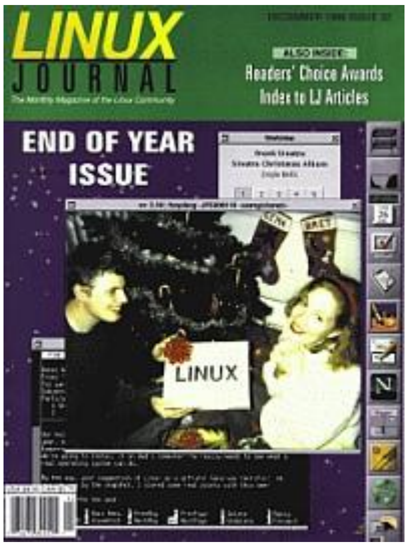
Advanced search

# *Linux Journal* Issue #32/December 1996



Archive Index

Advanced search

# Lurking with PGP

**Michael K. Johnson**

Issue #32, December 1996

Phil Zimmermann's PGP program was written primarily to allow people to be quite sure their private communications remain private. The messages are encrypted so that only the intended recipient is able to read them—as long as users have read the manual and paid heed to its security warnings. Pretty Good Privacy.

Lurking is a time-honored practice. If you don't want to seem foolish when you join a newsgroup, you *lurk* there, reading articles without posting any, learning what is expected of participants in that newsgroup. The same goes for mailing lists. Some people never quit lurking—and on the Internet, and on Usenet, that's all right.

While reading Usenet news or a mailing list, you have probably seen a PGP-signed message. It looks something like this (the signature has been slightly modified to fit in the magazine):

```
-----BEGIN PGP SIGNED MESSAGE-----
This is an example PGP-signed message.
-----BEGIN PGP SIGNATURE-----
Version: 2.6.2
iQCUAwUBMYlOKyd5aW9FNqjdAQHVpAP4vrpL2MoIm3MFk
95e7mRaYwRoKSL4lpCDR8WvDo13ICvaa/IbYxZwH/5IFM
vve7a+HnFPFd7pKegsJxSc8MgFnnBCxTJAEeimLCmZ+DA
VHPwqnEjxdeTWvwoysg2hm89CUOxvn4ArbG3yntlRlL+k
0HPjV+D0Uvi+LN0sNroi5A==
=G3yB
-----END PGP SIGNATURE-----
```

You can read the message just fine by ignoring all the PGP-specific stuff. But you can also make use of it by learning a few simple commands—and learning *how* to use them.

## Simple Security?

Are you intimidated by PGP's manual? Perhaps you should be. Unlike other software products, with which it is Okay to fool around in order to learn it piece by piece as you need it, cryptographic software needs to be well-understood to be truly secure. Using it incorrectly can give you a feeling of security—but that's worse than not using it at all, because with that false sense of security, you will use it to send sensitive information that you would never have dreamed of sending via e-mail otherwise. The manual section on key management isn't that long, but you do have to understand it, and to do that, you do need to read quite a bit of the manual.

If you want to write PGP-encrypted e-mail, you *have* to read the manual. This article can't substitute for the manual in teaching you how to keep your e-mail private. You have your choice of the free manual distributed with the source code, the printed version of the official manual sold by MIT Press, or the O'Reilly book, *PGP: Pretty Good Privacy*, ISBN 1-56592-098-8, written by Simson Garfinkel.

## So Why Read This Article?

So you can "PGP-lurk". You don't have to send PGP-encrypted e-mail in order to take advantage of PGP signatures; you can simply verify that messages sent by other people are *really* sent by them, and not by someone else masquerading as them. But in order to do that, you really need a little bit of background. Don't worry, this won't hurt a bit...

PGP is based on **public key cryptography**. Each person has a public key that everyone else needs to know, and a private key which must not be known by anyone else. A message is encrypted with both the sender's private key and the recipient's public key. The message can only be decrypted with the recipient's private key, which only the recipient knows.

It is possible, however, to simply *sign* a message without encrypting it. The sender's *private* key is used to sign the message, and the sender's *public* key is used to validate the signature. Notice that the recipient doesn't need any keys at all to validate a signature. As long as recipients are able to trust that their copies of the sender's public key are correct, they can trust the signature.

