I knew where I was.

The next step was really two. I wanted to display the critical GPS information and show boat speed. Of course, I couldn't do both at once, so I decided to display the GPS data in big, bold type. Well, the gdk reference on selecting fonts intimidated me for reasons I cannot explain now, so I decided to use the good old seven segment display scheme that you can see on the screen shot below. I rationalised this as being a good scheme for night time when yachtsmen adapt their eyes to the dark so that they are alert for dim lights on the horizon. (I must have been dim, too, because I didn't think about the bright light of the chart display. Oh, never mind for now, we'll find a way out of that one later. Perhaps, I'll reduce the overall brightness of the display at night.)

Anyway, you can see how the GPS display turned out. From top to bottom, I show GMT time, longitude, latitude and the number of satellites visible to the receiver.

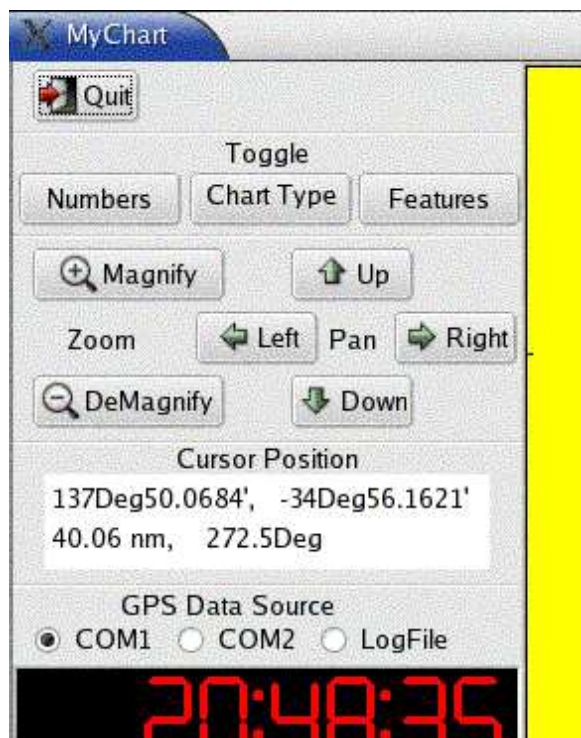
The next step was the graphical display of boat speed. There are a couple of considerations here. Firstly, small changes in the settings of ropes can make quite substantial differences to boat speed. I have seen a rope tensioned by just 150 mm and our boat began to steadily pull ahead of the boat beside us. However, there are large variations in boat speed from moment to moment as it rides over waves and adjusts to wind shift, so it is necessary to filter the speed quite heavily, as you can see from the image below which is actually from a journey in a car.



I calculate boat speed from changes in the position reported by the GPS receiver and these changes can have quite large errors in them, as much as 5 metres, so again, heavy filtering is advised.

The next question is what speed should I show? The speed of the boat through the water? Over land? Towards the next waypoint? All three? Probably. But for now, it's the speed shown by the GPS receiver.

Now I wanted to get really clever and show the latitude and longitude of the cursor and the distance and heading from the boat. You can see the result in the partial screen shot below.



Finally, there is an area for error messages.

mychart is a graphics application built with the following OSS tools:

- *Glade2* tool to built the window widgets,
- *Gtk+*, version 2.4 for manipulating events and widgets,
- *gdk* for primitive drawing functions, and
- *gcc* is the compiler.
- Naturally, *X11* is the underlying windowing system.

Interface Overview

I have adopted many of the conventions used in paper charts, such as expressing all latitudes and longitudes as degrees, minutes and fractions of a minute. Internally, I convert to struct Pos { double Lat, double Long} with the international conventions for positive and negative. Other nautical conventions are observed, such as distances which are displayed and input in nautical miles and speeds in knots, as you would expect.

Input devices are something to consider, too. Imagine a vessel in a large sea with a keyboard and mouse slipping around the chart table. It doesn't work for me and so *mychart* works from just a trackball that can be firmly attached to the chart table.

Hardware Overview

A standard PC architecture is all that is required to run *mychart*. However, the corrosive sea atmosphere will make very short work of the electronics if a PC were to be permanently mounted in a vessel. The obvious solution is to exclude the sea air from the PC and that's easy enough to do but the CPU has to be cooled by the same corrosive atmosphere that we are trying to exclude. You can forget about using seawater as that's another hole in the hull—no thanks.

However, my plan is to mount a motherboard and hard drive in a metal box. There will be an aluminium block that thermally connects the CPU to the outside and a good heatsink. The power supply should have no fans and will probably be in a separate box. All connections are made via short lengths of cable from the side of the box to the electronics inside so that they can be cheaply replaced when they corrode. A DVD/CD drive and any other peripherals will be USB connected. The box will be sealed with high temperature silicon. The display is a low power LCD model with all holes sealed. Heat will be dissipated from the display. I am told this will work.

My old 733 MHz 256 Mbyte PC is fine for the task. It computes and displays the 25,000 WVS points on the screen apparently instantaneously.

Data Overview

The World Vector Shoreline data that I use for the coastlines is readily downloaded but the data cannot be used as downloaded. Long shorelines are broken into sections and it cannot be guaranteed that contiguous segments are adjacent in the data file or even that they join correctly. That is, the last point in one segment joins to either the first or last point of the segment to which it joins.

It took me some time to realise this and then to join contiguous segments correctly. Now, the WVS data file (in ASCII text) is first sorted and then written for *mychart* to read in. I also use an ini file for settings and features, such as wrecks, markers, etc. I think that I should separate the file into two—one for settings and another for features as I expect that the features file will grow with time.

All data files are text so that they can be read and manipulated with a simple text editor.

Future Directions

Some of the developments that I want to do are described below.

Waypoints

One of the very useful tools is a means of guiding a vessel to a destination or waypoint. Staying on track or within a lane gives the vessel the best speed and in some circumstances can be a safety matter, too, if there are obstacles near the track. There are many ways of indicating a vessel's track and reminding the skipper to move right or left to return to the track. A flight director on an aircraft has a lane that rotates (to show that the aircraft is heading in the wrong direction) and a marker to show how far off the lane it is. GPS receivers depict a lane in the sea with the vessel placed on it to show its heading. I've experimented with both and I think I prefer the lane—somehow it's more intuitive. Here is a window that I have been experimenting with. My next task is to modify *mychart* so that I can set a waypoint with a right click and then a lane window will appear.

The next page shows a screen shot from my development. The vessel is to the left of the correct track (indicated by the red colour for port) but still within 100 metres of the track, and point off the correct direction by 24°.

Better Chart Data

WVS coastlines are very limited. Not only are they relatively inaccurate, but also small features may not appear as even a small rock, a meter or two in size, can make a big dent in a yacht. Nor is there any indication of depth or dangerous area, such as reefs, such as those shown on the charts that mariners use. Commercial chart plotters use proprietary data that is derived from, for example, the Australian Hydrographic Office (www.aho.gov.au). The markup on price can be astounding. AHO, it appears, will supply chart data in electronic form. Certainly raster is available, and possibly vector form as well, and can be purchased for a reasonable amount. So one of the enhancements I want to do is to use AHO raster data.



- The gadget part—how many gadgets you feel have to be added. This depends on a lot of factors, such as how much of a gadget person you are, your income, how many kids you have at school, etc, and
- Maintenance which is what you pay when the boat runs out of water. This is proportional to the cost of the boat, which in turn is proportional to the fourth power of the length. (Go figure!). Maintenance is also proportional to the speed the boat is going when it runs out of water, raised to some power. If the speed is low, its probably planned maintenance but if its large, then we've probably encountered a rock and that's expensive.

mychart helps to reduce the maintenance by reducing the chances of running out of water at high speed.

Marinised PC

I've described how I think a marinised PC can be built from a commercial motherboard. Since the infrastructure is Linux, a second hand machine running at under 1 GHz is fine. Another possibility is to under clock a new machine to lower its power requirements and heat generation.

USB instruments plus their interface

Traditionally, marine instruments for speed (called a log), depth and wind speed have been self contained, use proprietary and, sometimes, NMEA interfaces. I would like to build some USB sensors and use the PC for the smarts. Somebody else has probably thought of that already.

Sourceforge

Sourceforge has been a bit disappointing. I've been through the registration procedure twice, the first time the application was refused and I received no reply on the second attempt.

Conclusion

There is an old chestnut that posits that a boat is just a hole in the water into which you pour money. This is demonstrably unscientific and certainly not the case for a sailing vessel. The cost of running a sailing vessel is made up of three parts:

- The fixed part—insurance, repayments, berthing fees, registration, etc. This part depends on the value of the vessel, where you park it and your bank,

Minutes of AUUG board meeting, 5 May 2004

Location Sun Microsystems, North Sydney.

Attendees:

Elizabeth Carroll	EC
Adrian Close	AC
Jonathon Coombes	JC
Andrew Cowie	AfC
Gordon Hubbard	GH
Steve Landers	SL
Greg Lehey	GL
David Purdue	DP

Apologies:

Stephen Rothwell	SFR
Michael Still	MS

Meeting started at 10:00 am

1. Apologies
2. President's report.

In the past quarter, we held a security symposium in Canberra. It was quite successful, and it is another indication that our most successful activities are events.

AUUG is now also a member of the IT Council of South Australia. At the April board meeting I presented AUUG to the IT Council. The presentation was well received, and I believe that institutions like the IT Council can play an important role in AUUG's future.

In the coming quarter we hope to present a road show with John Terpstra, a one-day seminar in three capital cities on the topic of Samba migration. We had also hoped to hold the sixth Australian Open Source Symposium in Perth, but as of this report I have heard no confirmation that any significant planning is under way.

Since the meeting, plans for AOSS have been finalized. See page 61 for more details.

This brings me to my concerns: for the last three reports, I have documented the lack of activity which seems to be the order of the year. As I write this report, at the end of April, the first quarter issue of AUUGN has not made it to the membership. It will contain nomination forms for the AUUG board elections, to be submitted by 14 April. This alone would not explain the minimal number of

nominations for next year's board: only a single nomination from somebody who is not already on the board. A number of this year's board members, myself included, have not nominated for reelection, so there will be no election, and next year's board is two members short.

Other instances of inactivity abound: it appears that we have effectively been sidelined in our relationship with the Government, not because of our position, but because of our inactivity. Instead, groups like Linux Australia and even ACS are taking our place. It seems that nobody on the board or in the Open Computing for Government committee cares enough to make a contribution that would doubtless be valuable.

Even in our area of strength, things are not happening. I have already mentioned the AOSS 6, but at least there is some evidence of something happening there. The intention to hold a systems administration symposium seems not to have got off the ground, and preparations for AUUG 2004 are behind schedule. The editor of AUUGN has announced his intention to resign, and I can't even begin to think of who might replace him.

When I became president of AUUG two years ago, I put a large amount of personal effort into trying to turn AUUG around. I was helped by a number of other people who also put in far more effort than one could reasonably ask. Unfortunately, there were too few people involved, and none of us were successful in getting others to join in to any significant extent. I think this is the reason for the inactivity of the current board: those who were energetic are burnt out, and nothing is happening. Despite attempts at the beginning of the financial year, nothing at all has happened in the relationship with the chapters, which appear to be in continual decline.

The one possibility that I saw for getting out of this problem was a merger with Linux Australia, which contains a number of enthusiastic and energetic people. Unfortunately, it seems that cultural differences have predominated, and the idea didn't really get off the ground.

So whither AUUG? It appears that our membership continues to decline. I believe that next year's board must address this issue and find a solution for it, or we will soon have to shut down.

In view of the fact that we no longer have an AUUGN editor, I would suggest that we:

- Stop publishing AUUGN as a paper document, and do an online newsletter instead. Instead of including the articles, add just a link.
- Make membership free to to qualified individuals, those actively occupied in IT or related industries.
- Make membership free to people who attend a conference (possibly any conference, or maybe only the annual technical conference).

Obviously there's a lot to discuss here, including how we manage to pay for our business manager. I fear, however, that we have no alternative.

(end of report)

GL asked AfC for his take on GL's assertion that the AUUG board is tired and that LA seems more active.

- GL suggests we need to do something. at some point AUUG will be unable to pay EC's salary. Without EC, the organization would be changed beyond recognition.
- GH suggests GL is being "a little negative"
- AfC commends GL for his "heads up".

Motion to accept: AfC

Seconded: GH

Vote: accepted.

3. Secretary's report:

Not available at time of meeting.

4. Treasurer's report:

A copy of the Treasurer's report was submitted in a format that can't easily be printed in AUUGN.

- GH mentioned that AUUG made a loss in financial year 2002/3, slightly greater than 2001/2. Expenditure decreased over that time. Main reason for the drop in income was a decrease in membership.
- GH states that the last conference performed extremely well financially.
- GH suggests that our cash position is OK for now, but without action, we will have a problem in 12 months.
- GL speculates about the financial impact of dropping AUUG or reducing membership

fees.

Motion to accept: SFR

Seconded: DP

Vote: accepted.

5. Business Manager's report:

• AUUG 2004

We are close to the final date for the Call for Papers. Will then have a better idea of what papers have been received, and how many more including topics we will need to chase. Potential sponsors are being looked at including papers they can present.

• Security Symposium

A profit of \$1,493 was made from this event. Actual monies banked were higher, as this takes into account a cost for the Business Manager's time.

• Samba Roadshow

Registrations are currently being received, although less than originally anticipated. There is interest from Darwin which is currently under consideration.

• AOSS

Steve Landers has made a presentation in Perth to the various groups, looking at July. Steve to provide more detail at the meeting.

• Memberships

June renewals will go out in mid May. Still receiving December renewals.

• AUUGN

Con has resigned as Editor. March edition was late and will be received next week.

• Other

These are the key areas that I have been addressing recently, in addition to general issues, most of which will be discussed under the other topics at the meeting.

EC states that we need to run more events (for publicity et al)—smaller events can contribute significantly.

6. GH affirms that two or three events like the Security Symposium could turn around the financial situation.

7. EC mentions that corporate memberships are dropping off—probably perception that it doesn't represent value for money.
8. AfC expresses concern about the John Terpstra roadshow situation - did we establish that there was a market before proceeding?

Motion to accept: DP

Seconded: AfC

Vote: accepted

- Minutes of previous meeting

Motion to accept: DP

Seconded: JC

Vote: accepted

- Action items:
- see Wiki
- EC recommends we organise to accept Amex/Diners
- AUUG2004:
- programme stuff
- EC talking to sponsors
- Samba roadshow:
 - EC talked about how the event was publicised
 - GL related a chat with John Terpstra, where John suggested our mistake was to advertise it the way he suggested. Also that we needed to market it to Windows users.
 - GL asked for a show of hands on whether we should continue the road show format. General assent.
 - GH suggests the possibility of doing a security symposium/roadshow.
 - AfC talked about his concerns related to roadshows and comparing them with “installfests” (which are apparently now out of favour with LUGs). Not convinced that the market exists for roadshows.
 - JC comments that the only big open source events (e.g. opportunities for the open source community to get together) are LCA (start of year) and AUUG (end of year), with nothing much in between.

Motion to form a subcommittee to investigate minor events, to come up with a recommendation, reporting by end of June, comprising DP, SL and EC:

Moved: DP

Seconded: AC

Vote: Accepted.

- AOSS:
 - SL talking to SLPWA.
 - Venue probably “Technology Park”.
 - SL has asked SLPWA to be involved in terms of endorsement and publicity, as well as programme committee.
 - Possible timeframe July.
- Elections:

GL noted AfC non-nomination and enquired as to the reasons.
- Chapters and Chapter Activities:
 - NSW, VIC, SA active
 - TAS, QLD, WA, NT apparently dead.
- AUUGN:
 - Discussion about dropping paper format and going purely electronic.
 - GH
 - AC mentioned AutoSpeed publication model
 - need a new editor.
 - discussion of CDs et al
 - Web pages - online services:
 - AOSA:
 - Michael Paddon is the bunny. Publishing stuff RSN.
 - Other business:
 - GL: relationship with government.
 - motion to accept financial report from B.E. Clarke and Associates for year ended June 2003 to be an accurate statement of AUUG's financial performance in that financial year. Moved: DP Second: JC Vote: accepted.
 - liability insurance... AfC suggests the proposed cover looks good.
 - Sub-Committee EOL: All sub-committees to be automatically dissolved on 30th

June of each year. DP: Moved, AfC: seconded, Vote: accepted

Meeting closed 17:45 (or thereabouts).

Liberal license for ancient UNIX sources

We've talked about the following mail message, but this is the first time most people have seen it in print. It has been reformatted for AUUGN, but the text is (of course) the original, including a couple of typos. You'll note a number of other AUUG members mentioned in the list of recipients.

And yes, it did contain a typo for Warren Toomey's mail address (should be wkt, not wht). Amongst other things, this license allows us (finally!) to print the John Lions commentary that starts on page 17.

Date: Wed, 23 Jan 2002 15:03:37 -0800
 From: Dion Johnson <dionj@caldera.com>
 To: wht@minnie.tuhs.org
 Cc: dmr@bell-labs.com,
 ken@plan9.bell-labs.com,
 grog@lemis.com,
 John Terpstra <jht@caldera.com>,
 drew@caldera.com, maddog@li.org,
 evan@starnix.com, phatch@caldera.com,
 ransom@caldera.com

Subject: Liberal license for ancient UNIX sources

Dear Warren, and friends,

I'm happy to let you know that Caldera International has placed the ancient UNIX releases (V1-7 and 32V) under a "BSD-style" license. I've attached a PDF of the license letter hereto. Feel free to propogate it as you see fit.

I apologize that this has taken so long. We do not have a well regulated archive of these ancient releases, so we must depend upon you UNIX enthusiasts, historians, and original authors to help the community of interested parties figure out exactly what is available, where, and how.

Many thanks to Warren Toomey, of PUPS, and to Caldera's Bill Broderick, director of licensing services here. Both of these gentlemen were instrumental in making this happen. And thanks to our CEO, Ransom Love, whose vision for Caldera International prescribes cooperation and mutual respect for the open source communities.

Of course, there are thousands of

other people who should be acknowledged. I regret I do not have time or wisdom to make a list of them all, but maybe someone does, or has.

Anyway, here it is. Feel free to write to us if you want to understand more about how/why Caldera International has released this code, or you have any other comments that we should hear.

Sincerely,

Dion L. Johnson II - dionj@caldera.com

Product Manager and one of many open source enthusiasts in Caldera Intl.

Paul Hatch - phatch@caldera.com
 Public Relations Manager at Caldera International



January 23, 2002

Dear UNIX® enthusiasts,

Caldera International, Inc. hereby grants a fee free license that includes the rights use, modify and distribute this named source code, including creating derived binary products created from the source code. The source code for which Caldera International, Inc. grants rights are limited to the following UNIX Operating Systems that operate on the 16-Bit PDP-11 CPU and early versions of the 32-Bit UNIX Operating System, with specific exclusion of UNIX System III and UNIX System V and successor operating systems:

32-bit 32V UNIX
16 bit UNIX Versions 1, 2, 3, 4, 5, 6, 7

Caldera International, Inc. makes no guarantees or commitments that any source code is available from Caldera International, Inc.

The following copyright notice applies to the source code files for which this license is granted.

Copyright(C) Caldera International Inc. 2001-2002. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code and documentation must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

All advertising materials mentioning features or use of this software must display the following acknowledgement:

This product includes software developed or owned by Caldera International, Inc.

Neither the name of Caldera International, Inc. nor the names of other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

USE OF THE SOFTWARE PROVIDED FOR UNDER THIS LICENSE BY CALDERA INTERNATIONAL, INC. AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL CALDERA INTERNATIONAL, INC. BE LIABLE FOR ANY DIRECT, INDIRECT INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Very truly yours,

/signed/ Bill Broderick

Bill Broderick
Director, Licensing Services

* UNIX is a registered trademark of The Open Group in the US and other countries.

A commentary on the Sixth Edition UNIX Operating System, part 1

John Lions

This article is the first in a series of reprints of the infamous “Lions Book”, written by John Lions around 1977. See the editorial for more details.

The copyright notice below is the original. It is now of historical importance only. The current version is reproduced with the kind permission of Peter Salus, who has full rights to reproduce the documents.

Since the release of “Ancient UNIX” by Caldera in 2002 (see page 17), the source code in question is available under a BSD-style license. The source code is not included here, but is available at <http://www.tubs.org>. It will be included in an AUUGN CD in the near future.

Introduction

This booklet has been produced for students at the University of New South Wales taking courses 6.602B and 6.657G. It is intended as a companion to, and commentary on, the booklet *UNIX Operating System Source Code, Level Six*.

The UNIX Software System was written by K. Thompson and D. Ritchie of Bell Laboratories, Murray Hill, NJ. It has been made available under a license from the Western Electric Company.

