$2.50

MUUGlines

April 1998 Volume 10, Number 8

This Month's Meeting Topic TBA

I know this sounds a little like last month, but that turned out to be a rather interesting presentation on the BeOS. We will have the usual round-table discussion, and the main topic will be announced in the usual places (newsgroups and our web site, http://www.muug.mb.ca/). A tentative schedule of future meetings for this MUUG year (to June 1998) is:

April – TBA
May – Math Software and Tools for UNIX (Michael Doob)
June – Applied Multimedia Training Centre, 495 Portage

The door prizes this month are O'Reilly & Associates T-shirts and the Perl Resource Kit, Unix Edition. See the review in this issue for more details.

Our meeting this month is Tuesday, the 14th of April. We meet at IBM Canada's offices in the TD Centre, at the corner of Portage and Main. We'll gather in the lobby on the main floor – please try to be there by about 7:15 PM. Steve Moffat will then take us up to the meeting room just before the meeting starts at 7:30. Don't be late, or you may not get in.

Parking is available either in the parkade behind the TD building, off Albert Street, or in the ground level lot just north of the TD building. Entrance to the lot is from Albert Street, behind the parkade. Either way, parking is a $1.25 flat rate for the evening. You purchase your ticket from a dispenser, so make sure you've got exact change – a loonie and a quarter, or 5 quarters.

MUUG Site Ranks #14!

Gilbert Detillieux, our webmaster, came across this interesting tidbit while looking at a referrer report, generated by analog (a web server log file analyser), based on a log of sites that refer to ours (http://www.muug.mb.ca/).

The most frequently logged referrer page was this one:
http://www.100hot.com/unix/unix.chtml

The page contains the following list...

100hot Unix Sites

The most popular Unix related sites for the week of Friday, February 6, 1998 :

Rank	Site
1	Linux Online - The Linux Home Page
2	LinuxHQ
3	FreeBSD Inc.
4	Linux NOW!
5	Debian GNU/Linux - The Universal Operating System
6	The Linux Kernel Archives
7	www.freebird.org
8	Enhanced Software Technologies' Home Page
9	Ready-to-Run Software, Inc.
10	Linux International
11	The NetBSD Project
12	Unixware
13	HP Professional Web Edition
14	MUUG Home Page
15	Freenix: Home of the Free Unices
16	SCO World Magazine
17	CUUG - Home Page
18	Sys Admin Web Site
19	Open Systems EXPRESS! - UNIX and Windows NT Specialists.
20	Welcome to JRD.COM!

[... and so on ...]

Gilbert remarks "Cool! We ranked in the top 15 out of 100! Ahead of CUUG, I might add. :) Of course, I have no idea how their rankings are determined. I do find it odd that Red Hat doesn't show up in the list, even though many other Linux distribution sites do."

Note that the list has still not been updated, and there are only 34 items on it!

Kernel Choices

Should You Try a Development Kernel?

Contributed by CmdrTaco on Thursday April 02, @10:59

Raj Dutt has done a nice writeup on why you should, or shouldn't try a devel kernel. If you've never tried one, and wonder why or even if you should try, you should read on. Especially timely considering yesterday's feature freeze [Ed. note: Linus Torvalds announced recently no new features will be added during the current development phase], we're on the home stretch for 2.2.0. The following was written by Slashdot Reader Raj Dutt

In the world of Linux kernel development, both Development-releases and stable-releases are developed concurrently. This is an extremely effective method of development as it provides the basis for improvement at both unbelievable speed and quantity. However, since the model does not follow the traditional singular develop-release-develop-release cycle, it can create great confusion among users.

Before detailing the implications of development kernels, it might be beneficial to some to understand the kernel versioning convention:

(Generation.Major.Minor)

It is the Major number that is the key here. An even Major version represents a stable, release kernel and an odd Major version is indicative of an unstable, development kernel. For example, version 2.0.33 means "33rd minor release of the Second generation, 0 (even, therefore stable) Major version Linux kernel." Using the same classification, version 2.1.91 means "91st minor release of the Second generation, 1 (odd, therefore unstable) Major version Linux kernel"

Hence, the 2.2.x series, which will be released starting with 2.2.0 in the near future, will be the next Major version of stable kernel releases. With that out of the way, let us examine the PROs and CONs of using development kernels.

The PROS

Nobody (well almost nobody) will deny that the current crop of release kernels work great. Why then should one jump ship and run something that is not nearly as stable as 2.0.33? The development kernels offer three main advantages that often convince people to go with them :

Firstly, after 91 development kernels, a LOT of new hardware is now semiofficially natively supported. If you have a device that you can't use because of the simple lack of a driver, check out the 2.1.x series, there is a good chance that your hardware will be among the hundreds (Color Quickcams, Radio Cards and NCR53C9x SCSI just to name a few) of new components supported.

Secondly, there are numerous technical arguments for the superiority of the latest crop of creations. These include:

Faster access to user memory (covering the 4GB user space in its entirety)
Unbelievable improvements in SMP (Symmetric Multi-Processor)
Major tcp/ip updates - (read: faster better networking)
Better PCI access and support
Major disk swap / buffer updates (including dcache)
Lots of improvements for non x86 (intel-like) architectures
Improved x86 IRQ handling
Lots of filesystem code updates - FAT32 is natively supported, improved VFAT etc.