Since you don't need to protect the secrecy of a private key in order to use PGP to validate signatures, you don't need to read a book to learn how create a private key and keep it secret. You just need to know how to find other people's public keys and know how to use PGP to check signatures.

If you have read the newsgroup comp.os.linux.announce, you will have noticed by now that every message posted there is "digitally signed by the moderator, using PGP". Lars Wirzenius, the moderator, allows you to be sure that he approved the articles for posting by PGP-signing them. In his signature, he states, “Finger wirzeniu@kruuna.helsinki.fi for PGP key needed for validating signature.” If you run the command **finger wirzeniu@kruuna.helsinki.fi**, you will see something resembling Listing 1. Don't type that key in from the listing; if the key is going to do you any good, you will be able to get it over the internet. If you don't like finger, you can get it from his home page, www.cs.helsinki.fi/~wirzeniu/.

## I'm Still Not Convinced

If the possibility of a forged posting on the comp.os.linux.announce newsgroup doesn't worry you enough to make you want to install PGP, consider another situation. Suppose you see a newsgroup posting or e-mail message that looks like it is from Ted Ts'o (one of the main Linux developers), and which claims that:

- A security hole has been found in Linux,
- It is being exploited by crackers who are destroying the systems they crack into,
- An attached kernel patch will solve the problem, and
- You should install the patch on your system(s) as soon as possible.

Further suppose that the patch is beyond your comprehension. *Do you install the patch?* Let's assume that Ted is trustworthy (a pretty good assumption). How do you know that it is really Ted who made the posting or sent the message? Perhaps the patch really **creates** a security hole through which a cracker could attack, and a cracker is forging the message and pretending to be Ted in order to convince you to apply the patch.

You can know because Ted PGP-signed his patch, and because you have a copy of his public key. No one could sign it correctly without a copy of his private key, and Ted is a security expert who keeps his private key very safe.

## Getting Keys

In order to check a PGP-signed message, you need to have the public key. Where do you get the key? You want to get the real user's real key, not a fake one produced by a forger who is using the fake key to impersonate the sender.

Fundamentally, you need to get the key from a different source than the message. News and mail are notoriously insecure; anyone can fake an e-mail or news message, and anyone who wants to read this article to learn about PGP will probably not notice that the message is fake, and probably won't know how to tell the difference.

If someone is forging messages that are supposed to look like they come from someone else, and they are including PGP, they will also be attempting to propagate a fake key for the user as whom they are masquerading. This means that in an ideal world, you will collect public keys *before* you need to use them to verify a message. However, there's a good chance that you don't have the key yet when you want to read the message. Where do you look to find a key, and how do you know that it is correct?

Sometimes you can get a key from someone's home page on the Web. If you do that, you have to weigh the possibility that someone has broken security on that system and corrupted the key there. If you get the key well before you want to verify a suspicious message, chances are pretty good that the key you get will be good. The same goes for keys retrieved via finger. If you think you will ever want to verify a comp.os.linux.announce posting, get the key now. If you are paranoid, keep checking for the next few days to make sure there is no announcement that the key was forged or compromised.

One place to find lots of PGP keys is on **BAL's PGP Public Key Server**, at www-swiss.ai.mit.edu/~bal/pks-toplev.html. This web site has a large database of public keys, and the person you are looking for may have submitted the public key to the database. There's no security; anyone could post an update to anyone else's public key, even forgeries. You still have to verify the keys you get from BAL's server; it's only a convenient point of exchange.

None of these techniques ensure that the key you get is good. All security is a matter of degree, and this technique provides, for most purposes, a *reasonable* level of assurance that you have the right key. Don't bet the family fortune on it.

Essentially, PGP allows you to *extend* trust you already have. If you trust your brother to tell you the truth in person, and **he** has verified that your copy of his PGP public key is correct, then you can be rather sure that a message that is correctly signed with his private key really comes from him. You can be only as sure of that, however, as of his ability to keep his private key secret.