This document may contain information covered by one or more licenses, copyrights and non-disclosure agreements. Circulation of this document is restricted to holders of a license for the UNIX Software System from Western Electric. All other circulation or reproduction is prohibited.

© Copyright 1977 J. Lions

Preface

This book is an attempt to explain in detail the nucleus of one of the most interesting computer operating systems to appear in recent years.

It is the UNIX Time-sharing System, which runs on the larger models of Digital Equipment Corporation’s PDP11 computer system, and was developed by Ken Thompson and Dennis Ritchie at Bell Laboratories. It was first announced to the world in the July, 1974 issue of the “Communications of the ACM”.

Very soon in our experience with UNIX, it suggested itself as an interesting candidate for formal study by students, for the following reasons:

- it runs on a system which is already available to us;
- it is compact and accessible;
- it provides an extensive set of very usable facilities;
- it is intrinsically interesting, and in fact breaks new ground in a number of areas.

Not least amongst the charms and virtues of the UNIX Time-sharing System is the compactness of its source code. The source code for the permanently resident “nucleus” of the system when only a small number of peripheral devices is represented, is comfortably less than 9,000 lines of code.

It has often been suggested that 1,000 lines of code represents the practical limit in size for a program which is to be understood and maintained by a single individual. Most operating systems either exceed this limit by one or even two orders of magnitude, or else offer the user a very limited set of facilities, i.e. either the details of the system are inaccessible to all but the most determined, dedicated and long-suffering student, or else the system is rather specialised and of little intrinsic interest.

There seem to be three main approaches to teaching Operating Systems. First there is the “*general principles*” approach, wherein fundamental principles are expounded, and illustrated by references to various existing systems, (most of which happen to be outside the students’ immediate experience). This is the approach advocated by the COSINE Committee, but in our view, many students are not mature or experienced enough to profit from it.

The second approach is the “*building block*” approach, wherein the students are enabled to synthesize a small scale or “toy” operating system for themselves. While undoubtedly this can be a valuable exercise, if properly organised, it cannot but fail to encompass the complexity and sophistication of real operating systems, and is usually biased towards one aspect of operating system design, such as process synchronisation.

The third approach is the “*case study*” approach. This is the one originally recommended for the Systems Programming course in “Curriculum ’68”, the report of the ACM Curriculum Committee on Computer Science, published in the March, 1968 issue of the “Communications of the ACM”.

Ten years ago, this approach, which advocates devoting “most of the course to the study of a single system” was unrealistic because the cost of providing adequate student access to a suitable system was simply too high.

Ten years later, the economic picture has changed significantly, and the costs are no longer a decisive disadvantage if a minicomputer system can be the subject of study. The considerable advantages of the approach which undertakes a detailed analysis of an existing system are now attainable.

In our opinion, it is highly beneficial for students to have the opportunity to study a working operating system in all its aspects.

Moreover it is undoubtedly good for students majoring in Computer Science, to be confronted at least once in their careers, with the task of reading and understanding a program of major dimensions.

In 1976 we adopted UNIX as the subject for case study in our courses in Operating Systems at the University of New South Wales. These notes were prepared originally for the assistance of students in those courses (6.602B and 6.657G).

The courses run for one semester each. Before entering either Course, students are presumed to have studied the PDP11 architecture and assembly language, and to have had an opportunity to use the UNIX operating system during exercises for earlier courses.

In general, students seem to find the new courses more onerous, but much more satisfying than the previous courses based on the “general principles” approach of the COSINE Committee.

Some mention needs to be made regarding the documentation provided by the authors of the UNIX system. As reproduced for use on our campus, this comprises two volumes of A4 size paper, with a total thickness of 3 cm, and a weight of 1250 grams.

A first observation is that the whole documentation is not unreasonably transportable in a student’s brief case. However it must not be assumed that this amount of documentation, which is written in a fresh, terse, whimsical style, is necessarily inadequate.

In fact the second observation (which is only made after considerable experience) is that for reference purposes, the documentation is remarkably comprehensive. However there is plenty of scope for additional tutorial material, one part of which, it is hoped, is satisfied by these notes.

The actual UNIX operating system source code is recorded in a separate companion volume entitled “UNIX Operating System Source Code”, which was first printed in July, 1976. This is a specially edited selection of code from the Level Six version of UNIX, as received by us in December, 1975.

During 1976, an initial version of the present notes was distributed in roneoed form, and only in the latter part of the year were the facilities of the “nroff” text formatting program exploited. The opportunity has recently been taken to revise and “nroff” the earlier material, to make some revisions and corrections, and to integrate them into their present form.

A decision had to be made quite early regarding the order of presentation of the source code. The intention was to provide a reasonably logical sequence for the student who wanted to learn the whole system. With the benefit of hindsight, a great many improvements in detail are still possible, and it is intended that these changes will be made in some future edition.

It is our hope that this book will be of interest and value to many students of the UNIX Time-sharing System. Although not prepared primarily for use as a reference work, some will wish to use it as such. The indices provided at the end should go some of the way towards satisfying the requirement for reference material at this level.

Since these notes refer to proprietary material administered by the Western Electric Company, they can only be made available to licensees of the UNIX Time-sharing System and hence are unable to be published through more usual channels.

Corrections, criticism and suggestions for improvement of these notes will be very welcome.

Acknowledgements

The preparation of these notes has been encouraged and supported by many of my colleagues and students including David Carrington, Doug Crompton, Ian Hayes, David Horsfall, Peter Ivanov, Ian Johnstone, Chris Maltby, Dave Milway, John O’Brien and Greg Rose.

Pat Mackie and Mary Powter did much of the initial typing, and Adele Green has assisted greatly in the transfer of the notes to “nroff” format.

David Millis and the Publications Section of the University of New South Wales have assisted greatly with the mechanics of publication, and Ian Johnstone and the Australian Graduate School of

Management provided facilities for the preparation of the final draft.

Throughout this project, my wife Marianne has given me unfailing moral support and much practical support with proof-reading.

Finally Ken Thompson and Dennis Ritchie started it all.

To all the above, I wish to express my sincere thanks.

The co-operation of the “nroff” program must also be mentioned. Without it, these notes could never have been produced in this form. However it has yielded some of its more enigmatic secrets so reluctantly, that the author’s gratitude is indeed mixed. Certainly “nroff” itself must provide a fertile field for future practitioners of the program documenter’s art.

John Lions
Kensington, NSW
May, 1977

1 Introduction

“UNIX” is the name of a time-sharing system for PDP11 computers, written by Ken Thompson and Dennis Ritchie at Bell Laboratories. It was described by them in the July, 1974 issue of the “Communications of the ACM”.

UNIX has proved to be effective, efficient and reliable in operation and was in use at more than 150 installations by the end of 1976.

The amount of effort to write UNIX, while not inconsiderable in itself (~+10 man years up to the release of the Level Six system) is insignificant when compared to other systems. (For instance, by 1968, OS/360 was reputed to have consumed more than five man millennia and TSS/360, another IBM operating system, more than one man millennium.)

Of course there are systems which are easier to understand than UNIX but, it may be asserted, these are invariably much simpler and more modest in what they attempt to achieve. As far as the list of features offered to users is concerned, UNIX is in the “big league”. In fact it offers many features which are notable by their absence from some of the well-known major systems.

1.1 The UNIX Operating System

The purpose of this document, and its companion, the “UNIX Operating System Source Code”, is to present in detail that part of the UNIX time-sharing system which we choose to call the “UNIX Operating System”, namely the code which is permanently resident in the main memory during the operation of UNIX. This code has the following major functions:

- initialisation;
- process management;
- system calls;
- interrupt handling;
- input/output operations;
- file management.

1.2 Utilities

The remaining part of UNIX (which is much larger!) is composed of a set of suitably tailored programs which run as “user programs”, and which, for want of a better term, may be termed “utilities”.

Under this heading come a number of programs with a very strong symbiotic relationship with the operating system such as

- the “shell” (the command language interpreter)
- “/etc/init” (the terminal configuration controller)

and a number of file system management programs such as:

check	du	rmdir
chmod	mkdir	sync
clri	mkfs	umount
df	mount	update

It should be pointed out that many of the functions carried out by the above-named programs are regarded as operating system functions in other computer systems, and that this certainly does contribute significantly to the bulk of these other systems as compared with the UNIX Operating System (in the way we have defined it).

Descriptions of the function and use of the above programs may be found in the “UNIX Programmer’s Manual” (UPM), either in Section I (for the commonly used programs) or in Section VIII (for the programs used only by the System Manager).

1.3 Other Documentation

These notes make frequent reference to the “UNIX Programmer’s Manual” (UPM), occasional reference to the “UNIX Documents” booklet, and constant reference to the “UNIX Operating System Source Code”.

All these are relevant to a complete understanding of the system. In addition, a full study of the assembly language routines requires reference to the “PDP11 Processor Handbook”, published by Digital Equipment Corporation.

1.4 UNIX Programmer’s Manual

The UPM is divided into eight major sections, preceded by a table of contents and a KWIC (Key Word In Context) index. The latter is mostly very useful but is occasionally annoying, as some indexed material does not exist, and some existing material is not indexed.

Within each section of the manual, the material is arranged alphabetically by subject name. The section number is conventionally appended to the subject name, since some subjects appear in more than one section, e.g. “CHDIR(I)” and “CHDIR(II)”.

Section I contains commands which either are recognised by the “shell” command interpreter, or are the names of standard user utility programs;

Section II contains “system calls” which are operating system routines which may be invoked from a user program to obtain operating system service. A study of the operating system will render most of these quite familiar;

Section III contains “subroutines” which are library routines which may be called from a user program. To the ordinary programmer, the distinctions between Sections II and III often appear somewhat arbitrary. Most of Section III is irrelevant to the operating system;

Section IV describes “special files”, which is another name for peripheral devices. Some of these are relevant, and some merely interesting. It depends where you are;

Section V describes “File Formats and Conventions”. A lot of highly relevant information is tucked away in this section;

Sections VI and VII describe “User Maintained” programs and subroutines. No UNIXophile will ignore these sections,

but they are not particularly relevant to the operating system;

Section VIII describes “system maintenance” (software, not hardware!). There is lots of useful information here, especially if you are interested in how a UNIX installation is managed.

1.5 UNIX Documents

This is a somewhat miscellaneous collection of essays of varying degrees of relevance:

- *Setting up UNIX* really belongs in Section VIII of the UPM (it’s relevant);
- *The UNIX Time-sharing System* is an updated version of the original “Communications of the ACM” paper. It should be re-read at least once per month;
- *UNIX for Beginners* is useful if your UNIX experience is limited;
- The tutorials on “C” and the editor, and the reference manuals for “C” and the assembler are highly useful unless you are completely expert;
- *The UNIX I/O System* provides a good overview of many features of the operating system;
- *UNIX Summary* provides a check list which will be useful in answering the question what does an operating system do?

1.6 UNIX Operating System Source Code

This is an edited version of the operating system as supplied by Bell Laboratories.

The code selection presumes a “model” system consisting of:

- PDP11/40 processor;
- RK05 disk drives;
- LP11 line printer;
- PC11 paper tape reader/punch;
- KL11 terminal interface.

The principal editorial changes to the source code are as follows:

- the order of presentation of files has been changed;
- the order of material within several files has been changed;

- to a very limited extent, code has been transferred between files (with hindsight a lot more of this would have been desirable);
- about 5% of the lines have been shortened in various ways to less than 66 characters (by elimination of blanks, rearrangement of comments, splitting into two lines, etc.);
- a number of comments consisting of a line of underscore characters have been introduced, particularly at the end of procedures;
- the size of each file has been adjusted to an exact multiple of 50 lines by padding with blank lines;

The source code has been printed in double column format with fifty lines per column, giving one hundred lines per sheet (or page). Thus there is a convenient relationship between line numbers and sheet numbers.

A number of summaries have been included at the beginning of the Source Code volume:

- A *Table of Contents* showing files in order of appearance, together with the Procedures they contain;
- An *alphabetical list* of procedures with line numbers;
- A *list of Defined Symbols* with their values;
- A *Cross Reference Listing* giving the line numbers where each symbol is used. (Reserved words in “C” and a number of commonly used symbols such as “p” and “u” have been omitted.)

1.7 Source Code Selections

The source code has been divided into five sections, each devoted primarily to a single major aspect of the system.

The intention, which has been largely achieved, has been to make each section sufficiently self-contained so that it may be studied as a unit and before its successors have been mastered:

Section One deals with system initialisation, and process management. It also contains all the assembly language routines;

Section Two deals with interrupts, traps, system calls and signals (software interrupts);

Section Three deals primarily with disk operations for program swapping and basic, block oriented input/output. It also deals with the manipulation of the pool of large buffers;

Section Four deals with files and file systems: their creation, maintenance, manipulation and destruction;

Section Five deals with “character special files”, which is the UNIX term for slow speed peripheral devices which operate out of a common, character oriented, buffer pool.

The contents of each section is outlined in more detail in Chapter Four.

1.8 Source Code Files

Each of the five sections just described consists of several source code files. The name of each file includes a suffix which identifies its type:

“.s” denotes a file of assembly language statements;

“.c” denotes a file of executable “C” language statements;

“.h” denotes a file of “C” language statements which is not for separate compilation, but for inclusion in other “.c” files when they are compiled i.e. the “.h” files contain global declarations.

1.9 Use of these notes

These notes, which are intended to supplement the comments already present in the source code, are not essential for understanding the UNIX operating system. It is perfectly possible to proceed without them, and you should attempt to do so as long as you can.

The notes are a crutch, to aid you when the going becomes difficult. If you attempt to read each file or procedure on your own first, your initial progress is likely to be slower, but your ultimate progress much faster. Reading other people’s programs is an art which should be learnt and practised because it is useful!

1.10 A Note on Programming Standards

You will find that most of the code in UNIX is of a very high standard. Many sections which initially seem complex and obscure, appear in the light of further investigation and reflection, to be perfectly obvious and “the only way to fly”.

For this reason, the occasional comments in the notes on programming style, almost invariably refer to apparent lapses from the usual standard of near perfection.

What caused these? Sometimes it appears that the original code has been patched expediently. More than once apparent lapses have proved not to be such: the “bad” code has been found in fact to incorporate some subtle feature which was not at all apparent initially. And some allowance is certainly needed for occasional human weakness.

But on the whole you will find that the authors of UNIX, Ken Thompson and Dennis Ritchie, have created a program of great strength, integrity and effectiveness, which you should admire and seek to emulate.

- University of New South Wales
- University of Sydney
- University of Technology, Sydney
- Workcover Queensland

AUUG Corporate Members

As of 1 June 2004

- Apple Computer Australia Pty Ltd
- Australian Bureau of Statistics
- Australian Taxation Office
- BAE Systems
- Cape Grim B.A.P.S
- Corinthian Industries (Holdings) Pty Ltd
- Cray Australia
- CSIRO Manufacturing Science and Technology
- Curtin University of Technology
- Cybersource
- Deakin University
- Department of Land & Water Conservation
- Department of Lands
- Everything Linux & Linux Help
- EWA-Australia Pty Ltd
- IBM
- IBM Linux Technology Centre
- IP Australia
- KAZ Technology Services
- LPINSW
- Macquarie University
- Multibase WebAustralis Pty Limited
- NSW Department of Commerce
- Peter Harding & Associates Pty. Ltd.
- Powerhouse Museum
- Squiz Pty Ltd
- Sydney Water Corporation
- Tellurian Pty. Ltd.
- The University of Western Australia
- Thiess Pty Ltd
- TMD Computing
- University of NSW department of Computer Science & Engineering
- UNiTAB Limited
- University of New England

Managing Debian

Martin Michlmayr <tbm@cyrius.com>

Debian is one of the most unique Free Software projects in existence today. Not only is it one of the largest and most successful projects, it is also unique in its stance towards the philosophy of Free Software. The project has defined the Debian Free Software Guidelines¹ which have later been adopted as the foundation of the Open Source Definition.² Furthermore, Debian has a Social Contract³ in which the project lists its priorities and its promises explicitly. The document consists of five points:

1. *Debian Will Remain 100% Free Software:* the project promises that the Debian GNU/Linux distribution will always be free according to the Debian Free Software Guidelines.
2. *We Will Give Back to the Free Software Community:* software written for Debian will be free and made available to the whole community. Furthermore, bug fixes and enhancements for existing software will be given back to the author of that software.
3. *We Won't Hide Problems:* the project maintains an open Bug Tracking System (BTS) in which information about all known bugs can be found.
4. *Our Priorities are Our Users and Free Software:* we will be guided by the needs of our users and the Free Software community.
5. *Programs That Don't Meet Our Free-Software Standards:* software which is not free according to the Debian Free Software Guidelines might still be needed by our users. While such software cannot become a part of the Debian distribution, there are `contrib` and `non-free` areas on our FTP servers for software which is free but depends on non-free software and for non-free software itself, respectively.



Debian GNU/Linux is the most comprehensive Linux distribution available to date. It has over 10,000 packages and is available for 11 architectures.⁴ Debian is created by a large number of volunteers who are distributed all over the world (see above, or in more detail <http://www.debian.org/devel/developers.map.jpeg>). In total, there are over 800 official Debian developers and more than 200 other contributors. An interesting question is how Debian actually works. How can so many volunteers, distributed all over the world, work together effectively and produce a distribution with such a high degree of complexity? One important factor which helps to keep Debian together and work smoothly is certainly the project's organizational structure and its infrastructure. Debian has existed for almost 10 years and, over the time, the project has established a solid infrastructure which works very well. In the following, the organizational structure and infrastructure which makes Debian work will be introduced. It will also be investigated who carries out coordination and management tasks in the project and how this is done.

The Officers

In general, Debian has a very flat hierarchy. With some exceptions, all official Debian developers have the same rights. This is one of the reasons why Debian has a very thorough application process. Once an application is accepted, they can upload packages to the main archive of Debian. Those packages are installed on an individual's computer as `root` and therefore packages can maliciously or accidentally cause great damage. Hence, the New Maintainer process,⁵ which admits new Debian developers, checks the identity, philosophy (with regards to Free Software and the

1. http://www.debian.org/social_contract##guidelines

2. <http://www.opensource.org/docs/definition.php>

3. http://www.debian.org/social_contract

4. Debian GNU/Linux 3.0 (woody) has been released for alpha, arm, hppa, i386, ia64, m68k, mips, mipsel, powerpc, s390 and sparc. Other ports are in development.

5. <http://nm.debian.org/>

Social Contract) and technical skills of an applicant. The great exception to the flat hierarchy are Debian's officers who have a special status as defined in Debian's constitution.¹ As the Debian project grew, it became apparent that there needed to be a set of semi-formal rules to help in conflict resolution, and as a result the constitution was written. The Debian constitution describes the organizational structure for formal decision making in the Project. The constitution delineates who makes decisions, and what powers are attached to each such decision making individual or body. The officers listed in the constitution consist of the Debian Project Leader (DPL), the Project Secretary and the Technical Committee.

The Project Leader

The Debian Project Leader (DPL) is the official representative of the Debian Project. He or she has two main functions, one internal and one external. In the external function, he represents the Debian Project to others. This involves giving talks and presentations about Debian and attending trade shows, as well as building good relationships with other organizations and companies. Internally, the Project Leader manages the project and defines its vision. He should talk to other Debian developers, especially to the delegates, to see how he can assist their work. A main task of the Project Leader therefore involves coordination and communication. The Project Leader is chosen in an election in which all Debian Developers are eligible to vote. The Project Leader's term of office is one year. Nine weeks before the leadership post becomes vacant, the Project Secretary initiates a new election. During the first three weeks, any Debian Developer can become a candidate for this post by nominating themselves. The next three weeks are used for campaigning. Each candidate posts their platforms and everyone can direct questions to one or all candidates. The last three weeks consist of the polling period during which developers may cast their votes. Here are some examples of specific DPL tasks:

- Appoint Delegates or delegate decisions to the Technical Committee: the Project Leader may define a specific area of responsibility and delegate it to a Debian developer.
- Lend authority to other Developers: the Project Leader may make statements support of support for points of view or for other members of the project.

- Make any decision which requires urgent action
- Make any decision for whom nobody else has responsibility.
- Together with SPI, make decisions affecting property held in trust for purposes related to Debian: the Project Leader may make decisions about how money owned by Debian is to be used.

The current Project Leader is Martin Michlmayr.

The Project Secretary

Unlike other delegates, who are appointed by the Project Leader, the next Project Secretary is appointed by the Project Leader and the current Project Secretary. In case the current secretary and the project leader disagree, they must ask the board of Software in the Public Interest (SPI)² to appoint a Secretary.