There are lots more. Literally hundreds. For a complete listing of changes, visit Cutting Edge Linux - a great site which keeps track of the additions on a per kernel basis.

Thirdly, in addition to the technical benefits and new device support that you may grow to love, you can bask in the knowledge that you are helping the Linux Kernel development. By sending in bug reports, you are aiding in the quality control department - something extremely important when talking about code which needs to be bulletproof.

The CONS

The very word Development carries some connotations that cannot be simply ignored. To put it bluntly, when you install a development kernel you are venturing into uncharted territory. Don't expect to receive sympathetic support when something goes wrong. Development is proceeding at an unbelievable rate and nobody is going to stop to hold your hand. "It's broken? Well it's a development kernel, what did you expect?"

A lot of people run Linux for the stability. Be prepared to give some of that up if you choose to run with 2.1.x. Since they have not been tested (at least not as thoroughly as mainstream release kernels =), your mileage with development kernels will vary. Even experiences using the same kernel on different machines can be very different. After running version 2.1.91 for just under a week, I have noticed a few problems. None of them have been serious enough to bother reinstalling 2.0.33 though. Namely, I had to upgrade my pppd to 2.2.3 and twice I had to bear some serious disk thrashing for no apparent reason.

IN CONCLUSION

Obviously, a non release kernel is not for everyone. If you run a mission critical server or have a strong need for stability, stick with the even-numbered kernel releases. But if, like me, you have a personal workstation to fool around on and are always looking to try something new, then you might wish to consider "upgrading" to a development kernel

When it comes down to the crunch, It's really a question of balance. If you can sacrifice a little stability and the possibility of a few problems for the excitement and improvements of being right on the bleeding edge, go for it.

Ed: With yesterday's feature freeze, hammering out those bugs is very important, if you're looking for something new to try, the next few releases in the 2.1.* series should keep getting more stable for the soon-to-be-ready 2.2.0.

The Announcement

As referred to above, here’s Linus’ announcement of the kernel new-feature freeze:

Date: Wed, 1 Apr 1998 17:50:18 -0800 (PST)
From: Linus Torvalds <torvalds@transmeta.com>

To: Kernel Mailing List <linux-kernel@vger.rutgers.edu>
Subject: Linux-2.1.92 - Feature Freeze

Ok, there's a fairly large patch out there, but as of 2.1.92 I think we have a real feature-freeze, and we'll try to get a real code-freeze going soon. There are known problems with the sound drivers etc, which is why a code-freeze isn't the best suggestion right now, and there are probably still bugs with some of the new code, but I'll freeze new features for the upcoming 2.2 kernel.

Yes, some people will scream bloody murder, but others will be relieved that it finally happened. Thanks especially to David Miller who has been doing a great job of getting the TCP stack from its problems just a few weeks ago to really shining new heights. That was my main worry about 2.2 not all that long ago, and was the main reason for having such a slushy period for a while.

2.1.92 does:
- ISDN updates
- alpha update (yes, SMP finally works, although not really stable yet)
- networking fixes
- "getcwd()" system call (not very long, the dcache makes this so trivial it is scary)
- the mm responsiveness updates (they were in 2.1.92-pre2, people seemed to have found them very effective)
- some other (mainly driver updates)

Please do test it all out. Feature-freeze doesn't mean that it is supposed to be bug-free yet, but it does mean that we should be moving into bug-fixing mode in quick order.

And no, this is not an April 1 thing. But this way I can use April 1 as an excuse if something doesn't actually compile.

Linus

Perl Resource Kit

The Perl Resource Kit – Unix Edition from O’Reilly & Associates (http://www.oreilly.com/) is a Can$212.95 boxed set of Perl reference materials that any Perl programmer will find very valuable.

Included in the box are four printed volumes, a recent issue of The Perl Journal (http://www.tpj.com/) and the inevitable CD-ROM. The printed documentation is voluminous at about 1800 pages, most of it in the Perl Module Reference Volumes I & II. These two volumes are the core of the set, providing the first serious collection of Perl module documentation in printed form. Previously, this documentation was only available by browsing the source, which is not always the most convenient way of locating this information.

Not every module is included, but there are a broad array of some of the most useful Perl modules available in the CPAN (Comprehensive Perl Archive Network). The layout is excellent. If you note something missing from the book, you’ll likely find it on the CD-ROM. If you don’t find it there you will find instructions on how to locate one of the many mirrors of the CPAN.

There is a great deal of information in this kit, but not included is instruction on programming in Perl. For that, see one of the many other references on the subject, including the one from O’Reilly. However, there is a volume in the kit devoted to helping you get the most out of the modules that you now have available to you. If you are just starting out, your best bet is to get one of the aforementioned Perl books, run the installation program on the kit’s CD to setup the latest Perl binaries, and start learning.

The Perl Utilities Guide is the place to start, as it explains how to install whatever you need from the CD. That only takes you as far as chapter two, though. The remainder of the book is given over to JPL, the Java/Perl interface written by Larry Wall, creator of Perl.