## Signed Keys

If your brother is sure that his copy of Joe's public key is correct, and you trust your brother to give you a good copy of Joe's public key, then you can be fairly sure that a message that is signed with Joe's public key really comes from Joe. This is a very useful notion, and PGP supports it.

To verify that he is positive that Joe's public key really belongs to Joe, your brother can *sign the key* with his own private key. If you are certain that your copy of your brother's public key is correct, and your brother has signed Joe's public key, then *if you trust your brother's judgement in verifying keys* you can be mostly sure that your copy of Joe's public key is correct. Now that you know Joe's key, if you trust *his* judgement, you can be reasonably confident of the veracity of public keys which *he* has signed.

PGP users have organized PGP-key-signing gatherings (parties, BOF (Birds of a Feather) sessions at conferences, whatever) in order to meet face to face and sign each other's keys. As people go to different gatherings in different places, it becomes easier and easier to say, “I trust my brother, and my brother has signed Joe's key, and Joe has signed Jean's key, so I can be pretty sure that this message signed with Jean's key is really from Jean.” This concept has become known as the “web of trust”.

Don't go off the deep end when it comes to the web of trust. PGP doesn't make people trustworthy. As Zimmermann says in the PGP manual, “Trust is not necessarily transferable; I have a friend who I trust not to lie. He's a gullible person who trusts the President not to lie. That doesn't mean I trust the President not to lie. This is just common sense.” If you don't trust someone about other things, there's no reason to trust their signatures of other people's keys. They may be duped or careless or lying themselves.

## Verifying Keys

When people verify that their copy of someone else's key is correct, they don't laboriously check 1000-10000 characters of the ASCII representation of the key. Instead, they compare an abbreviated form called a *fingerprint*. Each key is verified by its fingerprint, which is represented by 32 hexadecimal characters. The possibility of two keys having the same fingerprint is so close to nil that you don't even have to consider it. So if the fingerprints match, the keys match.

Before checking fingerprints, it is best to quickly check the *ID*. The ID is even more abbreviated; it consists of only 8 hexadecimal characters. If, for instance, a user uses two different sets of PGP keys, the ID is used to differentiate between them.

## Examples

Since we started with comp.os.linux.announce, let's show how you can easily retrieve the public key for verifying those messages. (I assume that you have PGP installed. If you don't, I'll explain later how to do that. I just want to keep things simple for now.)

PGP normally tries to do the right thing without being told. If you feed it a public key, it figures that you want to add it to your “public key ring”, available for inspection or use at any time. Normally, you call it with the name of a file, so try this:

```
finger wirzeniu@kruuna.helsinki.fi > /tmp/cola
pgp /tmp/cola
```

If PGP complains “Key ring file '*home*/.pgp/pubring.pgp' cannot be created”, then you should create the directory ~/.pgp and try again:

```
mkdir ~/.pgp
pgp /tmp/cola
rm /tmp/cola
```

PGP will ask if you want to add the key to your public key ring; answer yes. When it asks whether you want to certify any keys yourself, answer no. You can't do that without having created your own public and private key. You will still be able to verify messages, but you can't use some of PGP's built-in features. This will look like Listing 2.

The comp.os.linux.announce key is now on your public key ring, and you can now use it to verify posts made to comp.os.linux.announce.

## Checking Signed Messages

Now let's try checking the signature on a message posted to comp.os.linux.announce. Using your newsreader, save a message to a file, preferably with the extension “.asc” (short for ascii). Let's save it in the file cola.asc, and then call PGP to check the signature:

```
pgp cola.asc
```

PGP is verbose (as usual), and checks the signature. Part of what it says lets us know that the signature checks out Okay—but it warns that it can't be sure, because you don't have a key of your own:

```
File has signature.  Public key is required to
check signature.  Good signature from user
"Lars Wirzenius <wirzeniu@cs.helsinki.fi>".
Signature made 1996/05/01 11:36 GMT
WARNING: Because this public key is not certified
with a trusted signature, it is not known with
high confidence that this public key actually
belongs to:
"Lars Wirzenius <wirzeniu@cs.helsinki.fi>".
```

If you want to get rid of that warning, you will have to create your own keys, and before you do that, you ought to read the manual and understand it.