- Conducting votes: The most visible task performed by the secretary is conducting votes for the project -- notably the Project Leader elections, but also any other votes that are run (General Resolutions, for example). Running a vote also entails determining the number and identity of the people eligible to vote, for the purpose of calculating quorum.
- Standing in for other Officers: The Project Secretary can stand in for the Leader, together with the Chairman of the Technical Committee. In this situation, they may jointly make decisions if they consider it imperative to do so -- but only when absolutely necessary and only when consistent with the consensus of the Developers. If there is no Project Secretary or the current Secretary is unavailable and has not delegated authority for a decision then the decision may be made or delegated by the Chairman of the Technical Committee, as Acting Secretary.
- Interpreting the Constitution: The secretary is also responsible for adjudicating any disputes about interpretation of the constitution.

The current Project Secretary is Manoj Srivastava.

The Technical Committee

The Technical Committee is the body which makes the final decision on technical disputes in the Debian project. It can consist of up to eight members and usually has at least four members. The Technical Committee may:

1. <http://www.debian.org/devel/constitution>

2. SPI is Debian's legal organization.

- Decide on any matter of technical policy: This includes the contents of the technical policy manuals, developers' reference materials, example packages and the behavior of non-experimental package building tools.
- Decide any technical matter where Developers' jurisdictions overlap: In cases where Developers need to implement compatible technical policies or stances (for example, if they disagree about the priorities of conflicting packages, or about ownership of a command name, or about which package is responsible for a bug that both maintainers agree is a bug, or about who should be the maintainer for a package) the technical committee may decide the matter.
- Make a decision when asked to do so: Any person or body may delegate a decision of their own to the Technical Committee, or seek advice from it.
- Overrule a Developer (requires a 3:1 majority): The Technical Committee may ask a Developer to take a particular technical course of action even if the Developer does not wish to; this requires a 3:1 majority.

The current Chairman of the Technical Committee is Ian Jackson.

Teams

In addition to the officers whose roles and powers are explicitly described in the constitution, several teams have formed naturally. These teams have clear areas of responsibility, and are almost exclusively technical in nature. Since very few Debian developers are paid for their work on Debian, they tend to do what they most enjoy. The teams (and there are a number of them) form simply; when more than one Debian developer wishes to work on a given task, and when that's technically feasible, they do. Having demonstrated both the skill to perform a given task, and the willingness to do so, the teams are typically well-staffed with knowledgeable and enthusiastic participants. The end result is that it's rare that any single Debian developer is overburdened, and a level of peer review and technical excellence that is widely held with respect.

Quality Assurance

The Quality Assurance (QA) team tries to make sure that high quality standards are held up. It maintains packages which temporarily do not have a maintainer. Also, it searches for inactive

maintainers and buggy packages. If a maintainer is found who does not maintain their packages anymore, the packages are taken away so other maintainers can take them and take care of them properly. Although no one gave the Quality Assurance group the explicit power for this, they have established authority by doing it.

FTP Master

The FTP masters are responsible for Debian's software archive. They maintain the software which drives the archive and perform the day-to-day work which is needed. This involves processing new packages and removing packages on the request of the maintainer of the QA group.

Listmasters

Debian offers about 150 mailing lists to facilitate the communication between developers and users. The listmasters make sure that the mailing lists are working properly and that as few spam as possible reached the mailing lists. They also deal with user questions and requests regarding the mailing lists.

Debian Admin

Debian Admin is responsible for the debian.org machines and hence much of Debian's infrastructure. Debian offers a wide variety of ports to many architectures and has many different machines on which architecture specific bugs and packaging issues can be tested. Furthermore, Debian Admin is responsible for the main infrastructure, such as machines running the web, FTP and mailing list services.

Web Team and Translations

The web team maintains Debian's extensive web pages. They are internally maintained in CVS and written in WML from which HTML is generated. There is a large number of volunteers who translate the web pages to other languages.

Security Team

The Security Team tracks security issues in released Debian packages and issues advisories and updated packages. There are full members and secretaries. The security secretaries cannot publish advisories on their own, but primarily track security issues and provide patches and updated packages to full members, who can then issue an advisory.

Release Management

Release management is a very important task in every Free Software project because someone has to do the coordination effort needed in order to get new releases out of the door. In the past, Debian had one Release Manager working on this. Since recently, he is helped by Release Assistants who help making sure Debian is ready to release. The decisions about the release are made solely by the Release Manager, however.

New Maintainer

New Maintainer (NM) is the process which admits new Debian developers. The Debian Account Manager (DAM) is a delegated position who has the authority to create new accounts. The NM Front Desk coordinates the overall process and makes sure that everything works smoothly. They coordinate with the DAM, the applicants and also with the Application Managers (AM) who lead applications through this process.

Policy

Debian has 10,000 packages created by almost 1000 different people. In order to ensure that Debian is an integrated system, a set of guidelines has been created which describe to which standards a package has to conform. This document, Debian Policy, is maintained by a group of experienced developers. There are also detailed guidelines how the Policy document may be changed.

Legal

There are always important legal issues to discuss when distributing software created by others. The legal team is responsible for making a coherent decision about legal questions. For example, they are the first contact when the question arises if a particular license can be regarded as Free Software.

Infrastructure

One of the most important factors which holds Debian together is the project's solid infrastructure. Most coordination and communication is carried out through public mailing lists, IRC channels and the Bug Tracking System (BTS). One aspect of Debian's development model is that it is open for anyone. All mailing lists (with the exception of `debian-private` where sensitive or confidential issues are sometimes discussed) are open and everyone can subscribe to them or read

the archives on the web. Also, all bugs found in Debian packages or feature requests are submitted through the Bug Tracking System. In accordance with the 3rd point of the Social Contract, *We Won't Hide Problems*, the BTS is open to anyone. If a bug is not documented there, we probably don't know about it. This infrastructure is essential for the way Debian works. The developers are distributed all over the world and hence effective means of communication had to be created. The mailing lists, the IRC channels, the Bug Tracking System and recently also the Package Tracking System (PTS) facilitate communication and coordination. In fact, they do not only allow communication between developers, but also between developers and users. Users can follow the mailing lists and describe their problems with the current system, their requirements and wishes. Debian's development model is truly open -- anyone can get involved and make a change, be it by reporting bugs, providing good comments or patches to known problems. This is also beneficial for companies which use Debian. They see exactly in which direction Debian is moving and can also get involved to drive a release forwards.

The Mailing Lists

There are about 150 mailing lists, each with a specific topic. There are mailing lists where users can ask questions, such as `debian-user`, as well as user mailing lists in specific languages. Most mailing lists, however, are mainly aimed for Debian developers or other interested parties to discuss specific technical aspects of Debian. The list `debian-devel` is the big development discussion list, but there are many smaller lists dedicated to a specific topic, such as various porting mailing lists. Additionally, there is `debian-devel-announce` which is required for all Debian developers since important announcements regarding the development of Debian are made there. The lists `debian-announce` and `debian-security-announce` are for general and security announcements, respectively, and are a must for every user of Debian. Also, `debian-news` is a good way to stay up-to-date what is happened around Debian. A complete index of mailing lists can be found on the web¹—everyone who is interested in Debian or specific aspects of Debian's development can subscribe to the lists of their choice.

1. <http://lists.debian.org/>

IRC

IRC (Internet Relay Chat) is a real-time chat system with different channels dedicated to specific topics. There are various Debian related channels on *irc.debian.org*, the biggest being channel and there are other channels more specifically aimed for developers.

Bug Tracking System

The Bug Tracking System¹ is an important foundation in Debian's development. Using tools like *reportbug*, users can easily submit bug reports and feature requests. The maintainer of the package automatically receives all bug reports and can then follow-up and ask for more information or immediately fix the bugs. When a developer uploads a new package to the unstable archive of Debian, they can automatically close bugs with the upload -- that way, users are informed that their bugs have been solved in that specific upload.

Package Tracking System

The Package Tracking System (PTS)² is a great way to see all kinds of information about a specific package at one spot. Different information is collected by the PTS and displayed on one summary page. Furthermore, the PTS allows users or developers of a piece of software packaged for Debian to subscribe to all bug reports filed against a specific package. This is a great way to stay informed of what is going on with a package and to help out -- when you know a solution for a bug, you can simply respond to the mail and it will be sent to the bug submitter, the maintainer of the package and get archived on the web so everyone has access to the useful information.

Summary

There are many means of coordination in Debian. The infrastructure, consisting of the mailing lists, the IRC channels and the Bug Tracking System among others, are a very solid foundation which enable efficient communication. Furthermore, there are various members of the project who are involved with coordination tasks. The constitution defines the roles and power of the Project Leader, the Project Secretary and the Technical Committee. However, there are many additional teams which have formed to fulfill a specific role. It is very often the case in Debian that organiza-

tional structure is generated implicitly over time when someone starts working on a job by themselves. Since all Debian developers are volunteers to Debian, nobody is paid for a specific task. Instead, everyone does what they like to do. Once someone has performed a specific job for some time and do a good job, other developers will recognize this and acknowledge their authority. The best advice for people who are interested in helping with Debian's development is therefore to simply get involved, and perform good work which is needed. Do not wait until someone assigns you a specific task, but find an area which needs help and get involved.

1. <http://bugs.debian.org/>

2. <http://packages.qa.debian.org/>

A Hacker's Diary

Greg Lehey <greg.lehey@auug.org.au>

This article contains the more technically interesting parts of my online diary, which you can find at <http://www.lemis.com/grog/diary.html>. There is no reason to believe that it has any specific relationship to the views of other AUUG members.

A number of URLs are given as relative to the directory in which the diary is kept. I haven't changed them, because it makes a mess of the format to have long, unbreakable URLs in two-column formats. They're all relative to the URL <http://www.lemis.com/grog/>.

Tuesday, 6 April 2004

Today my new Brother HL-2700CN colour laser printer finally arrived, after a wait of nearly two weeks. It had interested me because it came with an optional duplex unit (costing nearly as much as the printer itself, unfortunately). Installation took its time, particularly because of the terrible documentation, but the printer itself worked out of the box. I suppose it's typical of modern equipment: it had a network connection, and it sufficed for me to plug it into the network for it to go out and find a DHCP server and configure itself:

```
Apr 6 10:56:22 echunga dhcpd: DHCPDISCOVER
from 00:80:77:48:10:9c via x10
Apr 6 10:56:23 echunga dhcpd: DHCP OFFER on
192.109.197.98 to 00:80:77:48:10:9c
(BRN_48109C) via x10
Apr 6 10:56:24 echunga dhcpd: if IN A
BRN_48109C.lemis.com rrsset doesn't exist add
43200 IN A BRN_48109C.lemis.com
192.109.197.98: timed out.
```

The last appears to be a dynamic DNS request.

Setting up the *printcap* entry was correspondingly easy:

```
lp|Brother HL-2700CN networked printer:\
:rm=lp:sd=/var/spool/output/lp:\
lf=/var/log/lpd-errs:mx#0:
```

All it needed was a DNS entry for the printer and a queue directory to make things work.

Installing the duplex unit was a different matter. Apart from pretty terrible instructions, I was able to attach it without too much trouble. After installation, though, it still printed on one side only. The documentation said nothing about having to do anything special, and there was nothing on the web site's FAQ about the problem. Just before calling technical support, I booted up one of my laptops with the Microsoft software that Dell forced me to buy with the laptop, and installed

the management software. There, finally, I found a setting "DUPLEX ON/OFF", set to OFF. Set that, and it worked. While I was there, discovered at the same time all the protocols that the printer can handle. Quite a usable little program; it's really frustrating that the vendors have to supply it only for one operating system.

Things still aren't complete: for some reason, PostScript laser printers have a surprisingly long timeout on the last data, especially considering that PostScript tells you when it's done. On the final occasion, it took the printer something like 5 minutes to give up on the data and print the final page. I've found a couple of timeout parameters in the Microsoft-based program, but the help is non-existent ("no help found for this item"). It'll need some more playing around.

A number of people hate opinion surveys, but I like them. I feel that anybody who's as opinionated as I am should jump at the opportunity if somebody is actually interested in hearing these opinions. As a result, I've signed up on an opinion survey web site, Lightspeed (<http://au.lightspeedpanel.com/>). Today I discovered a mail message asking me to participate in a survey on food, something that greatly interests me.

When I tried to follow the link, though, I got:

```
Checking for scripting capability, and
browser/operating system..... If you are
not diverted in the next 10 seconds then
your browser is not able to support this
survey - this may be because you have
scripting disabled on your browser, your
browser is very old, you are not using a
Windows operating system, or your browser is
not Netscape or Microsoft Internet Explorer.
```

In other words, this survey is only open to users of Microsoft systems. To be sure, I checked the page I had received (shown reformatted for legibility):

```
n=navigator;
u=n.userAgent;
if((u.indexOf('Win')!=-1
||u.indexOf('Windows')!=-1)
&& parseInt(n.appVersion)>3
&&(n.appName=='Netscape'
||((n.appName=='Microsoft Internet Explorer')
&&u.indexOf('Opera')== -1)))
{
window.location.href=
'../scripts/mrwebpl.dll?start&' +
'panelID=a543&surveyID=18&
batchNo=4&project=B3qmw' + '&id=a543';
}
```

No question: this survey is only available to Microsoft users. How can an operator claim objectivity under those circumstances? I wonder if this doesn't offend against the Trade Practices Act.

Wednesday, 7 April 2004

More investigation of my profiling code. It's interesting to note that I wrote the code in one day, and minor issues are keeping me busy. It seems that the C library has two symbols for each system call, for example `open` and `__sys_open`. It's not clear why, since they're both the same address, but it makes things very difficult. Found out that the loader has a facility for wrapping system calls:

```
'--wrap SYMBOL'
Use a wrapper function for SYMBOL. Any
undefined reference to SYMBOL will be
resolved to '__wrap_SYMBOL'. Any undefined
reference to '__real_SYMBOL' will be
resolved to SYMBOL. This can be used to
provide a wrapper for a system function.
The wrapper function should be called
 '__wrap_SYMBOL'. If it wishes to call the
system function, it should call
 '__real_SYMBOL'.
```

That looks like being a possibility, but it'll have to wait until tomorrow.

My printer is now happily printing double-sided documents—even if they're parts of different jobs from different systems. They don't seem to have thought that one through, and I certainly can't find anything in the documentation. I should file a problem report.

Thursday, 8 April 2004

Spent most of the day working on the profiling code, and ended up choosing a different way to handle things. Discovered in passing that the system defines not two, but *three* different symbols for each system call, for example `open`, `_open` and `__sys_open`. Reading the loader documentation pointed to a different option, `--defsym`, which creates additional symbolic names. Ended up creating a library by loading a relocatable object and then archiving it:

```
gcc -O2 -g -Wall -D_FILE_OFFSET_BITS=64 \
-D_LARGEFILE_SOURCE -DDEBUG -I \
/lib_rocksoft_c/ -Wall -c -o profile.o \
profile.c ld profile.o -o libprofile.o -r \
--defsym _close=close --defsym _dup=dup \
--defsym _dup2=dup2 --defsym _fchdir=fchdir \
--defsym _fchflags=fchflags --defsym \
_fchmod=fchmod --defsym _fchown=fchown \
--defsym _fcntl=fcntl --defsym _flock=flock \
--defsym _fstat=fstat --defsym \
_fstatfs=fstatfs --defsym _pipe=pipe \
--defsym _pread=pread --defsym \
_pwrite=pwrite --defsym _read=read --defsym \
_readv=readv --defsym _write=write --defsym \
_writev=writev --defsym _access=access \
--defsym _chdir=chdir --defsym \
_chflags=chflags --defsym _chmod=chmod \
--defsym _chown=chown --defsym \
_eaccess=eaccess --defsym \
_getfsstat=getfsstat --defsym \
_lchflags=lchflags --defsym _lchmod=lchmod \
--defsym _lchown=lchown --defsym _link=link \
--defsym _lstat=lstat --defsym _mkdir=mkdir \
```

```
--defsym _mknod=mknod --defsym _open=open \
--defsym _readlink=readlink --defsym \
_rename=rename --defsym _rmdir=rmdir \
--defsym _stat=stat --defsym _statfs=statfs \
--defsym _symlink=symlink --defsym \
_sync=sync --defsym _unlink=unlink --defsym \
__sys_close=close --defsym __sys_dup=dup \
--defsym __sys_dup2=dup2 --defsym \
__sys_fchdir=fchdir --defsym \
__sys_fchflags=fchflags --defsym \
__sys_fchmod=fchmod --defsym \
__sys_fchown=fchown --defsym \
__sys_fcntl=fcntl --defsym __sys_flock=flock \
--defsym __sys_fstat=fstat --defsym \
__sys_fstatfs=fstatfs --defsym \
__sys_pipe=pipe --defsym __sys_pread=pread \
--defsym __sys_pwrite=pwrite --defsym \
__sys_read=read --defsym __sys_readv=readv \
--defsym __sys_write=write --defsym \
__sys_writev=writev --defsym \
__sys_access=access --defsym \
__sys_chdir=chdir --defsym \
__sys_chflags=chflags --defsym \
__sys_chmod=chmod --defsym __sys_chown=chown \
--defsym __sys_eaccess=eaccess --defsym \
__sys_getfsstat=getfsstat --defsym \
__sys_lchflags=lchflags --defsym \
__sys_lchmod=lchmod --defsym \
__sys_lchown=lchown --defsym __sys_link=link \
--defsym __sys_lstat=lstat --defsym \
__sys_mkdir=mkdir --defsym __sys_mknod=mknod \
--defsym __sys_open=open --defsym \
__sys_readlink=readlink --defsym \
__sys_rename=rename --defsym \
__sys_rmdir=rmdir --defsym __sys_stat=stat \
--defsym __sys_statfs=statfs --defsym \
__sys_symlink=symlink --defsym \
__sys_sync=sync --defsym __sys_unlink=unlink \
ar sru libprofile.a libprofile.o
```

There should be an easier way, but with this finally working, found out some interesting facts. This program handles a plethora of files, but the profiling code only treats individual file descriptors, which get reused all the time. It looks as if I should track individual files.

Saturday, 10 April 2004

Somehow all the TV web sites have conspired to produce no useful TV information today. I have a cron job that downloads information from eBroadcast (<http://www.ebroadcast.com.au/TV/>), formerly Sofcom, but today the page was empty. The front page announced:

```
"The Australian TV Guide is currently offline during maintenance. Please check back soon. In the meantime, take a browse around eBroadcast Australia."
```

"Maintenance" sounds to me like "crash". I had already found that ABC (<http://abc.net.au/tv/guide/>), who used to have quite a detailed list, have succumbed to the web designer syndrome and have hidden all their information behind a maze of twisty little web pages, all broken. Is this a sign of the times, possibly derived from the Microsoft GUI paradigm, where showing too much information too easily is apparently frowned upon? Or do they just want to make it difficult for people (possibly including eBroadcast) to download the information easily?

That would seem stupid, but then, the commercial channels all do the same: badly laid out pages with minimal information, often combined with JavaScript to make it impossible to follow links, and which open tiny little non-resizable windows without scroll bars, so small you can't read the bottom. Only SBS has a normal text programme, albeit with the rather strange option of sending you an SMS just before the programme begins.

The breakage is unbelievable. Spent some time investigating what it looked like under Microsoft. The answer: not as bad, though it was possible to completely lose the banner ads in the Seven Java popups, even with "Internet Explorer". But in general, the breakage under Microsoft isn't as bad, and I'm left wondering which of the browsers is broken. Based on prior experience, it should be Microsoft, but I haven't found anything obvious that points in either way.

While on the subject of broken software, discussed on IRC the issues I still have with my Brother HL2700-CN colour laser printer:

- The last page takes forever to print.
- The duplex unit doesn't understand the concept of jobs. It's possible to print two one page jobs from different systems and have them both on one side of the same sheet of paper.

Somebody suggested that I should install CUPS. I had understood that it would be able to configure the printer, but this proved to be incorrect. Installing CUPS is a good way, however, to lose faith in the Ports Collection. I found:

- The port didn't install the documentation.
- After installing the documentation, which involved rewriting the *Makefile*, it turns out that there is no man page for *cups(1)* or *cups(8)* anyway. The closest is *cupsd(8)*, the daemon, all of 35 lines long. There's nothing in the port installation process that tells you where to look.
- After installing the HTML documentation, discovered that the *index.html* file reference non-existent files. It seems that it's intended to be accessed with a web browser via the daemon (*http://localhost:631/*).
- Modern web browsers don't like the syntax *http://localhost:631/*: they want a fully-qualified name. So I put in the name of the system, and got an EPERM (permission denied) error.

- With some further investigation, discovered that the configuration file */usr/local/etc/cups/cupsd.conf* (referred to in the man pages as */etc/cups/cupsd.conf*) refuses even the local network, so it needed to be changed first. Not a big deal, except that there's nothing in the documentation to point to it.
- Finally I was able to configure the printers with a very ordinary web interface. It didn't do anything (apart from overwrite my */etc/printcap*) that *lpd* configuration couldn't do.
- Trying to print proved that the support programs, such as *lpr*, hadn't been installed. It's not clear why, but after installing them (in */usr/local/bin/* and */usr/local/sbin/*, they still didn't work properly. They also left the original *lpd* spooler files of the same name in */usr/bin/*:

```
# wh lpr
... Dec 18 2002 /usr/bin/lpr
... Apr 10 15:02 /usr/local/bin/lpr
```

Depending on the sequence of pathnames in the PATH variable, this means that you'd get one or the other.