The final volume is Programming with Perl Modules. Anyone new to modules in Perl will find this a goldmine. You are led carefully through many interesting examples, chapter by chapter showing how anything can be done in Perl! That’s the impression I got. Topics like building graphical interfaces with Perl/Tk, Mail and MIME modules, News, database and Web applications and more are all clearly presented. They almost amount to mini-tutorials on the topics, let alone the modules themselves! Unfortunately, these volumes are not available separately.

The software included requires Solaris 2.5.*, Linux 2.0.*, a CD-ROM drive, 16 MB RAM, 20 MB of disk space for the binaries, and 500 MB of disk space if you want to install all of the included CPAN archive. Note: to run the automatic installer, you need X running as well. If not, they include instructions on how to install it manually.

Overall rating: a must-buy for anyone with a serious interest in Perl.

Any Takers?

Sun cuts Solaris upgrade prices

from PC Week March 6, 1998 2:55 PM PST

Sun Microsystems Inc. is offering discounted pricing for its Solaris operating system in an attempt to lure customers away from its competitors.

During the rest of this month, Sun will cut prices on its Solaris Version 2.6 by up to 70 percent for customers wishing to upgrade from competing operating systems, including SCO Unix, SCO OpenServer, SCO UnixWare, OS/2, Linux and Sun's own Interactive Unix. With the offer, customers will get a two-year subscription to Solaris upgrades, including Sun's forthcoming release, Solaris 2.7, officials said. The deal is also available to customers looking to upgrade from older versions of Solaris.

Sun, of Mountain View, Calif., can be reached at (800) 786-7638 or www.sun.com.

The hard and true facts about NT's BSOD broken hardware mythology

by Nick Petreley, InfoWorld

Readers had a lot to say about the frequency with which I've been experiencing the Windows NT blue screen of death (BSOD). Many readers politely suggested that I wipe Windows NT off my system and switch to this or that other operating system. Others were less polite (read: flame), telling me the BSOD is an indication of a hardware problem and advising me to stop complaining about NT and fix my hardware.

A BSOD can occur for a wide variety of reasons other than hardware configuration conflicts, failing hardware, or bad memory. A BSOD can result from a wide variety of problems including kernel bugs, memory resource depletion, a corrupted registry, driver bugs, or even the inability of the OS to handle an unexpected condition. The list of bug codes at the URL http://support.microsoft.com/support/kb/articles/Q103/0/59.asp will give you an idea of just some of the potential problems that lead to a BSOD.

The reason most people assume a BSOD is hardware-related is because errors that cause a BSOD often occur in hardware drivers. This is true whether the fault is due to hardware failure, misconfiguration or a bug in the driver. Indeed, this is why you see the BSOD more often in NT 4.0 than NT 3.51. The Windows NT 3.51 kernel was protected from more drivers than is NT 4.0's. Microsoft now allows display drivers to have unrestricted access to critical parts of the system, for example.

When a hardware driver touches part of the system it isn't supposed to access, the damage to the system is done before Windows NT can isolate the error, prevent it, or correct it. So Windows NT figures it is safer to crash with a BSOD than to continue running. As odd as it sounds, this is typical behavior for an operating system. Many operating systems are better than Windows NT at protecting themselves from runaway hardware drivers, but they react the same way when the rules are broken.

Aside from the fact that hardware isn't always involved, the problem I have with the "fix the hardware" answer is that it implicitly excuses Windows NT from responsibility when a BSOD occurs, as if one shouldn't expect the operating system to handle hardware-related problems. I, for one, expect more from an operating system than that. And I know from experience that more is possible.

For example, I can get a BSOD simply by accessing (or, in some cases, simply inserting) any of three improperly written CD-Recordable disks that I had.

The system automatically tries to read the CD table of contents, but never succeeds. The result? BSOD. (Yes, I have tried SCSI card reconfiguration, new drive settings, latest drivers, hot-fixes, and so on.)

The same disks cause Windows 95 to hang hard. But of the three disks that neither Windows 95 nor Windows NT can handle, Linux mounts one disk successfully but, as one would expect, it fails to read the improperly written files. The second disk causes Linux to report that the SCSI device isn't responding. When I mount the third disk, Linux resets the SCSI bus. I issue a kill command to stop the mount process, then I continue working as usual.

When I need something done, what matters to me is what works and what doesn't. In many cases, Linux works and NT doesn't. But to those defensive souls who want to feel better about NT by blaming the BSOD on hardware, my return advice is to relax. I don't want a Linux-only world any more than I want an NT-only world. When I complain about NT, it's because I want Microsoft to fix it, not eliminate it.

A former consultant and programmer, Nick Petreley is editor in chief at NC World (http://www.ncworldmag.com). Reach him at nicholas_petreley@infoworld.com.

Contact Information

To contact the MUUG board for membership information or anything else, send e-mail to board@muug.mb.ca. We have a Web presence as well, at http://www.muug.mb.ca/, where you can find all kinds of information, including details of upcoming and past meetings and presentations and references related to them. E-mail the editor at editor@muug.mb.ca.