PGP also saves a copy of the message without the signature in a file called cola —it strips the .asc from the filename. If you had saved the original message in a file named “cola”, it would have asked you for a different filename to put the unsigned message in. Unfortunately, at the present time, the only way to check a signature without creating a new file containing the unsigned text is to press CTRL-C when PGP stops to ask you what to do.

## Checking Fingerprints

In order to verify that your copy of a key is the right one, you need to put the key on your keyring, and then use PGP to print its fingerprint. Use the command:

```
pgp -kvvc user_id
```

where ***user_id*** is either part of the recipient's e-mail address or the actual 8-character key ID. Listing 3 shows part of the output of this command for the key used to sign Announcements of the Linux Emergency Response Team (ALERTs), which are used to announce security issues related to Linux. This (**-kvvc**) shows a lot of information for each key; you can get a more concise listing with the command:

```
pgp -kvc user_id
```

Now try to print the fingerprint for the key Lars uses to sign comp.os.linux.announce postings. You can check it below.

Once you have generated the fingerprint, you need to compare it to a trusted version. That might mean the fingerprints listed in this article, or it might mean calling the sender on the phone, or it might include a fingerprint listed in a book. There are many options—it's up to you to judge if the option you choose provides good enough verification for your purposes.

Here is a list of IDs and fingerprints for important and useful keys in the Linux world:

- Lars Wirzenius's comp.os.linux.announce key:**4CBA92D1 E7FA89856D9B781D F530EBFBD811CA3F**
- Linux Security FAQ Primary Key used to verify ALERTS:**ADF3EE95 AB4FE7382C3627BC 6934EC2A2C05AB62**
- Ted Ts'o's signature key used to sign other people's keys (Ted organizes PGP-signing BOFs at conferences, and so has signed many other people's keys):**466B4289 9C056649DF837EEF D8AC7542A2334B91**
- Your humble author, who will also be organizing PGP-signing BOFs at some conferences, and wants to show off:**4536A8DD 2AEC88084064CED8 DDF8122B61438315**

(Please note that in order to fit the fingerprints into the article, we have removed many of the spaces. The spaces are just there to help you read the fingerprint, and the fingerprint is the same with or without spaces.)

Linux developers are starting to talk about listing their PGP key IDs and fingerprints in the CREDITS file in the Linux kernel source code.

## Pretty Good Privacy

There's a pretty good chance that after using PGP to quietly verify signatures for a few years, you will at some point want to use it for its original purpose—privacy. Perhaps you want to send a password to someone. Maybe you simply want to send your credit card over the Internet. You don't have to be a hero of the information underground to want to keep your mail private; there are many prosaic reasons as well. If you are already used to using PGP to verify signatures, you will not find it difficult to learn how to use PGP to encrypt your email. Just read the manual carefully so that your communications are truly secure.

## Installing PGP

Installing PGP is a bit of a mess, partially because there is a patent that is honored in the US and Canada on the public key algorithm used, and partially because of the US's insane ITAR regulations. If this were an editorial, I'd have a lot to say about how incredibly stupid the US government is acting in this case, but this isn't an editorial, so I won't say a word on the subject...

If you have Red Hat Commercial Linux, life is easy. You can install PGP from an RPM available via anonymous ftp from ftp.hacktic.nl in the /pub/replay/pub/redhat/ directory. For those outside the US, you can use either the US version or the international version; for those in the US, you can only legally use the US version because of patent law. As of this writing, the current version number of both versions is 2.6.3, and you just have to choose between pgp-2.6.3i-1.i386.rpm (the international version) and pgp-2.6.3usa-2.i386.rpm (the US version). There are also README files in that directory that explain the situation more fully. You also get one more benefit: since version 3.0.3 was released, all official RPM's created by Red Hat are PGP-signed so that you know you have the official version. Installing PGP will allow that feature to work.