Round about this time I gave up. There's no reason to believe that CUPS can help me with the issues I have, and it's just too painful. The real disappointment is that it doesn't seem to do anything much that *lpd* doesn't already do.

As if that wasn't enough, my Digitrex (*digitrex.html*) DVD player hung itself up again. That's six times in 22 days. I'm still wondering what the cause is, but one way or the other it's completely unacceptable. Something will have to happen soon.

Sunday, 11 April 2004

Daniel O'Connor along today to bring some beer and play around with DVDs. Spent some time trying to do useful things: in particular, I was interested to find out whether the DVD burner in his Inspiron 8600 could burn one of my DVD+Rs, since all attempts in the Digitrex had failed. Spent quite some time trying to read in a pre-recorded DVD, during which Daniel discovered that his DVD reader had great difficulties with one of my pre-recorded DVD+RWs, though the Digitrex did not. Somehow the reliability of optical media is marginal at best.

In the meantime, installed *mplayer* on a number of machines, finally finding one on which it

would work: it refused point blank on a system without audio hardware. Also discovered that you can mount a DVD as a normal file system. Here's a DVD+RW that I recorded

```
# mount /cdrom
# 1 -R /cdrom/
total 1
... 2048 Mar 22 08:05 video_rm
... 2048 Mar 22 08:05 video_ts
/cdrom/video_rm:
total 1
... 116736 Mar 22 08:05 video_rm.bup
... 8192 Mar 22 08:05 video_rm.dat
... 116736 Mar 22 08:05 video_rm.ifo
/cdrom/video_ts:
total 8775
... 14336 Mar 22 08:05 video_ts.bup
... 14336 Mar 22 08:05 video_ts.ifo
... 65536 Mar 22 08:05 video_ts.vob
... 102400 Mar 22 08:05 vts_01_0.bup
... 102400 Mar 22 08:05 vts_01_0.ifo
... 1073676288 Mar 22 08:05 vts_01_1.vob
... 1073676288 Mar 22 08:05 vts_01_2.vob
... 1073676288 Mar 22 08:05 vts_01_3.vob
... 1073676288 Mar 22 08:05 vts_01_4.vob
... 305528832 Mar 22 08:05 vts_01_5.vob
... 100352 Mar 22 08:05 vts_02_0.bup
... 100352 Mar 22 08:05 vts_02_0.ifo
... 1073676288 Mar 22 08:05 vts_02_1.vob
... 1073676288 Mar 22 08:05 vts_02_2.vob
... 1073676288 Mar 22 08:05 vts_02_3.vob
... 1073676288 Mar 22 08:05 vts_02_4.vob
... 305528832 Mar 22 08:05 vts_02_5.vob
```

Even stranger is that you can play back the *.vob* files with *mplayer*, though I haven't yet understood the real implications. I must spend some time looking at the structures on disk. On the one hand the data is there and can be played, on the other hand I'm told it's encrypted, and the issues of region code are well known. But what's to stop somebody copying these files? I was able to copy them to disk with no problems.

Monday, 12 April 2004

Spent some time catching up on yesterday's events—the Digitrex (*digitrex.html*) hung itself up again—and considered alternatives. The fact that *mplayer* works relatively well made me consider downloading stuff from the TiVo (*http://www.tivo.com/*), and found a version of *mplayer* which works with the TiVo, but didn't have the energy to try to work out where they had hidden the documentation. I think I'm building up to a big rant on how bad "Open Source" documentation is.

Tuesday, 13 April 2004

Got a couple of mail messages from people who thought I was being unfair to CUPS. Maybe they're correct, but both seem to have missed the point that it's the documentation that's the trouble, not the software. Specific issues were:

- *CUPS should be able to turn duplex on or off.* This seems like a possibility, but since I can't find anything about it in the documentation, it's only theoretical.
- *Debian GNU/Linux should be able to do it better.* This too sounds potentially possible, and I'll try it. But the writer goes on to say *Install instructions are on the site*, which suggests a similar problem exists there.
- *It's not a bug, it's a feature. There are no man pages because there's HTML documentation.* This one seems to be based on incomplete reading of my rant: first, there *are* man pages, and secondly you can't access the man pages until you have read them.

I'm still left with the feeling that CUPS documentation is inadequate.

Wednesday, 14 April 2004

Spent a little time this morning following up the claim that Debian installs CUPS better than FreeBSD. I was really expecting some improvement, but I didn't see it. I found exactly the same problems that I had had with FreeBSD. It *did* show that my accusations to the FreeBSD port were partially unjustified, though.

During the ensuing discussion on IRC, discovered some other interesting facts. The most important is probably that the printer has a complete web server installed in it, and that this web server appears to supply all the functionality of the maintenance program I talked about earlier, somewhat marred by the fact that some settings don't seem to show up correctly: although I was able to set a default Internet gateway, I couldn't get it to display correctly. Refreshing the display brought the home page, and returning to the TCP/IP config page showed a null gateway. Still, it's a far cry from the frustration of setting up serial printers 25 years ago.

In mail, some people on the UNIX heritage society (*http://www.tubs.org/*) were talking about John Lions' "Commentary on the Sixth Edition UNIX Operating System", which had been leaked to the *alt.folklore.computers* newsgroup just under ten years ago. Surprisingly, it wasn't on the net anywhere, so dragged out the copy I made at the time and put it up on this web site (*Documentation/Lions/*). I must actually get round to reading it some time.

See the first instalment of the commentary in this issue of AUUGN, starting on page 17. —Ed.

Also did some real work and managed a low-hanging 7% performance improvement.

Thursday, 15 April 2004

Heard from a number of people who have had problems with their Digitrex DVD recorder. I'm wondering if the things being sold on eBay are defective or rejects. Had a chat with the ACCC about the subject, and they took down a "complaint", but it will only make sense if more people complain.

Saturday, 17 April 2004

More mail on the subject of *CUPS* today. Although the correspondent didn't agree with my opinion of the *CUPS* documentation, he did point me to Eric Raymond, who does agree and has written his own account. His reasoning is very close to mine. A few days ago, I wrote: *Installing CUPS is a good way, however, to lose faith in the Ports Collection*. Eric writes: *It has proved a textbook lesson in why nontechnical people run screaming from Unix*. Eric had more stamina than I, and his document is well worth reading.

Sunday, 18 April 2004

Spent the morning reading more of Eric Raymond's article on CUPS (<http://www.catb.org/~esr/writings/cups-horror.html>). There are some interesting parallels: he has written it only recently, and one of the printers he was trying to connect was a LaserJet 6MP. On the other hand, there are differences. Eric also used the web interface. To judge by the lack of commentary on the subject, it appears to have worked out of the box. But how? How did he solve the chicken and egg question at the beginning? In general the article goes into more detail than I would have liked, but he does come up with some good suggestions about how to document software.

Monday, 19 April 2004

Another day spent with code analysis and documentation. There's an interesting tradeoff between basic functions and more comfortable ones built on top: the former are more difficult to use and (assuming a good implementation of the latter) more prone to error, but the latter are difficult to remember because of their sheer quantity. As functionality becomes increasingly complex, having such a large number of similar functions becomes more of a liability than an advantage.

Tuesday, 20 April 2004

Up early and into town to a board meeting of the IT Council of South Australia, the first one I managed to attend: last month I arrived 12 hours late. Presented AUUG, which evoked a certain interest.

Afterwards talking about "Open Source" to Peter Griffith, chairman of the SA Chapter of ACS. I told him that I couldn't stand Microsoft "Windows XP", because it was so difficult to find my way around a GUI. He agreed: he uses Microsoft NT 2000 "Server". After thinking about it, it occurs to me that that's exactly the problem: GUIs are sufficiently non-intuitive that you have to learn them, and once you've learnt one, even a small difference is difficult.

Wednesday, 21 April 2004

At work, we still haven't got through our discussions on code structure and documentation. Exchanged a grand total of 35 detailed mail messages on the subject, without coming to a complete agreement. The big issue at stake is: where and how do you document externally visible functions? My view is that there should be external documentation (typified by user guides and man pages, though man pages aren't necessarily the best example of such documentation), and that the commentary in the source code should be geared to the needs of the maintainer. As a result, the obvious place to describe how a function works is directly in front of the definition. For example, from Vinum (<http://www.vinumvm.org/>):

```
/*
 * Build up a request structure for writes.
 * Return 0 if all subdisks involved in the
 * request are up, 1 if some subdisks are not
 * up, and -1 if the request is at least
 * partially outside the bounds of the
 * subdisks.
 */
enum requeststatus build_write_request
    (struct request *rq)
```

The others consider that the purpose of this documentation an externally visible function should be documented for the user, and that it should thus be in the header files, since that's the only part of the source that the user should be allowed to see. From my viewpoint, this is inconvenient because:

1. The user shouldn't be reading the header files just because he has them. He should be reading the documentation.
2. This makes two different rules for where to document a function, depending on whether they're externally visible or not.
3. The maintainer doesn't normally need to read header files unless he's changing the function itself, in which case he needs to get it in sync.
4. Most programming tools, such as *ctags* and *etags*, bring you more naturally to the function definition rather than to the declaration.

5. Having to look for things in two different places is just plain inconvenient.

None of this is a show-stopper, of course, but it's surprising how difficult it is to come to an agreement about it. Spent some time investigating alternatives to *etags*, and came up with GNU Global (<http://www.gnu.org/software/global/global.html>), which looks something like *etags* on steroids. It installed nicely and works pretty much like *etags*. I'll need to work out which of the additional functionality it provides is really of use; converting source files into hyperlinked web pages doesn't seem very useful to me, for example. One thing it doesn't do is to provide any easy way to get to the function declaration.

Saturday, 24 April 2004

Spent some time investigating computer-based TV systems. It seems that issue 8/2004 of *c't* (<http://www.heise.de/ct/>), which I haven't received yet, has a number of articles on the subject, including a project based on Linux and the Hauppauge WinTV PVR 350 which seems to be the current favourite card. Spent some time looking for sources in Australia, which proved to be very expensive; in Singapore they're only half the price, so it may well make sense to have one sent here. It seems that there are still no FreeBSD drivers for the card, so there would be a bit of work involved.

Also considering film scanners; I have thousands of slides and negatives, though I wasn't able to find the ones that I was really looking for. Most better quality scanners now seem to have facilities for negative scanning, so it's really a matter of deciding which to buy. Sent out a message to a number of mailing lists. I wonder how they'll do with the colour negatives of the 60s, which are much more orange than nowadays.

Sunday, 25 April 2004

Tidied up the HiFi cupboard, which was in need of it. In the process managed to crash *sat-gu*, my downlink box, not once but twice: first I accidentally plugged a vacuum cleaner into the UPS, which promptly gave up the ghost, and the second time I managed to dislodge a power cable. That must be what happened to Yvonne when I was in Singapore last month.

Monday, 26 April 2004

In the afternoon, thought it was about time to try out DragonFlyBSD, and downloaded an ISO snapshot of the current development version. Installation was less than edifying: for whatever rea-

son, the kernel didn't like the DVD drive from which it had been loaded, and I had to replace it with a CD-ROM drive. After that, the installation proceeded easily enough, considering there isn't any installation program. It does start a live system from the CD, though, along with an *ssb* daemon. *ssb* wants all kinds of security things, of course, and inside my firewalled local network I thought that I could risk *telnet*. Not so the dragonflies, it seems: there appears to be no *telnet* in the distribution.

Tried adding a normal user, somewhat difficult to do with a read-only file system. Ended up giving myself the home directory */tmp/grog*, and with a bit of help from *dbcp* I was able to *ssb* into the system and do everything from my normal environment. Installation proceeded well enough, I thought, but at the end it didn't want to boot. More to look at tomorrow.

Tuesday, 27 April 2004

Back to work today, and wanted to get right into doing some performance testing. Despite a relatively minor change, which I expected to cause almost no improvement, performance went to hell in a handbasket, and eventually panicked the machine repeatedly; it looked like something to do with core dumping in a KSE environment. Spent some time looking at that and decided to install 4.10 Beta on one machine (*zaphod*, called *daemon* when it's running release 4), and to upgrade *beeble* to the latest -CURRENT.

Both took far longer than they should do. I've done this hundreds of times before; why should it be so difficult every time? People discussing things like total cost of ownership should be looking at these issues (not that it seems that commercial vendors are doing any better, but why should we measure things by them?). By the evening was able to reboot both machines, but it looks like I need Yet Another *mergemaster* run. sigh

I've been advertising SPF (<http://spf.pobox.com/>) records in my DNS for some time now, but yesterday Yvonne got a reject based on it:

```
<ers@mubo.de>: host
ipev-mail.saar.de[192.109.53.24] said: 550
5.7.1 <yvonne@lemis.com>... Please see
http://spf.pobox.com/why.html?sender=yvonne
%40lemis.com&ip=192.109.197.80&
receiver=ip-comserv.saar.de (in reply to
MAIL FROM command)
```

I had set up my SPF TXT record using the "wizard" at <http://spf.pobox.com/wizard.html>, which told me:

Let's set up SPF records for lemis.com Begin

lemis.com's IP address is 192.109.197.137 (echunga.lemis.com).
Does that server send mail from lemis.com? a yesno

This wizard found 5 names for lemis.com's MX servers.
MX servers receive mail for lemis.com. Do they also send mail from lemis.com? mx yesno

Do you want to just approve any host whose name ends in lemis.com? ptr yesno

Do any other servers send mail from lemis.com?
You can describe them by giving "arguments" to the a, mx, ip4, and ptr mechanisms. To keep the wizard short we left out ptr: but it works the same way.

IP networks can be entered using CIDR notation, eg. 192.0.2.0/24

Could mail from lemis.com originate through servers belonging to some other domain? If you send mail through your ISP's servers, name the ISP here. include:

Do the above lines describe all the hosts that send mail from lemis.com? all yesno

lemis.com IN TXT
"v=spf1 a:a:ozlabs.org a:192.109.197.0/24 -all"

v=spf1 a:a:ozlabs.org a:192.109.197.0/24 -all

v=spf1	v=spf1	This identifies the TXT record as an SPF string.
<input checked="" type="radio"/> a	a	lemis.com's IP address is 192.109.197.137 (echunga.lemis.com). That server is allowed to send mail from lemis.com.
<input checked="" type="radio"/> a:	a:ozlabs.org	ozlabs.org is also allowed to send mail from lemis.com.
<input checked="" type="radio"/> a:	a:192.109.197.0/24	192.109.197.0 is also allowed to send mail from lemis.com. So are any other servers in the same /24 subnet.
<input checked="" type="radio"/> all	-all	No other servers are allowed to send mail from lemis.com. This is a good default.

On the other hand, the web error page said:

```
Why did SPF reject my mail?
Feb 6 2004
SPF is a standard extension to Internet
email which protects people from email
forgery. ip-comserv.saar.de rejected a
message claiming to be from
yvonne@lemis.com. ip-comserv.saar.de saw a
message coming from the IP address
192.109.197.80 which is
blackwater.lemis.com; the sender claimed to
be yvonne@lemis.com. However, lemis.com has
announced using SPF that it does not send
mail out through 192.109.197.80. That is
why the mail was rejected.
```

After a lot of investigation, discovered that the SPF wizard is at fault. It didn't catch a non-obvious error that I made: when setting address ranges, you should put them in the "IP networks" field (ip4:), not the a: field, which is reserved for server names. The wizard didn't notice that problem. The correct string should have been "v=spf1 mx ptr a:ozlabs.org ip4:192.109.197.0/24". Another indication that you should not put your trust in "wizards".

Wednesday, 28 April 2004

Spent the day trying to get my system upgrades completed, without success. Once upon a time it was as simple as make world, but nowadays every upgrade is a minefield. The FreeBSD project really needs to attend to this issue. It wasn't helped by a power failure in the middle of everything: I need to get some new UPSes.

Gave up on daemon, which kept SIGSEGVing login, and went back to running it as zaphod while I tried to coax beeble back to life: it had decided to return EPERM on any network packet. Got some work done, but not as much as I would have liked.

Thursday, 29 April 2004

More work on upgrading systems today, and finally got things up to date. It seems that the networking problems were due to a change in the startup procedures that didn't configure the firewall code correctly, so it ended up with only the one rule:

```
65535 0 0 deny ip from any to any
```

I shouldn't have had firewall code on that box anyway, of course, but it seems that configuration file changes that make such a difference should be avoidable.

Tried doing an "upgrade" install on daemon to bring it back to 4.9, so that I could restart the real upgrade, but after a successful upgrade it still SIGSEGVed on login, so ended up doing a cold install. This is really unacceptable.

Friday, 30 April 2004

Off to buy some hardware, including a Canon 9900F Scanner (http://www.canon.com.au/products/home_office/scanners/scanners_low_medium_volume/canoscan9900f.html), with which I intend to scan in all my old photos.

After that to visit Peter Cassidy, who has a SPARCstation 5 which he can't boot. Proved to be an invalid ID PROM contents. We were able to boot from the disk, but of course we didn't know the root password. What a pain that even single-user mode requires a password. Noted also that software as recent as Solaris 2.5, on the disk, still starts a login with the erase character set to DEL. Considering that the main keyboard doesn't generate that character, this is really unbelievable. Took the machine with me; I'll play around with it for a while, though with only 32 MB of memory it might be better to cannibalize it and make one faster machine with flame.lemis.com, also a SPARCstation 5

Saturday, 1 May 2004

Planned to spend today with my new toy, the Canon 9900F Scanner, and so I did. It wasn't quite the way I had intended, though. I have gradually given up on programs like SANE (which, I contend, contains a typo in the expansion: it should read Scanner Access Not Easy). On the one hand, it severely limits the choice of hardware, and on the other hand scanners already come bundled with lots of software. Yes, it's in Microsoft eye candy format, but it should be easy to use.

The truth is a little different. Just installing the software must have taken 100 clicks, including requiring me to accept eight end user license agreements, presumably all different. I suppose they do illegal things like disclaim all notion of “fitness for purpose”. But I could only see a little of them, and the parts that I saw seemed innocuous enough.

Running the software was another matter: it looks as if it’s broken. The user interface certainly is. To read in a set of negatives or slides, you place them with the emulsion side up (unlike in an enlarger, and a detail that was obviously too technical for the superficial manual to mention) and start a new program (“scanner driver”, I suppose), which blocks the previous program until it’s finished. This driver decides for itself where the boundaries of the images are—with about 70% accuracy, meaning that in any one set of 8 slides about 2 are incorrectly framed. I can’t see any way in this toy interface to influence this action.

I had been warned about the scan speed, but without accurate estimates. I haven’t tried timing it, but my best bet is that each slide takes about 3 minutes to scan, about 25 minutes total for 8 slides (this is at 1600 bpi, which gives resolutions roughly equivalent to my other photos (*photos.html*). When they’ve finally finished scanning, you have to save them—one at a time, with multiple mouse clicks for each photo. By default it tries to save them in some proprietary format, and you have to select JPEG (“JPEG”) format every time you start the program. It also forgets its numbering scheme when you stop it, so if you run the program multiple times, you will end up having to store the files in different directories or rename each individual image for the subsequent scans. At least it remembers resolution parameters, and I was able to set up Samba to store the images directly on a real computer.

Got so frustrated that I checked if I could use SANE with the device. FreeBSD recognized the Firewire connection, but of course SANE doesn’t support it.

The whole thing reminds me yet again of how bad software has become of late. The “Microsoft Mentality” seems to accept software that has bugs of this nature, as long as it’s “intuitive” (i.e. follows the eye-candy conventions). The lack of published standards means that I can’t just go and build a driver that will work properly. This problem is probably more important than all the issues that people are trying to legislate about “Open Source”: free software can’t be much good if you have to reverse engineer everything you do.

Sunday, 2 May 2004

More work scanning slides. The software is Just Plain Broken. It has great difficulty recognizing boundaries, and even the online manual is wrong: it’s in HTML, and the main file has the intuitive name *canoscan.btm* (sic). In it is stuff like:

```
<TD WIDTH="95"><IMG \
SRC="CanoScan/img/header_logo.gif" \
WIDTH="95" HEGHT="34" HEIGHT="28"></TD>
<TD WIDTH="631"><IMG \
SRC="CanoScan/img/hd_01.gif" WIDTH="631" \
HEGHT="34" HEIGHT="28"></TD>
```

HTML tidy finds 53 errors in this document, including proprietary extensions. The real problem, though, is that there is no directory *CanoScan*; it’s called *canoscan*, so nothing worked until I put in a symlink. I’m continually appalled by the poor quality of software nowadays.

The manual did tell me how to get around incorrectly recognized negatives, though: turn off thumbnails, which gives the entire area. Unfortunately, contrary to the documentation, and probably due to a bug, it also turns off retouching, which left some very dirty negatives. What a pain. Despite everything, managed to scan in two 36 exposure slide films, which, at an average scan time of 5 minutes per slide, took all day. The first film was taken on the Bay of Bengal (*Photos-19670421.html*) on the trip from Penang to Madras, and the second one from Madras to Agra (*Photos-19670504.html*).

Tuesday, 4 May 2004

To an AUUG board meeting in North Sydney, where we discussed—yet again—the future of AUUG, taking all day. Lunch at a Thai place which would have been excellent had it not been for their emphasis on creamy sauces. At least they had nice presentation, including chilis cut in the shape of a flower (*Photos-20040504.html #Chili-flower-1*).

In the evening was the AUUG NSW chapter meeting, with John Terpstra, who is doing a road show over the next couple of weeks. Started off with an interesting viewpoint on software marketing economics, using such fashionable terms as “mercantilism” and “feudal”, but I had to leave for the airport before he was done.

Wednesday, 5 May 2004

Today spent some time thinking about storage hierarchies, and came up with a use for *Monkey*, a B-tree storage system I wrote in the early 90s, also coincidentally the last program of any importance

that I wrote in C++. I'm still wondering why I stopped using C++, but the fact that the language kept changing certainly has something to do with it, as does the fact that it wants you to do things such as memory allocation and exception handling its way.

Today was a good example: I could no longer compile code that worked fine 8 years ago. It seems that the syntax of the friend declaration has changed, and I spent a lot of time trying to work out why it was complaining about constructs like

```
struct sigaction newdisp = {&caught, 0, 0};
...
utility-lib.cc:42: warning: aggregate has a
partly bracketed initializer
```

It looked as if the caught was the culprit, since struct sigaction is defined as:

```
struct sigaction {
union {
void (*__sa_handler)(int);
void (*__sa_sigaction)
(int, struct __siginfo *, void *);
} __sigaction_u; /* signal handler */
int sa_flags; /* see signal options below */
sigset_t sa_mask; /* signal mask to apply */
};
```

So I changed it to the following, and got the following error messages:

```
struct sigaction newdisp = {{&caught}, 0, 0};
.
utility-lib.cc:42: warning: aggregate has a
partly bracketed initializer
utility-lib.cc:42: warning: aggregate has a
partly bracketed initializer
```

In other words, it was happy with that one, and it was complaining about something else. I tried a number of different forms, but didn't find out what the problem was.

I still don't know. If you, gentle reader, know the answer, please let me know.

While doing that, also looked at some of the problems I was having on Monday. Upgraded *zaphod*—yet again—to the latest FreeBSD-CURRENT, and also revived *beeble* (dual processor Celeron machine) as *brynbild*, a Debian box that I set up last July and upgraded the kernel to Linux 2.6.5 in the hope that we can get some useful information about the effectiveness of threading under Linux.

Since that wasn't keeping me busy either, also took a look at the SPARCstation 5 that Peter Cassidy gave me on Friday, and tried to install Solaris 8 on it. It proved that the CD-ROM drive he gave me had the wrong SCSI ID (7, which looks dangerous), but even after setting it to the regulation

ID 6, I couldn't read from it. Fortunately I had an identical one myself, and it worked. After over 20 minutes booting from the CD-ROM, I got the message:

```
64Mb of memory is required for Solaris
32 Mb was found.
Exiting
```

Took the box apart and confirmed that the memory chips looked identical to the SDRAMs of a few years back, of which by coincidence I had a 128 MB chip over.

On trying to insert the 128 MB chip, discovered that the index grooves are offset by approximately a millimetre, presumably for a good reason. Put the box together again and tried to boot NetBSD, somewhat hampered by lack of documentation, in particular the name of the file to boot. Tried again with OpenBSD, and was able to mount the root file system of the existing installation (Solaris 2.5, which apparently is able to get by with less memory. Fixed the */etc/passwd* and */etc/shadow* files, and was then able to boot and log in to the machine.

That didn't help much: there's no compiler or anything on the box, and so I couldn't do very much. I suppose I could try installing the compilers from Solaris 8, but who knows what problems I'd have there. Time for a bigger box, preferably with multiple processors.

Still fighting this brain-dead Canon 9900F Scanner. It still only recognizes about 70% of the slides I put into it, and puts ridiculous frames around the ones it doesn't recognize. This is a real pain to use.

Thursday, 6 May 2004

Less activity than yesterday, but still got some useful work done. Finally worked my way through the C++ code and got it to work, not without running into another problem:

```
cc -o fconv fconv.o -g -L. -lmonkey -lm
fconv.o:/home/monkey/monkeyfs/fconv.cc:190:\
undefined reference to\
'__gxx_personality_v0'\
./libmonkey.a(utility-lib.o):/home/monkey/\
monkeyfs/utility-lib.cc:42: undefined \
reference to '__gxx_personality_v0'\
(many repeats)
*** Error code 1
```

This proved to be due to the rather unobvious fact that I was using the *cc* driver for the compiler instead of the *g++* driver. This is a real pain: all this code used to work. After changing the link driver, it did work, but now I need to adapt it to where it needs to go.

Friday, 7 May 2004

More work on *Monkey* today, and I'm now at the point where I can start modifying code. The big change is from C++ to C, which proved to be less difficult than I thought, mainly because I got used to the idea of not using many C++ extensions fairly early on. It's a pity, though: C++ has a number of features, notably the implicit use of a class instance, that would work equally well in C.

Finally got fed up with the Canon 9900F Scanner, which has a driver that can't identify where the slides are in the frame. You'd think that it would be able to do at least that, since it's apparently written exactly for this scanner. After establishing that I did, indeed, have the latest version of the driver, called up their support line. I would much rather have sent an email, but they didn't give me that choice. Spoke to Ejaz, who initially had difficulty understanding what the problem was, but who asked me to "uninstall" the driver and reboot. OK, this is Microsoft XP, so I did that, and was astounded by the fact that the driver wasn't gone after all. I wish I knew what these Microsoft terms really mean; I've given up hoping that they might use terms at their face value.

Of course, the problem didn't go away, and so Ejaz first suggested that I return the scanner, because it was obviously defective (because the driver can't identify the edges of the slides!). When I suggested that he was misunderstanding the situation, he suggested that I should install Adobe PhotoElements, a stripped down version of PhotoShop which also came with the scanner, and which I hadn't installed because it asked for a serial number. Finally found that and installed it, but of course it didn't make any difference, since it called the same driver. The only thing I did get to notice is that PhotoElements require even more mouse-pushing: for *every single image* I have to tell it to save in JPEG format and not in its own proprietary format. There are preferences, but such an idea as specifying a preferred directory and save format doesn't seem to be one of them.

Then Ejaz suggested that I should try connecting the scanner to another machine and see if it worked there. I got completely fed up and asked whether, if that also didn't work, he would then ask me to try running it in a different state to see if that made any difference. It seems that second-level support won't even look at it unless people go through this ridiculous rigmarole. Finally agreed with him that I'll document the whole issue (amongst other things, here on this web site) and send him an email, which he can then forward to their second-level technical support.

Saturday, 8 May 2004

Came into the office this morning to discover that *wantadilla* had panicked. I no longer had the original kernel tree for this system—it dated back to October 2002—so gave it up as a bad idea and upgraded to a modern -CURRENT, which worked better than I expected.

Apart from that, spent most of the day scanning in old photos and documenting them. Interestingly, though it takes about 4 minutes to scan a single photo, I wasn't idle. Setting up this scanner to recognize the slides is an iterative process: I have never yet had it recognize the frames of all 8 slides correctly, though in some cases it just adds a black frame along one or two sides. In other cases, it gets things completely broken, as the incorrect frame in the following image shows:

Follow the link to get something visible. The final screen shot is in original size, but that's only 1024x768, and the images are made much smaller than that, so it's very difficult to recognize them. This image, of the uninterpreted slides, shows clearly that the software has placed the frame incorrectly. The corresponding interpreted ("thumbnail") view follows this frame, making it impossible to use.

Also spent some time playing around with the *Makefile* for building my photo pages, and it now creates thumbnail images and updates the index file (*photos.html*). Still, it's very frustrating to use this scanner.

Sunday, 9 May 2004

Another day spent seemingly only scanning in photos, though I spent a couple of hours working out a Problem Report (*Canon-breakage.html*) for the Canon 9900F Scanner, in the process discovering a few more issues about the nature of the problem: it has a box for selecting the kind of film, but only in raw display mode. Unfortunately, it ignores the box, and tends to reset it. What a crock. I wonder if anybody else has had problems of this nature; Ejaz says no, but that's to be expected.

Monday, 10 May 2004

Spent most of the day converting my Monkey subset from C++ to C. It's painful business, and it shows that C++ really does have a number of advantages over C. If only there were a way to use the clever object semantics without the environmental bloat, something like a C+. In particular, it's a real nuisance that there's no way to incorpo-

rate a structure into another structure without leaving its wrapper behind. In C++, I can write something like this:

```
struct fcb
{
/* magic name of function friendly to monkeys */
friend int Tarzan (Monkey *, int, void *);
int32 flags;
.
struct Ks_info: public fcb
{
.
class Monkey: public Ks_info
{
.
.
```

After that, within a Monkey class function, I can refer to the fcb flags simply as flags. By contrast, in C I have to write:

```
struct fcb
{
/* C has no friends */
int32 flags;
.
struct Ks_info
{
.
struct fcb fcb;
};
struct Monkey
{
.
struct Ks_info ksinfo;
```

To refer to the fields at all, I need an explicit additional argument *this* to the functions, and I need to refer to the same flags field as *this->ksinfo.fcb.flags*. There's probably no difference in what the compiler generates, so it's not a question of efficiency, simply legibility.

In the evening spent some time updating my Canon 9900F problem report (Canon-break-age.html). Discovered that there *is* a way of telling the driver what kind of films are in the scanner. The problem is, it doesn't just ignore the information, it changes it.

This is obviously an interesting matter: I've been getting a lot of web site hits on the subject. I wonder how many other people have similar experiences but have been fobbed off by the support people.

Tuesday, 11 May 2004

More work on converting *Monkey* to C today. It's getting faster, but it still takes forever. By the evening I had only one source file to go.

I haven't made up my mind on how bad it looks in C. I end up with lots of things like this:

```
if (! (M->K.flags & OPENFLAG))
return errno = MONKEY_NOTOPEN;
startreq (M);
Monkey_FlushBuffers (M);
Monkey_cache_dealloc (M);
M->K.flags &= ~ (OPENFLAG | CORRUPT | DIRTY);
if (M->K.flags & FCB_CHANGED)
Monkey_flush_fcb (M, 1);
```

In C++, the code looked a lot simpler:

```
if (! (flags & OPENFLAG))
return errno = MONKEY_NOTOPEN;
startreq ();
FlushBuffers ();
cache_dealloc ();
flags &= ~ (OPENFLAG | CORRUPT | DIRTY);
if (flags & FCB_CHANGED)
flush_fcb (1);
```

The examples above and immediately below have had the comments stripped. It's too painful to try to fit them into the columns of AUUGN. They're included in the original web page.

The big changes here are the implicit pointers and the naming of the functions. The question is, is this a good idea? It's very likely that the code generated by both these fragments would be identical, but the C++ hides a lot of the detail. I can't make up my mind whether it's better hidden or not. And yes, I've deliberately shortened the pointers and coalesced a structure; for example, originally I had written *Monkey->ksinfo.fcb.flags* instead of *flags*, but that seemed too much, so now it's *M->K.flags*.

In fact, the original was much larger. I'm stripping out lots of functionality. *Monkey* does all sorts of nice things that we don't need in the current application, including multiple alternate keys, compression, field descriptions, auditing and user-defined collation sequences. In fact, the original C++ code looks like this:

```
if (! (flags & OPENFLAG))
return errno = MONKEY_NOTOPEN;
startreq ();
FlushBuffers ();
cache_dealloc ();
if (flags & (ICOMPR | DCOMPR))
free (compbuf);
flags &= ~ (OPENFLAG | CORRUPT | DIRTY);
if (! (flags & ISALTFILE))
{
if (altkeys)
{
altkeyeof = altfile->eof;
altkeysindexlevel = altfile->indexlevels;
}
if (flags & FCB_CHANGED)
flush_fcb (1);
close (fcbfile);
if (fdsize)
free ((char *) fd);
if (altkeys)
{
altfile->Close ();
free ((char *) ak);
}
}
if (flags & (MY_COLLATION | ASCII_COLLATION))
free ((char *) collation);
```


That has nothing to do with C++, of course, but it's interesting to strip down a program for once. Usually you incrementally add things, and you don't see how much bigger it gets all at once. You do when you remove the things again.

Playing around with Microsoft, trying to print from "Word". I had forgotten that Microsoft needs a "driver" (apparently a definition file) for printers, and I had mislaid the CD-ROM that came with the printer, so had to print to files instead and get FreeBSD to print.

Setting up networked printers under Microsoft is really strange, though: a popup told me: "To set up a printer that is not attached to a print server, use the "Local printer" option". I also found another amusing pop-up that asked me: "Always trust content from Microsoft Corporation?".

Wednesday, 12 May 2004

Spent more time than I would like working on administrative stuff today. One of the things about my new job is that I'm coming more in contact with the Microsoft world, and I keep getting documents in Microsoft "Word" format. For some reason, possibly related to the last update of *wantadilla*, OpenOffice decided to crash on me with a SIGSEGVs, so had to upgrade that. Jonathon Coombes tells me that it's possible to get it to do things in command-line mode, but I haven't seen much evidence yet. In any case, was obviously too stupid to set up printing on OpenOffice as well, so ran that through my FreeBSD file system as well.

Thursday, 13 May 2004

In the morning, spent some time thinking about data structures, and came to a breakthrough in understanding how *Monkey* could be the solution to just about every problem. Then continued working on the *Monkey* conversion, and finally got it to compile. Even had time to spend a bit of time rearranging things to better fit the environment in which it will be living.

In the afternoon into town for a meeting. We spent three hours talking about a number of things, including of course the use of *Monkey*. That topic proved to be less successful: without a white board, it was difficult to explain the concepts. I think it would have been difficult even with the white board, and I left with the action item to write up *Monkey*, which I'm going to defer until I have something to show for myself. I did come up with a couple of interesting suggestions, though.

Friday, 14 May 2004

How many things I have to do! Getting this *Monkey* prototype working looks so simple now, though inevitably it will take longer than it seems.

Unfortunately, didn't get very far down that road. Follow-ups to yesterday's meeting took up a surprising amount of time. Still, spent some time going through the remaining *Monkey* code and reviewing it. The original *Monkey* used three files:

- The main file, which had a Microsoft-like convention of ending in the characters `.KSF`, for example *Monkey.KSF*.
- A separate file for the alternate key file, structurally identical to the main file, but with the ending replaced by `.AKF`, for example *Monkey.AKF*.
- An FCB file with basic information about the other two files, including block and record lengths, and key information. It had the base name of the other files stripped of the endings, i.e. *Monkey*.

In particular the FCB file seems to have outlived its usefulness. I developed *Monkey* initially under Microsoft MS-DOS, and it's a hangover from those times. The alternate key file still does have a justification, since it contains different records from the main file, and the block sizes can be different. I think it's too hard to store different block sizes in a *Monkey* file, but if the block size is the same, there should be no reason why we can't store more than one tree in a single file.

Sunday, 16 May 2004

Tried to install the GIMP today. It failed with too old a version of X, not the first port to do so; it seems that something has changed in the X rendering libraries. Somehow the Ports Collection is in trouble; all sorts of dependencies failed to compile, and I ended up recompiling the entire X distribution several times, without getting past the problems.

Monday, 17 May 2004

Continued with the rebuild of my ports; there has to be an easier way. Several of them didn't build, and I ended up installing the GIMP from a package. Spent the entire day running *portupgrade*, unfortunately in interactive mode, and ended up having to answer questions about ports I had never heard of. When it finally came to upgrade *gcc*, I hit `^C` and broke things completely:

```
/usr/local/sbin/portupgrade:35:in 'require': \
No such file to load -- pkgtools (LoadError)
from /usr/local/sbin/portupgrade:35
```

I wonder how to recover from that one.

Tuesday, 18 May 2004

Finally got round to starting on my userland test tool. Made relatively good progress in setting up a framework, incorporating a command language parser that I had originally used in *Vinum*, and which I had used in other programs too. The problem was that the keywords appear in different places in different guises, and getting them in sync was a pain. Solved that with a little shell script which generates the necessary bits and pieces, and by evening I had the framework set up and compiling and “running” cleanly. All I need now is for it to do something.

Sent another message to Ejaz about the Canon scanner problems (*Canon-breakage.html*), and asking for immediate action. This did bring some reaction: a reply from Ejaz, stating that he had received my message, but not addressing the other issues (isn't that so often the way when people answer the wrong way round?), and a call from Mitchell in Sydney, who told me that he had been able to reproduce the problem, that he had escalated it to Japan, and that he hoped for a response within a week. Interestingly, he drew a lot of attention to the fact that he had tried it on two different computers with two different mother boards (both presumably Microsoft-based) and also on an Apple. Maybe they do have a lingering suspicion that it could be a problem with the computer hardware.

Also managed to get my ports sorted out, sort of. Whatever breakage I did with *portupgrade* yesterday seems to be unrecoverable, and I ended up building packages on *adelaide* and installing them on *wantadilla*. There must be an easier way: one would seem to be to start from scratch with the ports when installing a new system, and only installing the top-level ports (i.e. the ones you explicitly install in the first place), as opposed to all the myriad support ports. There should be a way to note which are which; I'll do some thinking.

Wednesday, 19 May 2004

Continued with the coding today, and made fair progress. The original *Monkey* has now shrunk to about 10% of its initial size, reminding me of the quotation:

Hoare's Law of Large Problems:

Inside every large problem is a small problem struggling to get out.

Still a number of global changes happening, in particular to the cache layer, which will probably require further attention. It proves to be interesting to put the entire source for *Monkey* (5 files) into a single file for editing. When it's more or less in the shape I want, and I actually get round to testing it, I can take it apart again.

Thursday, 20 May 2004

One of the first things I do every morning is to check my beer stocks, and when brewing also the temperature and fermentation rates of the beer. For that I use *brewmaster.lemis.com*, an ancient Dell laptop prototype that I keep in the kitchen and connect to the network via wireless. It runs OpenBSD, mainly because that was the first system that would understand its PCMCIA bus. The files are all stored on *wantadilla* and accessed via NFS. Today, the network hung, and I couldn't get the interface back up again. Checked with *firefly*, Yana (*/yana/*)'s machine, which was connected to the same access point, and that worked fine. Rebooting *brewmaster* didn't help, nor did changing the wireless card. Finally I rebooted the access point, and things came back to life. It seems that the AP only failed partially.

The AP runs Linux (*diary-sep2003.html#8*), as do two other machines in its immediate physical proximity: *tivo.lemis.com*, a TiVo, and *sat-gw*, a Red Hat box which runs my satellite downlink. I don't know what it is about them, but they're three of the biggest problem machines I have.

Had intended to continue with my new program today, but realized that the cache abstraction was still far too closely coupled with *Monkey* data structures, and spent the day tidying that up as well. It looks a lot cleaner now, but I'm still left with the conviction that it's too primitive. On the other hand, the kernel already provides buffer cache, so we're not looking for absolute performance here; as long as the cache search times don't distort the relationships, there should be no problem.

Friday, 21 May 2004

Continued working on my program today, spending most of the time looking at how to modernize *Monkey*. It was based on a 30 year old product. It was good, but looking at things from a modern standpoint makes it look a little old-fashioned in some areas. Spent all day redesigning and reimplementing the data and index block structures.

Monday, 24 May 2004

On with the work on my program today, and spent more time tweaking the cache implementation. I must resist fixing things that don't make much difference; the important thing is to get the thing running (or at least stumbling) at all. Made reasonable progress nevertheless, and got as far as starting to test the initial functionality. It currently looks like I overwrite something *malloced* almost before I start. Maybe it's time to investigate Valgrind (<http://valgrind.kde.org>).

Tuesday, 25 May 2004

Continued work on my initialization code today and found that my *malloc* corruption was non-existent: I had forgotten to initialize a member of the superblock, so called *malloc()* with a 0 length—and it returned a pointer 0x800. Strange, and non-intuitive, but it seems that the standard allows it.

Apart from that, a typical day of debugging, and by the end of the evening had some data structures on disk to show for it. Also created some *gdb* macros to look at the data structures. Programming in *gdb* macro language used to be like pulling teeth. Now it's a lot better: it's just annoying.

Wednesday, 26 May 2004

Further work on my project today, and made faster progress than intended: I hadn't planned what to do next, and spent some time doing that. Nothing much else of interest.