Life is also easy if you use Debian. There are .deb files available for both the international and US versions available in the non-free directory of selected archive sites. If you live outside the US, please download your copy from a Debian archive outside of the US to avoid causing Debian legal trouble. You can get a list of archive sites by connecting to ftp.debian.org with ftp. As of this writing, the file you want is pgp-i-2.6.2i-5.deb (the international version) or pgp-us-2.6.2i-5.deb (the US version). A new version using the ELF binary file format will probably be available with the ELF-based Debian 1.1 when it is released.

With other distributions, you will probably have to build PGP from source. You can get the source via ftp from net-dist.mit.edu in the /pub/PGP/ directory. However, MIT makes you jump through several hoops to make sure that you are really a US resident to protect themselves from over-eager US law enforcement officials. Instructions for building PGP are included, and I wish you good luck.

Warning: Within the US, you can use the free version of PGP only for non-commercial purposes. For commercial purposes, you are required to buy a copy of ViaCrypt's PGP. You can reach ViaCrypt at viacrypt@acm.org or (800)536-2664, or you can buy the product from the company that originally ported ViaCrypt's PGP to Linux, SSC (*LJ*'s publisher).

Michael K. Johnson is only slightly paranoid... His public key ID and fingerprint are listed above; his public key is registered with Bal's public key server.

Archive Index Issue Table of Contents

Advanced search

# *Linux Journal* Archives

**LJ Staff**

Issue #32, December 1996

*Linux Journal* presents a complete listing of articles from issue 1 through issue 31.

Many *LJ* readers think of *Linux Journal* more as reference material than as a magazine. For these poor souls, keeping track of published articles is becoming increasingly difficult.

So, in the interest of helping you to keep track of your extensive collection of *LJs* (and, perhaps, convincing you to add to your extensive collection with “must have” back issues) we have compiled an index of all our articles from issues 1 through 31.

The listing appears in the following format:

- Article Title **[Issue #:Page#]**
    - by Author
    - URL (if available on our web site or other URL)

For those of you so technical that you eschew a paper index, all the information listed here is also available at our web site. We have a search engine that does keyword searches on all the issues of *Linux Journal* at http://www.linuxjournal.com/search

## Announcements

- Cool ... it Works with Linux: Tell the World You Support Linux **[26:55]**
    - by *LJ* Staff
    - /article/2405

    - /article/1143

- Cyberia, Life in the Trenches of Hyperspace **[5:44]**
    - by Putnam Barber
    - /article/2816

- Firewalls and Internet Security: Repelling the Wily Hacker **[6:36]**
    - by Danny Yee
    - /article/2837

- HTML for Fun and Profit **[18:15]**
    - by Brian Rice
    - /article/1141

- IPv6: The New Internet Protocol **[25:45]**
    - by Danny Yee
    - /article/5541

- Inside Linux **[31:63]**
    - by Richelo Killian
    - /article/0123

- Internet Public Access Guide **[3:23]**
    - by Morgan Hall
    - /article/2781

- Linux Anwenderhandbuch Leitfaden fur die Systemverwalktung **[8:53]**
    - by Martin Sckopke
    - /article/0015

- Linux Configuration and Installation **[23:56]**
    - by Scott Wegener
    - /article/0084

- Linux Installation and Getting Started **[1:10]**
    - by Phil Hughes
    - /article/2745

- Linux Universe **[21:47]**
    - by Christopher Boscolo
    - /article/0085

- Linux vom PC zur Workstation Grundlagen, Installation und praktischer Einsatz **[8:52]**
  - by Martin Sckopke
  - /article/0015
- Making TeX Work **[8:50]**
  - by Vince Skahan
  - /article/0006
- Newton's Telecom Dictionary **[3:23]**
  - by Phil Hughes
  - /article/2780
- Prime Time Freeware for Unix **[24:10]**
  - by Preston Brown
  - /article/1243
- Running Linux **[14:8]**
  - by Grant Johnson
  - /article/0066
- Seamless Object-Oriented Software Architecture **[22:36]**
  - by Dan Wilder
  - /article/1214
- Sendmail: Theory and Practice **[16:12]**
  - by Phil Hughes
  - /article/0073
- Teach Yourself Perl in 21 Days **[19:15]**
  - by David Flood
  - /article/0077
- The Future Does Not Compute **[20:58]**
  - by Danny Yee
  - /article/1185
- The HTML Sourcebook **[18:15]**
  - by Brian Rice
  - /article/1141