Thursday, 27 May 2004

On with my project today. Ran into what appears to be a day 1 bug in *Monkey*: if a record exists with a key shorter than the total key length (i.e. the record is short and the key at the end is only partial), it was not possible to insert a record with the same initial key but of different length. This means that if the file contains a record with the key *foo*, it was not possible to create entries *fo* or *foox*. I don't understand why I didn't fix it: there was a XXX as a comment at the comparison, and it was easy enough to fix (if the keys are the same for their common length, compare their lengths instead).

Also ran into some problems I had caused when mutilating *Monkey* into shape, so didn't get as far as I had expected. *Mañana*.

Friday, 28 May 2004

More work on the functions I had created yesterday, and got them working quite nicely—I thought. Then decided that it would be a good test of the system to have detailed list functionality. It turned out I was correct: it was a good test, and the program failed. Spent the rest of the day investigating the block structure, which I suspect I broke during the code changes of the last couple of weeks. In the process worked out a number of useful *gdb* macros, so I can now do things like this:

```
(gdb) pcache
$22 = {
  blockcount = 1024,
  blocksize = 65536,
  alloccount = 19,
  first = 1021,
  Block = 0x8055000,
  stats = {
    reads = 0,
    writes = 0,
    updates = 0,
    flushes = 1,
    hits = 79,
    misses = 18,
    blockin = 0,
    blockout = 0
  }
}
Slot RBN   block  flags
1021   3 807c000 dirty data block, 5 records
1019   5 809d000 dirty data block, 5 records
1020   4 808d000 index block, level 1, 1 records
1022   2 806c000 index block, level 1, 1 records
(etc)
(gdb) block 1019
$15 = (struct Block *) 0x809d000
$16 = {
  h = {
    rbn = 5,
    nrecs = 5,
    ilevel = 0 ' '
  },
  sl = {
    nextrbn = 4294967295,
    prevrbn = 4294967295
  },
  data = " 04"
}
Block at RBN 0x5, level 0, 5 records:
block: 809d000, bptr: 80acffe, offset: 16
Rec 0, offset 10, length 72, absaddr 0x809d010
Contents: -----@-----
$17 = {
  inode_num = 4,
  mode = 493,
  nlink = 1,
  uid = 1004,
  gid = 1000,
  atime = 1085720266,
  atimensec = 0,
  mtime = 0,
  mtimensec = 0,
  ctime = 1085720266,
  ctimensec = 0,
  flags = 0,
  birthtime = 0,
  birthtimensec = 0,
  eof = 0,
  tlib = 0,
  indexlevels = 0
}
Rec 1, offset 58, length 72, absaddr 0x809d058
(etc)
```

That shows a nice, clean inode; unfortunately, what the list function gets to see is complete junk. Monday.

Saturday, 29 May 2004

In the afternoon turned my attention to a temperature logger kit that I had bought from Ozitronics kits (<http://ozitronics.com/>), and which I had seen at Linux.conf.au in January. In the process, realized that I had forgotten a lot of common knowledge about electronics components, and that the kit instructions didn't help. Which is the positive pole of an electrolytic capacitor? Which way round does a diode go? Spent some time confirming that my suspicions were correct (the capacitors I have have a marking next to the negative lead, which is also shorter; diodes have a bar at the cathode end, the one that is shown as a bar in the circuit symbol). Didn't take long to put the kit together, but connecting up the temperature sensors, which look like small transistors, is terrible. There just don't seem to be any components that you can use for this sort of thing. I need to find a better solution to this issue, but for the time being just kludged it by soldering things together, along with some lengths of (German) telephone wire, which has the most confusing colour coding I've ever seen: all conductors red, most with varied-spacing blue stripes. The spacing is such that you can't tell them apart without stripping at least 10 cm of the outside insulation, so I ended up using a continuity meter to find the ends. The result works, but looks terrible.

More investigation is required before I can really use these things, but at least I'm getting an output like this:

```
$ cu -s 2400 -l /dev/cuaa0
Connected.
R V1.0 2002-01-06 20:37:37 C
1 0024.25
3 0023.50
4 0024.68
1 0024.25
```

The first number on each line is the sensor number (note from the photo that 2 isn't connected), and the second is the temperature in °C.

Sunday, 30 May 2004

Spent most of the afternoon writing a program to control fermentation temperatures. ending up with over 1600 lines of code which could actually do something, though of course not control temperatures (yet). Still, things are looking a lot more obvious now.

Monday, 31 May 2004

What a waste of a day! After getting through my mail and documentation in the morning, got a phone call from Yvonne on a land line, because she couldn't get through to me on the mobile. There was a good reason for that: I couldn't find it, and in fact I couldn't recall seeing it after returning from yesterday's ride. Searched the house, but couldn't find it, so started preparing for a new SIM card. That's always more difficult than expected, of course: the phone is still registered in the name of Linuxcare, who do not exist any more in Australia. I've tried to transfer it into my name, but for that I require the agreement of Linuxcare: catch 22. Spent some time on the phone trying to negotiate things, but in the end I was left with the recommendation just to forget the account and get a new number. I can't terminate the account, of course, so that would leave Telstra out of pocket. What a ridiculous setup.

In the afternoon, off to town for a meeting of the IT Council Open Source Committee. We came to the conclusion that the best thing we could do would be to recommend that people consider Open Source when shopping for software, but that our real issue should be Open Standards. With any luck we'll get a new policy on Open Standards in the not too distant future.

Tuesday, 1 June 2004

Back to work on my program testing today, and noted how difficult it is to get back into the swing of things after a few day's away. Fixed the problems I had had on Friday, so now the commands are functional. That rather caught me by surprise, so now I have to think through what to do next.

In the evening playing around more with my temperature control equipment, and got it mainly to work. Using parallel ports under FreeBSD is less than obvious: `/dev/lpt0` is really only for parallel connected line printers, and other equipment, including my relay board, should use `/dev/ppi0`. The interface is less than obvious: all I/O is performed by `ioctl` calls. Still, it was relatively trivial to get it to work, and now things seem to be more or less complete. Once I sort out the mechanical issues, I should be ready to control temperatures.

Wednesday, 2 June 2004

Up earlier than intended this morning at 5:45 am: we had a long-lasting power failure. The important systems have UPSs which can handle short power failures, which are never more than 30 seconds, but if the power fails for longer than 30 sec-

onds, it's likely to be out for 2 hours while ETSA (<http://www.etsautilities.com.au/>) send out somebody to look for the problem and fix it. This failure carried on for a minute, so I had to find my way out in the dark (proving the value of putting torches in well-defined places) and fired up the generator. Even so, lost *battunga*, which was on a UPS. I wonder what happened there. I don't think it can be the duration of the outage.

The systems in the Mike Smith Memorial Room [*my laboratory; see the web site for an explanation*] were a different matter: the battery of the ancient APC UPS I had in there had failed long ago, and it doesn't like the dirty generator power anyway, so I lost all machines. Time for a new UPS.

In mid-morning, Dan Shearer showed up with a satellite dish mount that he had brought back from Canberra for me after the Security Symposium (*diary-feb2004.html#19*). While he was there, he took a look at *filth*, the Microsoft-based laptop that I am using for scanning, to find out why it didn't automatically mount Samba shares, and got that to work. He also managed to navigate the minefield of Microsoft printer setup and configure the colour laser printer as a remote printer. Quite impressive. For some reason, he thought I wouldn't like it to be known that I have used Microsoft, and promised not to tell anybody about it.

More work on my project, but I hadn't had enough sleep, and I was really too tired to think my way through things. As a result also missed a reception associated with the IT Council of South Australia. What a nuisance.

Thursday, 3 June 2004

Continuing work on my program today, and discovered a number of omissions and bugs that kept me going all day. They're all more indicative of haste than conceptual problems; maybe I should slow down a bit. Certainly the pace I've been keeping has tired me significantly.

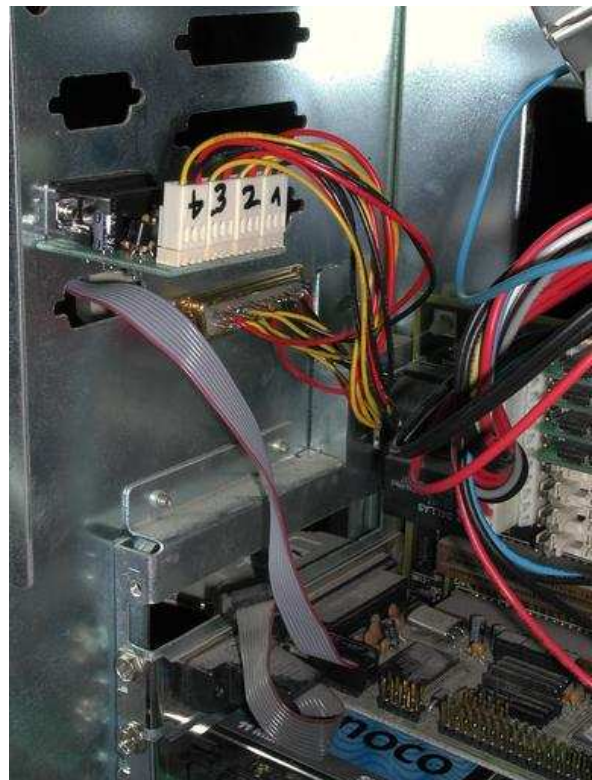
In the evening, more work on the fermentation control, and came to the blindingly obvious solution to my connector problems: what I needed were the kind of connectors that are used on just about every PC to connect the front panel to the motherboard.

I had plenty of them, so spent some time putting things together. The rest worked nicely with a serial cable with 25 pin connectors at each end, which I was able to connect relatively cleanly. It's sad that I have to resort to this sort of solution rather than to get standard solutions.

Friday, 4 June 2004

In the evening, found the energy to do some work on the fermentation temperature controller, and mounted things in the housing of an old computer I have, a 486 DX/2, 66 MHz, 16 MB RAM, running an equally old version of FreeBSD, one of my test kernels at the very beginning of release 5 of FreeBSD:

```
$ uname -a
FreeBSD brewer.lemis.com 5.0-CURRENT FreeBSD \
5.0-CURRENT #1: Tue Dec 12 18:45:30 CST 2000 \
grog@monorchid.lemis.com:/src/FreeBSD/ \
5.0-CURRENT/src/sys/compile/MONORCHID i386
```



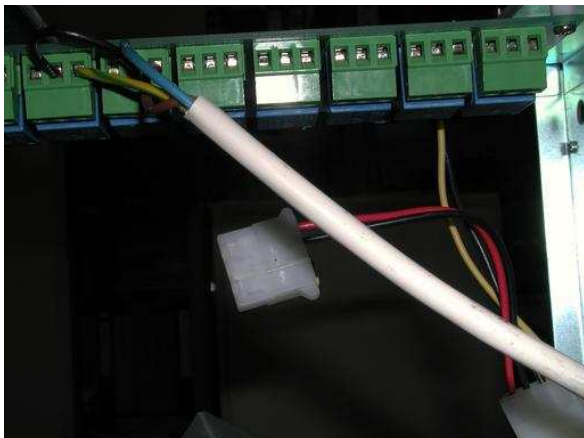
It's connected to the rest of the world by wireless, which I'm beginning to appreciate more and more for this kind of connection. It's now cheaper to install a wireless card than to run cabling.

The results are more functional than pretty. The first one shows the temperature probe assembly. There are no mounting holes on the probe board, so I had to mount it by its 9 pin serial connector. I had already connected to probe cables to a 25 pin connector. I wanted it inside the case, so I had to connect the flat cable to the serial port on the outside of the case (the grey cable going out through another cutout just below the probe board). I need to find some kind of plate that I can use to mount it inside the case.



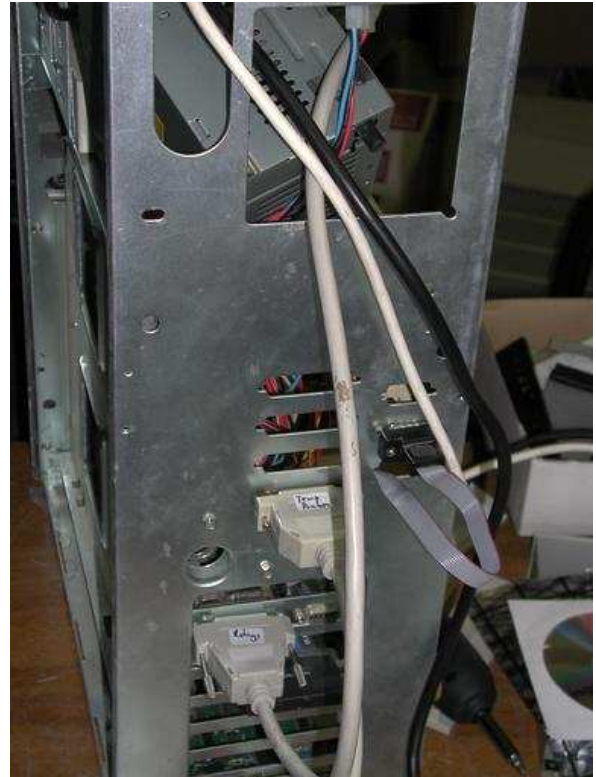
The image above shows the 12V connection to the relay board. I mounted it from the top of the cabinet, and the 12V input is from the computer power supply.

The following one shows the other side of the relay board with the mains power connections:



The next photo shows a view of the back of the computer. This shows a number of things:

- The lower cable goes from the parallel port back inside to the relays. It would be nice to have internal cabling, but I don't know of any parallel ports that connect to a header on the board. They're all connected directly to an external connector.
- Above that is the temperature probe cable, as shown before.
- Higher and to the right, the flat band cable mentioned previously.



- At the top are the relay power outputs (white) and the computer power cable (black). The power supply is in at an angle because it was originally designed for a smaller case, and the internal cables are too short to allow normal mounting. Some time I must buy a proper power supply.

Saturday, 5 June 2004

Somehow today was all spent with temperature control. Started by installing my new computer-controlled fermentation temperature controller in the laundry (next photo). Note the position of the temperature sensors:

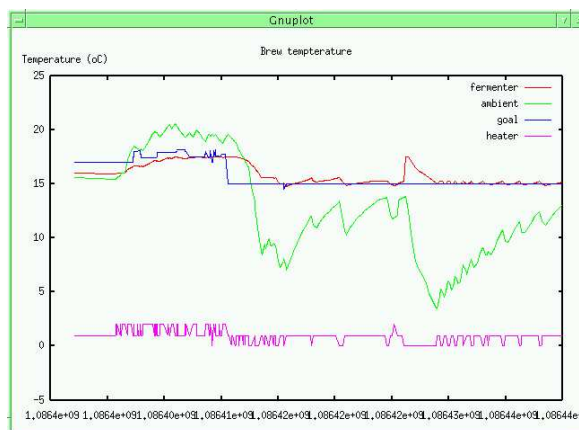
- The external ("room") sensor is on the side of the fridge, not an ideal place, especially when I replace the power supply, when it'll be in the exhaust area. To be relocated.
- The internal ("ambient") temperature sensor is barely visible in front of the 25 pin connector. It's fastened to one of the bars of the grille.
- The wort temperature sensor is taped to the outside of the fermenter. It's covered with some bubble foil to minimize the effects of the ambient air.



- The fourth sensor is intended for a second fermenter. It's hanging down in front of the light.



Spent a lot of time fine-tuning the software, which still isn't ready. I'll make it available on the web when it is, but at the moment it's not in any good condition to publish. Still, it works well. By the evening I was working on graph plotting software—how I hate *gnuplot*—and, with the exception of the ugly plotting, came up with a most gratifying graph:



It looks as if I'll be able to get better than 0.2° accuracy either way.

Monday, 7 June 2004

Back to work again today, and made some progress on my program, though not as much as I want. Writing programs is something like making ice cream by hand: when you start stirring, it's easy, and you make good progress, but as things progress, it gets stiffer and stiffer, and it becomes really difficult to stir. I'm at about that stage now: every detail I need to change seems to require going back and changing lots of details. Still, I've broken the back of it, I hope, and things might get better from now on.

Tuesday, 8 June 2004

My fermentation temperature control stuff has been working really nicely, and it has also generated a fair amount of interest, to judge by the web site hits. I still have some problems with overshoot, but I also have some ideas about how to fix it.

Thursday, 10 June 2004

Spent most of today investigating the indexing status quo that we had been talking about on Tuesday, and made some progress, though after some months still don't really understand the issues. Based on what I find today, though, it looks as if I'm not the only one.

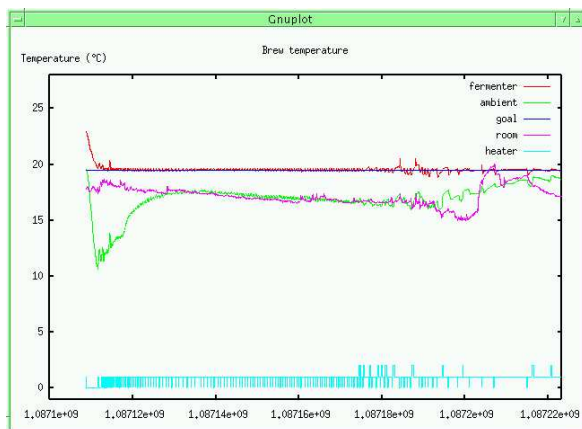
My responses to Computerworld (<http://www.computerworld.com.au/index.php/id;51084128;fp;16;fpid;0>) have generated some interest. Almost immediately I heard from Detlef Borchers of C't (<http://www.heise.de/ct/>), and today I got mentioned on a Slashdot article (<http://slashdot.org/articles/04/06/09/1046206.shtml?tid=102&tid=187&tid=88>). That always has an effect on the web site hits, and today my SCO pages experienced a hundredfold increase in hits. People

seemed particularly amused by my suggestion of sending 20 tonnes of paper to SCO.

Monday, 14 June 2004

Today looked at the proposed amendments to the constitution of the IT council of South Australia, most of which was taken up trying to use Microsoft “Outlook Express” to read the messages and Microsoft “Word” to print them out. What a pain this software is! It’s not so much that it’s bad as that it’s built on broken concepts, that people are too stupid to learn to think.

Spent a lot of time looking at the beer temperature control, though, and made many improvements, some of which seem to have made things worse. On the other hand, I was testing on a very vigorous fermentation, and that won’t have made things easier. Still didn’t find time to get *gnuplot* to interpret times correctly, so what we’re left with is:



From about 60% into the time, the temperature control gets pretty ragged. Admittedly, it’s not as simple as it seems: the first temperature spike was when I poured in 3 litres (1/8 of the total volume) of water at about 24°, and the second was when I taped the temperature sensor to the fermenter more carefully. I may need to look more carefully at how I measure the surface temperature.

Tuesday, 15 June 2004

Into Adelaide for an extraordinary meeting of the IT council of South Australia to discuss the proposed changes to the constitution. Looks like we’re going to split into an executive board and a less active council with many more members. In many ways, the meeting was like the AUUG meetings on similar topics, and after two hours we were still far from agreed. Still, hopefully Graham has enough to go on to bring out a draft for vote in 3 weeks time.

Monday, 21 June 2004

Back to work on my program today. for some reason, I end up with a `SIGSEGV` in `memmove`, a likely enough place—except that the parameters passed to it look fine. Spent some time investigating that one; it could be hiding a very interesting and obscure bug.

Tuesday, 22 June 2004

On with my work today, and made further progress. My obscure bug turned out to be difficult to find but easy to fix: the Monkey record offsets are 16 bits, and in the past there’s been no problem. Now I have increased the block size to 64 kB, and the offsets can grow into the sign bit, causing some interesting data corruption as the data lands 64 kB from where it should be. Changing everything to unsigned solved that problem. The issue was that the problem showed up in `memmove`, which has no stack frame, and so my macros were showing the wrong values. Ended up having to dump the top of stack directly on the call instruction, and that showed a negative count being passed.

Wednesday, 23 June 2004

In the evening to town to an information evening by Solution City (<http://www.solutioncity.com.au/SolutionCity/>), including the SA Minister for IT and the Mayor of Adelaide. Heard an interesting talk about the Adelaide delegation (the only delegation from all of Australia) at the World Conference on IT in Athens, mainly the same photos as on the web link. It’s interesting to see what the others are doing, but it’s rather a long way from my particular areas of activity.

Thursday, 24 June 2004

On with my program today, and made some progress, at the same time noting that *gdb* is really bad at some things. I haven’t looked at the new version 6, though I probably should do, but the previous versions have such obvious problems as being unable to directly specify a line in file (you need to find a function in the file first and list that, making that the default file; then you can specify a line number). One of the things that I found particularly annoying was the lack of a “canonical” raw memory dump function such as the kind that *hexdump -C* does:

This output is wrapped to fit the format. The original consists of three lines. See the web site for more details.


```

00000000 2f 2a 20 24 49 64 3a 20
          73 75 67 61 72 2e 63 2c
          /* $Id: sugar.c,|
00000010 76 20 31 2e 32 20 32 30
          30 33 2f 31 31 2f 31 37
          |v 1.2 2003/11/17|
00000020 20 30 30 3a 34 38 3a 34
          73 75 67 61 72 2e 63 2c
          /* $Id: sugar.c,|

```

Set to writing one, and was surprisingly successful:

```

# Dump memory in "canonical" form.
# dm offset length
# This version starts lines at addr & ~0xf
define dm
set $offset = (int) $arg0
set $len = (int) $arg1
while $len > 0
# Print a line
printf "%08x: ", $offset
# byte address of start of line
set $byte = (char *) ($offset & ~0xf)
# first byte number to display
set $sbyte = $offset & 0xf
set $ebyte = $sbyte + $len
if $ebyte > 16
set $ebyte = 16
end
# And number of bytes to print on this line
set $pos = 0
while $pos < 16
if $pos < $sbyte || $pos >= $ebyte
# just leave space
printf " "
else
printf "%02x", *((char *) $byte) & 0xff
end
if $pos == 7
printf " "
end
set $pos = $pos + 1
set $byte = $byte + 1
end
printf " "
# Now start again with the character
# representation
# Start byte number on line
set $pos = 0
# byte address of start of line
set $byte = (char *) ($offset & ~0xf)
while $pos < 16
if $pos < $sbyte || $pos >= $ebyte
# just leave space
printf " "
else
if ((*$byte & 0x7f) < 0x20)
printf "~"
else
printf "%c", *$byte
end
set $byte = $byte + 1
end
set $pos = $pos + 1
end
printf "\n"
set $len = $len - 16 + ($offset & 0xf)
set $offset = ($offset + 16) & ~0xf
end
end

```

This enabled me to write other macros that call the macro for a raw dump:

```

Rec 0, offset 10, length 32,
      absaddr 0x822f010
0822f010: 4d f6 dd b7 eb b3 76 c9
          45 6e 57 d8 55 d8 8c f9
          MčÝ·ë³vÉEnWØUØ~ù
0822f020: 00 00 0e 00 00 00 00 00
          b4 00 15 28 01 00 00 00

```

Somehow I find programming in *gdb* macro language particularly frustrating, perhaps because the language is (gratuitously) different enough from C to not be obvious, and there appears to be no documentation. There's also some kind of implicit typing in the parameters, with the result that I could make changes, test them and confirm that they work, but after restarting *gdb* I'd get messages like:

```

Rec 0, offset 10, length 14, absaddr
      0x81bf010
081bf010: 00 00 00 00 00 00 00 00
          0e 00 00 00 59 06
Invalid type combination in ordering
comparison.
(gdb) dm 0x81bf010 14
081bf010: 00 00 00 00 00 00 00 00
          0e 00 00 00 59 06
          -----Y-

```

More work to be done. I'm pretty close to being able to read files, though.

Friday, 25 June 2004

Finally bought a new UPS for my laboratory machines. The number of power failures recently has been completely unacceptable; we must have had 5 or 6 this month.

Saturday, 26 June 2004

Rather to confirm the correctness of my purchase of another UPS, we had a further power failure this morning, which at least gave me a chance to install the new UPS. I don't like taking machines down when they've been up for a long time, but after a power failure I don't mind. Still, we're a long way from the uptimes I had at the beginning of the year; the longest uptime is *flame*, with only 180 days.

Con Zymaris has resigned as editor of AUUGN (<http://www.auug.org.au/auugn/>), and we haven't found a replacement for him. Why do I always end up holding the baby? In any case, given the work that it caused Con (admittedly using OpenOffice), I had been putting it off, at least partially with the excuse that people due to hand in their copy hadn't done so yet. Today I finally had to admit that the end of June is approaching, and got down to work, converting the whole thing to *groff*—not necessarily a recipe for getting things done quickly. To my surprise, by the end of the day I had about 30 relatively well formatted pages. There's still a lot to do, but most of it involves getting more copy from people. Makes me wonder what the fuss was with OpenOffice.

Monday, 28 June 2004

Somehow hardware was the flavour of the day. In the afternoon, received in the mail 9 36 GB SCSI disks donated by Ade Lovett. They're ex his servers, and they've given him some minor problems in the past, but they're almost certainly good enough for things like performance measurement. Then discovered I had received an almost new (2 month old) Dell Inspiron 9100 laptop—that was the good news—and that the display had failed. Spent some time on the phone with Dell, who were remarkably efficient, and on IRC discovered that the display failure is apparently a known problem with this model.

To round things off, in the evening *echunga* died again. I'd already had a problem of this nature at the end of February (*diary-feb2004.html#28*), but this time it happened while I was there: the screens went dead, and the audio CD in the CD-ROM drive stopped dead, something that almost never happens (once you start it playing, it carries on without further activity on the part of the CPU). Spent some time trying to revive it, without success, and ended up putting the disks into *zaphod*. Had the machine up and running in about 70 minutes, but then took some time longer to configure X (*echunga* is the new machine with three displays described in my hardware configuration page (*hardware.html*)). Looks like *another* dead motherboard, and this one was only nine months old.

Tuesday, 29 June 2004

Up this morning to find that *echunga* had spent hours doing level 2 dump. At first I thought it was because of the slower processors (*bzip* really takes a lot of processor power), but then I noted:

```
dump -2uf - / | bzip2 > \
/dumpa/echunga/2/root.bz2
DUMP: Date of this level 2 dump: Mon Jun 28 \
21:00:00 2004
DUMP: Date of last level 0 dump: the epoch
```

Fortunately (in this case), I hadn't *fsckd* my main source disk, */src*, which would have dumped 80 GB and completely overflowed the dump disk. Looking at */etc/dumpdates*, I saw:

```
/dev/ad0s1a 0 Tue Jun 1 21:00:02 2004
/dev/ad0e 0 Tue Jun 1 22:37:14 2004
/dev/ad0s1a 2 Sun Jun 27 21:00:00 2004
/dev/ad0e 2 Sun Jun 27 21:04:24 2004
```

There was the problem: *dump* goes by file systems, not by mount points. The ASUS BP6 motherboard that *zaphod* used has four IDE controllers, and the names of the disks had changed to */dev/ad4s1a* etc. I had accounted for that in

/etc/fstab, but I hadn't thought of it in */etc/dumpdates*.

Next, tried to *fsck /src*. The system froze during phase 2 and wouldn't even let me into the debugger. Rebooted, repeated, froze again during phase 2. And a third time, after which I decided to try doing it on a different machine. *beeble* is about all that's left, and that turned out to have file system corruption on the FreeBSD 4.10 side. Tried again with FreeBSD 5.2, and had to boot several times before the disk was recognized. When I did, decided that it was flaky enough to try a partial backup of those directories I knew that I had changed since the last backup (on Saturday); the second time round, that worked. Then set to doing an *fsck*, which, surprisingly, also worked. I was also able to read the entire disk:

```
# dd if=/dev/ad6c of=/dev/null bs=128k
```

That took almost exactly 40 minutes, or 33 MB/s, and reported no errors, so I felt relatively safe.

Still, I didn't trust the disk, and spent some time looking at the SCSI disks that I had received yesterday. The results were discouraging: I couldn't get any of them to come ready. They're IBM DDYS (36.7 "GB") and IC35L036 (36 "GB") LVD drives. They were all jumpered for auto spin, and the host adaptor (a Symbios 875 LVD adaptor) recognized them without any trouble. It just couldn't "spin them up". Tried also with an old DPT host adaptor and an Adaptec 2740 UW, and also tried forcing SE (another jumper on the drives), all without success. At least the Adaptec reported the problem better. Symbios said "Cannot verify disk". Adaptec reported:

```
CDB: 03 00 00 00 0E 00 70 00 02 00
Status: 0 - no host adaptor
Target status: 2: Check condition
Sense key: 2 - not ready
Sense code: 0
Qualifier: 0 (2 disks), 85 (another 2 disks)
```

This shows that the CDB (command descriptor block) was a request sense command (first byte 3), and that it was expecting up to 14 bytes back (0E, rather less than the drives can probably return). It seems strange that this should happen across the board, but I can't see anything I'm doing wrong.

Wednesday, 30 June 2004

And another day spent in hardware hell. During the morning it became clear that my */src* disk drive was seriously sick, and sent Yvonne into town to pick up some new hardware. In the

meantime tried to improve my system upgrade procedure, with only moderate success; did manage to install FreeBSD 5.2.1 on a disk destined to be the new *wantadilla*, and built a -CURRENT world on it.

Yvonne back with a new computer case and a MSI K7N2 Delta motherboard, the same kind as in *wantadilla*. I had bought it because I knew it worked in *wantadilla*. First, though, I connected up the old motherboard to the new power supply. I couldn't even power on. By contrast, the power supply in the old case would run even without the motherboard, so it looks as if the motherboard is well and truly dead. Hopefully it hasn't taken the processor and memory with it. I can't test the latter, since the new motherboards also take new memory. I didn't want to test the former, since it's such a pain to replace processors nowadays.

Installing the new motherboard wasn't as simple as I thought. It proved to be slightly different from the other motherboard I had; in particular, the BIOS allowed setting the process FSB speed correctly: I had had some trouble with that when installing the last one. Decided that it would make more sense to build the kernel on the new box (2500XP+ with 1 GB memory) than on *zaphod* (in this incarnation dual 466 MHz Celerons with 128 MB memory). Only: I couldn't boot. It got as far as trying to start *init* and then hung.

After some messing around discovered that the problem didn't exist with Linux, but it existed with all versions of FreeBSD I tried (mainly -CURRENT, 5.2.1-RELEASE and 4.10-RELEASE). The first one (really the system disk from *beeble*, which currently doesn't have a system to run in) had a kernel debugger, and it showed that the processes were ready to run, but that the USB subsystem was interrupting at about 200,000 times a second. Reset, disabled USB, and was able to boot. Then I had trouble with the network interfaces, both a 3Com 3C905C and a Realtek 8139. In each case, they, too, interrupted continuously, depending on the system between 78,000 and 80,000 times a second. Disabling the APIC solved that problem.

In the meantime spent some time trying to recover the */src* file system onto the new disk I had bought, which involved a number of hangs. This old disk is well and truly past it. I only bought them less than two years ago (*diary-jul2002.html#23*).

SEQUENT: Asynchronous Distributed Data Exchange Framework

Devraj Mukherjee <devraj@eternitytechnologies.com>

This paper was originally presented at the AUUG 2003 Conference "Open Standards, Open Source, Open Computing" and is reprinted with the permission of the author. See the end of the article for important copyright information.

The SEQUENT idea

The project code named SEQUENT intends to produce a software framework that would enable applications exchanging data of a similar type, to query and exchange data based on a trust list per user over a computer network asynchronously, regardless of the device, platform, and/or programming language.

The framework intends to use an XML based protocol for the data exchange and integrate it with popular authentication mechanisms that are commonly used in the industry.

The framework in this document is hereby referred to by its code name as SEQUENT.

Origin of the idea

While designing a look alike of the Palm Desktop environment, which is available for Windows and Macintosh from PalmSource Inc. for Linux, it was identified that the only two practical ways, two handheld users could exchange data was either by participating on an enterprise network (such as Microsoft Exchange server, or Palm Enterprise Server) or be physically present to transfer the data using a physical, infrared, or bluetooth technology.

Unfortunately all of us don't have the resources to participate on an enterprise network, especially from our personal workstations. Most of us also comprehend the need of not having to meet physically to exchange the data. To explore the idea further at most instances, we would like the data exchange to be automatic and not even bother us; this given that we are able to verify that the person is who he says he is.

In today's technically savvy world, desktop computers are probably the best connected, considering the cost per time for connectivity. Therefore it

would make perfect sense, to allow the desktop software, which would manage the information downloaded from the handheld to participate in an open network, where users could exchange data based on credentials defined for the other users on the same network.

A data exchange solution was thus conceived (Code-named: SEQUENT) which would allow any two applications exchanging data of the same kind, regardless of the device, platform and programming language to communicate and exchange data in a simple and secure fashion. It is also defined that, the data be exchanged via a middle body (a server), which can authenticate users as well as allow asynchronous data exchange to facilitate different users who fulfill queries at their own convenience.

As the idea matured, it became obvious that the framework could provide a connectivity platform for many other applications and was thus considered for conception as a framework, rather than a feature in the original application.

The idea expressed

SEQUENT is not about objects or just about exchanging data between two applications connected to a network. The framework aims to provide a structured manner for querying data, over distributed data stores across different platforms and different data formats as long as the interest of the applications are similar.

For example, if Company A manufactures a calendaring product (Let's call it Product A) on a UNIX based platform and chooses to use the Berkeley styled DB format to store the information. Company B however wishes to manufacture a calendaring product (Let's call it Product B) for the Windows platform and chooses to use a proprietary data format for storing data. These two products still deal with the same concept: calendaring. SEQUENT would then act as the bridge between the two applications where using the client API, it would allow Product A to encode the query and allow Product B to interpret the query. Thus enabling communication between the two products with the same aim but totally different development philosophies.

The need for a common and open standard

Collaboration is an important feature in modern software applications and it is important to be able to collaborate regardless of platform, and the manufacturer of the software product. It is the

kind of application (or the service provided by the application) that should tie the applications together.

The reason for success of products like Yahoo Messenger would be, its capability to communicate over computer networks, regardless of the platform and the implementation. One can choose to use the official client if they were on the Windows or the Macintosh platform or a product like Gaim (Gnome AOL instant messenger).

If an open source standard is established and implemented as a part of this project, it will enable applications attempting to exchange similar kind of information to perform this operation over a computer network, without any modification to the design of the product itself.

Conventions used in the document

Implementation specific details of the framework are out of the scope of this article. This article merely outlines in detail the idea of the distributed query and data exchange framework. Prototyping of this framework is likely to be done using the Java programming language.

1.4.1 Requesting application - refers to an application that uses the SEQUENT client side API and places requests to query or exchange data.

1.4.2 Data exchange service - refers to a SEQUENT relay server, which acts as a central point and keeps track of the the access control lists and provides services like authentication and asynchronous exchange of queries and data.

Background research

Existence of similar solutions

The problem obviously is not new; there are applications that do similar things in various capacities. The most obvious examples would be collaborative tools such as Yahoo Messenger, ICQ and the likes. These applications enable users to leave offline messages making it obvious that they have an asynchronous data exchange network in place. Access control lists, referred to as "Buddy Lists" are also saved on the server that allow the users to use any computer around the world and still have their preference of access stored.

XML has received enormous attention and is now called the "ASCII of the web" as it allows such transparent exchange of structured data over the web infrastructure. Most programming languages

have support for XML parsing. XML is definitely a powerful solution to expressing data, because the data expressed using XML has rules and every application irrespective of the programming language they are written in or the platform they run on can validate the information.

Background reading of various technologies and/or implementations suggests that many attempts towards such implementations have been made and that the technologies required to make such a data exchange network already exist.

Protocol design considerations

Described in XML

The protocol used by SEQUENT has been described in XML. XML is a widely accepted standard and most programming languages can interpret the syntax with great ease. The acceptability of XML and the availability of parsing APIs should encourage developers to make implementations for of SEQUENT servers and client in different languages.

Apart from being a W3C standard, XML has many technical advantages. XML data can be verified for integrity, which is an important factor if you are transferring data across from one information base to the other. As the data is well described it is easier to translate and encode both on the client and server end. XML data is described using plain text allowing messages in multiple natural languages to be encoded between tags and MIME allows encrypting data such as binary files, images to become a part of the XML message.

Application centric

The "Application" is the most important design issue of SEQUENT. The protocol exchanges data based on the kind of application that is requesting information, or more specifically the kind of data that needs to be exchanged. The platform, language, device or transport layer should be transparent to the application attempting to exchange data.

MIME encoding and XML

MIME allows encoding of information in Internet messages. Many XML protocols use MIME to encode binary data in the messages, this method is very well researched and documented and used fairly widely as a standard way of transporting information over the Internet.

The security (encryption) issues are addressed in the security section where it is described how SE-

QUENT uses a digital certificate based model to secure the communication channel.

Framework design

The design of the framework is divided into two major parts, the server or the data exchange service and the client side API (application programmers interface).

The data exchange service is responsible for holding access control lists; verify the authenticity of a user and make the exchange of data or the request.

The client side API provides a transparent framework that allows applications to easily integrate features provided by the SEQUENT network.

The theory of data exchange

SEQUENT exchanges information using a distributed network, by encoding the data using a XML data protocol. The data exchange framework addresses all possible exchange issues; including security, authentication, id stamping and the data exchange process itself.

Establishing a secure communication channel

The primary step of communication between two computers is establishing a secure communication channel between the two parties. Servers usually wait for the client to initiate the connection. SEQUENT requires the client to establish a secure communication channel with the server before it starts sending information over the port. This is to ensure that the data encrypted using XML is securely transported over this channel, as most of the information is very sensitive, at least to the person who is releasing the information.

SEQUENT chooses to use a digital envelope model (RSA Data securities, 2003) for securing a transport channel between the server and the client. When a client initially contacts the server to communicate, the server checks to see if it posses a secret key for the client.

If the server does not posses a secret key for the requesting client, the server makes it's public key available to the client. The client then generates a secret key using a previously agreed (agreed by the developers of the framework) encryption algorithm for its session with the server. It then encrypts a message containing the generated secret key and information about the client itself with the server's public key using an asymmetric encryption algorithm and sends it back to the server. As this message can only be decrypted by the

server, it enables securely transferring the secret key over to the server.

Once the secret key is obtained, the client and the server can securely communicate by encrypting messages using a symmetric encryption algorithm. The server chooses to expire the secret key generated by a client using a time limit.

The protocol can thus be summarized with the following sequence of pseudo messages. Scenario 1: The server does not posses the secret key for a client:

```
Client -> Server: Hello, I am Client A
Server -> Client: Hi, I don't have your
                  secret key. Here is
                  my {public key}, why
                  don't you send me your
                  secret key
Client -> Server: Here it is {secret-key}
Server -> Client: Thanks, I got it. Bye.
```

Scenario 2: The server posses the secret key for a client

```
Client -> Server: Hello, I am Client A
Server -> Client: Hi Client A, I already
                  have a secret key for
                  you and you can use
                  that to communicate
                  with me. Bye.
```

This mechanism only manages to secure the communication channel between the server and the client, issues related to authentication and user trust lists is addressed later in this document.

Authentication

The framework looks forward to integrating with industry standard networking authentication models, such as Kerberos, NT challenge response, LDAP based authentication models.

For more simplistic models the framework is built in with a simple username and password based authentication model. The user access list is maintained by the SEQUENT server and is basically a list of encrypted usernames and passwords.

There is a fair chance that SEQUENT network users on a particular server might not necessarily be users of the server system or be a part of any other authentication list that the server is linked to.

The client chooses to request an authentication mode and the server responds to it if it has the capability to. If an authentication module is not implemented on the server then the client may choose to try another method of authentication method. If all else fails the default Username and

Password based authentication will always be available for user verification.

A sample authentication request using a username and password would look like an XML based request such as

```
<?xml version="1.0"
      encoding="ISO-8859-1"?>
<SEQUENT:Authenticate method="Standard">
  <Username>Administrator</Username>
  <Password>TheMagicWord</Password>
</SEQUENT:Authenticate>
```

User trust lists

For all examples quoted in this section we will consider "User A" to be the requesting party and "User B" and "User C" to be the receiving parties of the query.

In a SEQUENT enabled network, every user maintains a trust list. A trust list for "User A" can be defined as a list of access permissions for other users on the a SEQUENT server to the information base (data on "User A's" local computer/device) maintained by "User A".

A SEQUENT server uses these trust lists to evaluate if a request sent by a user should be forwarded to the other users. If the user initiating the request does not have sufficient credentials to query another users information base, then the server drops the requests there itself.

A sample trust list expressed in pseudo language would look something like this:

```
Trust list for User A:

  User B: Application A: Full Access
  User C: Application A: Query, Write
  User B: Application B: Query,
                WriteWithDiscretion
```

SEQUENT identifies that applications really need to do three things, Query information, Submit information for addition or request deletion of an entry and thus defines three kinds of permissions Query, Write, Full Access (allow Query, Write, and Delete). It is to be kept in mind that all applications referred deal with similar kind of data.

Permissions can be classified into two types, one a permission that does not require any discretion from the user owning the information base that satisfies the request and the other where user discretion is required for the action to be taken. We can thus summarize permission to the following

```
Query
Write
Full Access
QueryWithDiscretion
```

```
WriteWithDiscretion
FullAccessWithDiscretion
```

If "User A" does not have permissions to access "User B's" database at all then the query initiated by "User A" is never forwarded to "User B", saving network traffic and time on "User B's" workstation or device.