- The Linux Sampler **[13:58]**
    - by Harvey Friedman
    - /article/0061
- The Tcl and the Tk Toolkit **[11:60]**
    - by Phil Hughes
    - /article/0038
- The Unix Philosophy **[17:51]**
    - by Belinda Frazier
    - /article/2877
- The Web in Print: Civilizing Cyberspace **[28:51]**
    - by Danny Yee
    - /article/1301
- The Whole Internet User's Guide and Catalog **[5:44]**
    - by Putnam Barber
- The Mosaic Handbook for the X Window System **[12:52]**
    - by Morgan Hill
    - /article/2689
- UNIX—An Open Systems Dictionary **[8:52]**
    - by Laurie Tucker
    - /article/0016
- Unix Systems for Modern Architectures **[9:30]**
    - by Randolph Bentson
    - /article/0034
- X User Tools **[15:8]**
    - by Danny Yee
    - /article/1092
- Your Internet Consultant **[11:59]**
    - by Phil Hughes
    - /article/0039

    - /article/1279

- Backup Strategy **[22:15]**
    - by Malcolm Murphy
    - /article/1208
- Basic FVWM Configuration, Part 1, Getting (fvwm) Started **[27:50]**
    - by John M. Fisk
    - /article/1198
- Basic FVWM Configuration, Part 2, Customizing the Desktop **[28:54]**
    - by John M. Fisk
    - /article/1236
- Beyond Your First Shell Script **[28:27]**
    - by Brian Rice
    - /article/1299
- Building a Linux Firewall **[24:49]**
    - by Chris Kostick
    - /article/1212
- Choosing an Internet Service Provider **[24:26]**
    - by Michael K. Johnson
    - /article/1001
- Compile C Faster on Linux Using lcc **[25:32]**
    - by Christopher W. Fraser and David R. Hanson
    - /article/5540
- Connecting Your Linux Box to the Internet **[9:18]**
    - by Russell Ochocki
    - /article/1001
- Diff, Patch and Friends **[28:33]**
    - by Michael K. Johnson
    - /article/1237
- Disaster Recovery **[4:10]**
    - by Mark Komarinski
    - /article/2787

- EZ as a Word Processor **[4:5]**
    - by Terry Gliedt
    - /article/2786
- EZ for the Programmer **[5:5]**
    - by Terry Gliedt
    - /article/2803
- Eagles BBS **[4:16]**
    - by Ray Rocker
    - /article/2789
- Efficient, User-Friendly Seismology in the Field **[16:25]**
    - by Sid Hellman
    - /article/1131
- Emacs: Friend or Foe? **[5:9]**
    - by Matt Welsh
    - /article/2801
- Ethernetting Linux **[12:12]**
    - by Terry Dawson
    - /article/1051
- Filters: Doing it Your Way **[27:23]**
    - by Malcolm Murphy
    - /article/1224
- Flexible Formatting with Linuxdoc-SGML **[18:51]**
    - by Christian Schwarz
    - /article/1127
- Flicker-free Animation Using Java **[30:26]**
    - by Paul Buchheit and Michael J. Hammel
    - /article/0169
- Hints and Tips for Optimizing Linux Disk Usage **[2:1]**
    - by Jeff Tranter
    - /article/2751