All permissions that require discretion of the responding party will wait for the user to authorize the response if the information base residing on that workstation or device can fulfill the query. These permissions are very useful if the users want to provide access to delete or write data in their information bases, especially when dealing with data such as "Address Books".

ID stamping queries on the server

Each workstation trusts the server blindly. Information received from the server is accepted to be valid. By doing so the client becomes independent of many tasks, thus making the client side implementation simpler. The client also does not receive any request that the client would reject on receipt due to permissions defined in the access control list. To the client unless it really wanted to know which user initiated the request, the request was initiated by the server and the response is to be sent back to the server. It is then to the server to figure out who initially initiated the request and to sent it back to the initiator.

When a request is initiated, the server knows who initiated the request because the user is logged in. The server accepts the requests and makes a note as to who initiated the request, stamps the request with a Request ID so that it can refer back to the original information if and when it receives any response to the query. It then recreates the query with the vital information required by all the recipients inclusive of the generated Request ID and forwards it to the valid recipients.

The recipients of the request download the queries and if possible respond to it. The response to every query quotes the Request ID so that when the server receives the response it knows whom it has to be sent back to. The original requests to contain the username of the user who initiated the request incase the recipient of the query wishes to gather more information about the user.

Initial query sent by a user

```
<SEQUENT:Request nature="Query"
    RequestFrom="AddressBook"
    RequestTo="AddressBook"
    Initiator="samar">

<Request:FieldList>
    <FieldName>First Name</FieldName>
    <FieldName>Last Name</FieldName>
</Request:FieldList>

<Query BasedOn="PhoneNumber"
    Match="Exact">02-69258392</Query>

<Query BasedOn="Name"
    Match="Close">Devraj</Query>

</SEQUENT:Request>
```

The request as seen by the recipients after ID stamping on the server

```
<SEQUENT:Request nature="Query"
    RequestFrom="AddressBook"
    RequestTo="AddressBook"
    RequestID="001">

<InitiatedBy>samar</InitiatedBy>
<Access>QueryWithDiscretion</Access>

<Request:FieldList>
    <FieldName>First Name</FieldName>
    <FieldName>Last Name</FieldName>
</Request:FieldList>

<Query BasedOn="PhoneNumber"
    Match="Exact">02-69258392</Query>

<Query BasedOn="Name"
    Match="Close">Devraj</Query>

</SEQUENT:Request>
```

The response sent by one of the recipients

```
<SEQUENT:Response
    ResponseFrom="AddressBook"
    ResponseTo="AddressBook"
    RequestID="001">

<Response>
    <First Name>Devraj</First Name>
    <Last Name>Mukherjee</Last Name>
</Response>

</SEQUENT:Response>
```

The response from the server to the initiator

```
<SEQUENT:Response
    ResponseFrom="AddressBook"
    ResponseTo="AddressBook"
    Responder="vinayak">

<Response>
    <First Name>Devraj</First Name>
```

```
<Last Name>Mukherjee</Last Name>
</Response>

</SEQUENT:Response>
```

Sending outgoing requests or responses

Once authenticated with a SEQUENT server the client can send out the pending requests that applications have made while the workstation or device was not connected or more specifically from the last log-on.

Responses to previously received requests are also sent back to the server for redistribution during this process. If a particular information base does not have the response to a request it may choose to ignore the request thus saving network traffic.

Sample Request Sent by a client

```
<SEQUENT:Request nature="Query"
    RequestFrom="AddressBook"
    RequestTo="AddressBook"
    Initiator="samar">

<Request:FieldList>
    <FieldName>First Name</FieldName>
    <FieldName>Last Name</FieldName>
</Request:FieldList>

<Query BasedOn="PhoneNumber"
    Match="Exact">02-69258392</Query>

<Query BasedOn="Name"
    Match="Close">Devraj</Query>

</SEQUENT:Request>
```

Sample Response sent by a client

```
<SEQUENT:Response
    ResponseFrom="AddressBook"
    ResponseTo="AddressBook"
    RequestID="001">

<Response>
    <First Name>Devraj</First Name>
    <Last Name>Mukherjee</Last Name>
</Response>

</SEQUENT:Response>
```

Receiving incoming responses or requests

During every communication session with the server, responses to previously sent requests along with the new requests initiated by other users on the network are also downloaded. Once downloaded the responses are sent to the sub part of the application for appropriate action. The new requests are passed on to the different extensions (discussed in detail later in the document, Client Side API) made using the SEQUENT API, which fulfills the queries initiated by the request.

Requests forwarded by the server

```
<SEQUENT:Request nature="Query"
    RequestFrom="AddressBook"
    RequestTo="AddressBook"
    RequestID="001">

<InitiatedBy>samar</InitiatedBy>
<Access>QueryWithDiscretion</Access>

<Request:FieldList>
    <FieldName>First Name</FieldName>
    <FieldName>Last Name</FieldName>
</Request:FieldList>

<Query BasedOn="PhoneNumber"
    Match="Exact">02-69258392</Query>

<Query BasedOn="Name"
    Match="Close">Devraj</Query>

</SEQUENT:Request>
```

Responses forwarded by the server

```
<SEQUENT:Response
    ResponseFrom="AddressBook"
    ResponseTo="AddressBook"
    Responder="vinayak">

<Response>
    <First Name>Devraj</First Name>
    <Last Name>Mukherjee</Last Name>
</Response>

</SEQUENT:Response>
```

Application Type recognition and Cross application queries

As mentioned before the kind of application dealing with the data is what is important. Application Types must be defined in order to be able to identify which query gets forwarded to which application on the client end. Servers need not be aware of the client application types, as it will depend from client to client. Servers merely act as information routers irrespective what the query is and what the response is going to be.

The client side API (as discussed further in the document) will provide the infrastructure to define extensions of the SEQUENT framework that will act as interaction points for the custom applications. These extensions will run as child threads to the parent SEQUENT client thread, to reply to queries received by a workstation.

If an Application Type is not recognized by a workstation then the application sends a response to the request informing that the kind of application sending the request is not available on the responding client. The server can then choose to ignore telling the requesting client or choose to generate an error message that can then be down-

loaded by the client and ignored or an action can then be taken based on the error response.

The usual scenario is generally similar applications communicating with each other, however an application can choose to address the query to another application and receive information for it's own use. A classic example could be a calendaring application requesting information from an address book application to link it to an event. Cross application communication defiantly required the two applications to be aware of the kind of information that is required to be sent across to make a query complete.

The server engine

The server engine, is primarily running two kinds of services, a data exchange service that allows the exchange of data requests and responses between two connected and authenticated clients and an identification verification service which allows user accounts to be verified and maintenance of a user access list enabling the verification of the right queries being forwarded to the right user.

These two services are a must on every SEQUENT server as it creates the heart of every SEQUENT server.

The client side API

SEQUENT will rely heavily on a standard well-defined client side framework that can integrate easily with existing applications. The client API will implement the SEQUENT protocol in full to enable the implementing applications to communicate with the server without worrying about the data exchange process.

The client API will require the implementing application start a thread already defined by the API, which will attempt to communicate with the server after a fixed interval of time if a connection can be established to perform the data and query exchange.

Every application that is then written, using the SEQUENT framework must implement some pre-defined functions, which will act as the bridge between the main thread to resolve the inbound queries and the implementing application for generating queries.

The SEQUENT client side API will also implement functions that will allow change of preferences

such as password, access control lists and permissions that are stored on the server. It will also feature a default implementation of a GUI based control center, allowing the end user to change the information. The application programmer however may choose to write his or her own control center application that might be more suited to their application style.

The communication thread

The client side API requires the application to start the main thread so that it takes care of the communication and data exchange while the application performs the it's usual tasks in foreground.

This thread handles issues such as authentication once a connection is established, formulating queries based on the information collected from implementing applications and also redistribution of the queries using the application specific extensions once they have been received on the client end.

The main communication thread is invisible to the end user and all the developers using the SEQUENT framework need to know is that they need to start the thread when their application starts.

Application specific extensions

The client API defines a set of rules for custom applications to allow themselves to blend in with the SEQUENT framework. Applications that are SEQUENT enabled exists on their own, this is to enable existing application to become SEQUENT enabled very easily.

The API allows extension (if talked in terms of object oriented programming then super classes) of the SEQUENT system by implementing some mandatory methods that allows each application to integrate into the main client communication thread. These over ridden methods define how an application responds to a query when a request or response needs to be formulated.

Each custom extension that is a part of the custom application but an extension of the framework runs as a child thread to the main communication thread (section 3.4.1) completing the client side implementation of the framework.

Many applications or one application divided into many sub applications can live in the same mem-

ory space and some may choose not to be part of the SEQUENT model, due to nature of the information they publish, while others do.

Future

Inter server communication

The first obvious extension such as a framework would be being able to communicate across servers. The client side of the first design of the framework allows keeping track of a list of server that one user is allowed to access. It also requires users to define which query should be floated on which network.

With the growth of such an idea the next obvious integration is inter server communication, where by a user is able to define permissions across servers. Usernames would then look much like email addresses such as devraj4@eternitytechnologies.com and szutshi@devraj.org

DNS is already in place all across the TCP/IP networks including the Internet. The server would the have to be featured with a capability of asynchronously exchanging information with each other to form this worldwide network of distributed data exchange.

Application development framework

Most applications usually perform the same sort of tasks. The client side extension would then involve production of a client application development framework that allows development of applications to become simpler.

The framework will address issues like workspace management, layout management, and connections to the database as well as the network. The distributed data exchange framework can also be extended to produce a thin client application-authoring framework, which allows defining layouts of user interfaces using an XML styled language that can send messages using the framework for execution of functionality on a different workstation.

The details of such an implementation is out of the scope of this paper and would totally be dependent on the existence and strength of a parent framework like SEQUENT.

References

1. How SSL works? from Netscape—
<http://developer.netscape.com/tech/security/basics/index.html>
2. Key Agreement from RSA data securities—
<http://www.rsasecurity.com/rsalabs/faq/2-2-3.html>
3. RSA data security on Digital Envelopes—
<http://www.rsasecurity.com/rsalabs/faq/2-2-4.html>

Trademarks and Copyrights

All documents that the author makes references to have been credited for. All trademarks and copyrights are owned by the respective copyright and trademark holders.

All software, ideas, and concepts manufactured by Devraj Mukherjee and Eternity Technologies is Copyright of Eternity Technologies and may not be reused without prior written permission of the corporation. All software published as a part of this project will be published under an Eternity Technologies Open Software license, please refer to the software license for further details on distribution and usage of the framework.

Letters to AUUG

This column contains selected messages from the AUUG-talk mailing lists. To sign up for this mailing list, visit the mailman pages at <http://www.auug.org.au/mailman/listinfo/talk>.

From: Greg Black <gjb@auug.org.au>
Date: Wed Sep 8 16:26:22 2004
Subject: [AUUG-Talk]: The future of AUUGN

Well, I've read Greg Lehey's editorial "The new face of AUUGN" in the current issue and it seems worthy of comment. I must say I'm disappointed the decision has been made "to stop production of the paper version of AUUGN by the end of the year."

However, if the cost of printing is so high and if we really could see a significant reduction in membership fees in return for this move, I'm happy to accept the decision.

I do hope that AUUGN will continue — preferably on the web rather than in the form of CDs,

which also cost money to produce and distribute. In fact, I'd think a new form of AUUGN, perhaps built around a blog, might be a good thing. Then we could just subscribe to the RSS feed and be kept up to date.

I do like the inclusion of conference papers in the current issue and would hope that might continue.

Cheers, Greg

From: Greg 'groggy' Lehey <Greg.Lehey@auug.org.au>
Date: Wed, 8 Sep 2004 16:38:54 +0930

On Wednesday, 8 September 2004 at 16:56:14 +1000, Greg Black wrote:

> Well, I've read Greg Lehey's editorial "The
> new face of AUUGN" in the current issue and it
> seems worthy of comment. I must say I'm
> disappointed the decision has been made "to
> stop production of the paper version of AUUGN
> by the end of the year."

> However, if the cost of printing is so high
> and if we really could see a significant
> reduction in membership fees in return for
> this move, I'm happy to accept the decision.

This caused a surprising amount of discussion at the AGM. In general, the sentiment expressed was like yours: "pity, but understandable". I assume that this sentiment is not the only one: a significant number of members didn't read it, so they had no cause to complain about the decision. But I'd like to remind you about the option of having it printed. If people are interested (but only then), we can do some enquiries about the cost of such printing. It's possible that it would be comparable to the costs of producing AUUGN.

> I do hope that AUUGN will continue --
> preferably on the web rather than in the form
> of CDs, which also cost money to produce and
> distribute.

We're currently thinking of both.

> I do like the inclusion of conference papers
> in the current issue and would hope that might
> continue.

Thanks. That's an easy source of material for us, so if it meets with approval, then we can easily keep it up. That doesn't stop people from contributing, though.

Date: Wed, 8 Sep 2004 17:22:19 +1000

From: Greg Black <gjb@auug.org.au>

On 2004-09-08, Greg 'groggy' Lehey wrote:
 > On Wednesday, 8 September 2004 at 16:56:14
 > +1000, Greg Black wrote:
 >> Well, I've read Greg Lehey's editorial "The
 >> new face of AUUGN" in the current issue and
 >> it seems worthy of comment. I must say I'm
 >> disappointed the decision has been made "to
 >> stop production of the paper version of
 >> AUUGN by the end of the year."

>>
 >> However, if the cost of printing is so high
 >> and if we really could see a significant
 >> reduction in membership fees in return for
 >> this move, I'm happy to accept the decision.

>
 > This caused a surprising amount of discussion
 > at the AGM. In general, the sentiment
 > expressed was like yours: "pity, but
 > understandable". I assume that this sentiment
 > is not the only one: a significant number of
 > members didn't read it, so they had no cause
 > to complain about the decision.

I've always read it, from cover to cover, and usually within a couple of days of receiving it. It's not always what I want, but I don't complain because I'm not prepared to provide the extra stuff myself and I know just how hard it is to produce a regular magazine.

> But I'd like to remind you about the option of
 > having it printed. If people are interested
 > (but only then), we can do some enquiries
 > about the cost of such printing. It's
 > possible that it would be comparable to the
 > costs of producing AUUGN.

I would not be interested in a printed version if we turned it around into a web-based publication — so long as that version really was produced and accessible. (I note that there's a section of the AUUG web site entitled "AUUGN On the Web" that claims to include the text of AUUGN starting from December 2000, with issues added six months after publication — the last one listed is July 2002, so this is not promising.)

>> I do hope that AUUGN will continue --
 >> preferably on the web rather than in the
 >> form of CDs, which also cost money to
 >> produce and distribute.

>
 > We're currently thinking of both.

Are there plans to engage the membership in the decision making process, or would that be too difficult to manage?

>> I do like the inclusion of conference papers
 >> in the current issue and would hope that
 >> might continue.

>
 > Thanks. That's an easy source of material for
 > us, so if it meets with approval, then we can
 > easily keep it up. That doesn't stop people
 > from contributing, though.

Indeed, but having the conference papers on hand does provide for some meat in the magazine and it does get the papers out to a wider audience.

Cheers, Greg

Date: Wed, 22 Sep 2004 10:01:21 +1000

From: Russell Standish <R.Standish@unsw.edu.au>

To: Greg Black <gjb@auug.org.au>

I would definitely be interested in having AUUGN distributed and archived via the web. There should also be a table of contents distributed via email. I already read web versions of the IT sections of major newspapers, and I also read electronic manuscripts of scientific papers - these I download into my laptop, and I read them as I have time. Interesting papers are filed in a subdirectory of my "read" directory, otherwise I delete them as I read them.

Paper journals tend to just end up hogging space on my bookshelves. CDs just get lost!

Cheers



First Australian UNIX Developer's Symposium Call for participation

AUUG is proud to announce the 2005 UNIX Developers' Conference, to be held in Adelaide on 8 and 9 April, 2005. Attendees will be accomplished programmers who wish to develop software for UNIX-like systems using open source tools. We are planning two concurrent streams. One stream will be aimed at programmers who have little or no experience with UNIX, and wish to learn the UNIX philosophy, environment and tools; the other will be aimed at developers who already have significant experience programming for UNIX, and wish to learn new or advanced tools and techniques.

The tutorial programme for the Newcomers stream will comprise four 90 minute presentations, which will give attendees a solid understanding of the mechanics of developing for UNIX:

- Introduction to the UNIX environment
- Shells and scripting
- Make and gcc
- Debugging with GDB

The Programme Committee invites proposals for tutorials and papers. In order to ensure complete coverage, the programme for the "newcomers" stream is a little more rigid. Our intention is to have four 1½ hour tutorials roughly covering the following topics:

- Introduction to the unix environment
- Shells and scripting
- Make and gcc
- Debugging with GDB

We invite proposals that don't disrupt this framework. For the advanced tutorials and the papers, the following list of topics is intended to illustrate the direction of the conference, but papers on other related topics will be considered:

- Introduction to the UNIX environment

- Shells and scripting languages
- Editors
- Programming languages
- Build tools (e.g. make)
- Version control
- Debugging programs
- Writing graphical programs
- Integrated development environments
- Kernel programming and debugging
- Network programming

This is an opportunity for you to help foster and strengthen the open source developers community in Australia.

Tutorials can be 90 or 180 minutes long, and papers should be 45 minutes, including time for questions. We would prefer a written paper or tutorial notes, for inclusion in the conference proceedings, for all presentations.

Submission Guidelines'

Those proposing to submit papers should submit an abstract and a brief biography, and indicate whether their paper is intended for the Newcomers or the Advanced stream. Those submitting tutorial proposals should submit an outline of the tutorial and a brief biography, should indicate whether their paper is intended for the Newcomers or the Advanced stream, and should clearly indicate the duration of their presentation.

Proposals should be sent in electronic form to the Programme Committee at developers2005@auug.org.au.

Important Dates

Abstracts/Proposals Due	18 December 2004
Authors notified	14 January 2005
Final copy due	26 March 2005
Conference	8 and 9 April 2005

Please refer to the AUUG website for further information and up-to-date details.



First Digital Pest Symposium

Melbourne, 8
February 2005

The inaugural Digital Pest Symposium (DPS1) will be held in Melbourne on Tuesday, 8 February 2004.

The Digital Pest Symposium is a one-day event, focussing on the twin global scourges of “spam” email and malicious software (including, but not limited to, viruses, worms, trojans and spyware). Just about anyone who uses a computer these days (and that’s most of us) has to deal with these in some way, so all are welcome and encouraged to attend, and indeed participate.

Goals

- to promote the sharing of information and experience relating to digital pest countermeasures,
- to raise awareness of the role of good software and systems design in controlling digital pests,
- to raise the awareness of open-source countermeasures within the broader ICT community.

Format

The event is made up of 45 minute talks, comprising mostly speaking time and an allowance for questions. We encourage both technical and non-technical presentations, on topics including:

- anti-virus and anti-spam solutions
- anti-virus and anti-spam software internals
- methodology—vulnerabilities exploited by spammers and virus authors
- user education
- legalities (e.g. fighting spam in the courts)
- ISP strategies
- the role of government

All speakers receive free registration

Call For Participation

Please send abstracts (around 100 words) or expressions of interest by close of business on Friday, 17 December 2004. A formal paper is desirable, but consideration will be given to presentations without a paper if of sufficient quality or interest. Likewise, shorter presentations than 45 minutes will also be considered.

Registration

Registration costs are as follows:

- \$99 for full AUUG members,
- \$125 for associate members, and
- \$150 for non-members.

Please contact AUUG to register.

AUUG Chapter Meetings and Contact Details

City	Location	Other
Adelaide	Marcellinas Pizza Bar 273 Hindley Street Adelaide	Meetings are held at 7 pm on the second Wednesday of each month.
Brisbane	Inn on the Park 507 Coronation Drive Toowong	For further information, contact the QAUUG Executive Committee via email (qauug-exec@auug.org.au). The technologically deprived can contact Rick Stevenson on (07) 5578-8933.
Canberra	Australian National University	For updated information, see http://www.canb.auug.org.au/cauug/
Hobart	University of Tasmania	Chapter appears to be dormant. The last known URL for updated information was http://www.tas.auug.org.au/
Melbourne	Various. For updated information see http://www.vic.auug.org.au/	The meetings alternate between technical presentations in the even numbered months and purely social occasions in the odd numbered months. Some attempt is made to fit other AUUG activities into the schedule with minimum disruption.
Perth	The Victoria League 276 Onslow Road Shenton Park	For updated information, see http://www.auug.org.au/wauug/waug.html .
Sydney	Sun Microsystems Ground Floor, 33 Berry Street (cnr Pacific Hwy) North Sydney.	The NSW Chapter of AUUG holds meetings once a quarter in North Sydney in rooms generously provided by Sun Microsystems. More information at http://www.auug.org.au/nswauug/ .

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/> or call the AUUG office on 1-800-625655.