- Hints and Tips for Optimizing Linux Memory Usage **[3:11]**
    - by Jeff Tranter
    - /article/2770
- HTML Forms: Interacting with the Net **[16:34]**
    - by Eric Kasten
    - /article/1096
- HTML: A Gentle Introduction **[15:35]**
    - by Eric Kasten
    - /article/1081
- IP Masquerading with Linux **[27:15]**
    - by Chris Kostick
    - /article/1238
- Index of *LJ* Articles **[20:22]**
    - by *LJ* Staff
- Interview with James MacLean **[5:15]**
    - by Michael K. Johnson
    - /article/2802
- Interview with Linus, the Author of Linux **[1:4]**
    - by Robert Young
    - /article/2736
- Interview with Patrick Volkerding **[2:10]**
    - by Phil Hughes
    - /article/2750
- Interview with Fred Van Kempen **[3:16]**
    - by Phil Hughes
    - /article/2772
- Interview with Mark Bolzern **[14:19]**
    - by *LJ* Staff
    - /article/1087
- Introducing Modula-3 **[8:9]**
    - by Geoff Wyant

    - /article/0009
- Introduction to Eiffel **[14:34]**
    - by Dan Wilder
    - /article/1077
- Introduction to Gawk **[25:46]**
    - by Ian Gordon
    - /article/1156
- Introduction to LINCKS **[11:26]**
    - by Martin Sjolin
    - /article/1017
- Introduction to Lisp-Stat **[16:44]**
    - by Balasubramanian Narasimhan
    - /article/1064
- *LJ* Interviews Linus Torvalds **[29:13]**
    - by Phil Hughes and Gena Shurtleff
    - /article/0146
- *LJ* Readers' Choice **[20:60]**
    - by *LJ* Staff
    - /article/2870
- Languages: Introducing Scheme **[11:23]**
    - by Robert Sanders
    - /article/2879
- LaTex for the Slightly Timid **[18:34]**
    - by Kim Johnson
    - /article/1163
- Learning C++ with Linux **[6:24]**
    - by Jeff Tranter
    - /article/2819
- LessTif and the Hungry ViewKit **[19:56]**
    - by Malcolm Murphy
    - /article/1123

- Leviathan: A Linux-Based Internet Information Server **[12:28]**
    - by Paul Sittler
    - /article/1048
- Licenses and Copyright **[29:34]**
    - by Michael K. Johnson
    - /article/1297
- Linus Torvalds in Sydney **[8:60]**
    - by Jamie Honan
    - /article/2867
- Linux Code Freeze **[1:1]**
    - by Linus Torvalds
    - /article/2733
- Linux Command Line Parameters **[8:35]**
    - by Jeff Tranter
    - /article/2866
- Linux Conference at Open Systems World/FedUNIX'94 **[10:20]**
    - by Belinda Frazier
    - /article/0048
- Linux Distributions Compared **[23:24]**
    - by *LJ* Staff
    - /article/1230
- Linux Distributions Compared **[29:20]**
    - by Bryan Phillippe and *LJ* Staff
    - /article/0139
- Linux Does Comics **[4:26]**
    - by Robert Suckling
    - /article/2790
- Linux Goes 3D: An Introduction to Mesa/OpenGL **[31:21]**
    - by Jorg Rudiger-Hill
    - /article/0174

- Linux Goes to Sea **[16:19]**
    - by Randolph Bentson
    - /article/1111
- Linux Performance Tuning for the Faint of Heart **[7:26]**
    - by Clarence Smith
    - /article/2859
- Linux in Amsterdam **[11:6]**
    - by Michael K. Johnson
    - /article/1026
- Linux in the Real World: Redesigning SCADA at Virginia Power **[9:23]**
    - by Vance Petree
    - /article/1005
- Linux in the Real World: SCADA-Linux Still Hard at Work **[10:14]**
    - by Vince Petree
    - /article/1015
- Linux in the Trenches: The Roger Maris Cancer Center—Depending on Linux **[5:21]**
    - by G. W. Wettstein
    - /article/2804
- Linux on Mobile Computers **[26:19]**
    - by Kenneth E. Harker
    - /article/1260
- Linux—It's Not Just for Intel Anymore **[12:20]**
    - by Joseph Brothers
    - /article/0063
- Linux Programming Hints: A Method for Building Shared Libraries **[12:9]**
    - by Eric Kasten
    - /article/1052
- Linux vs. Windows NT and OS/2 **[1:1]**
    - by Bernie Thompson
    - /article/2734

- Majordomo: A Mailing List Handler **[13:13]**
    - by Piers Cawley
    - /article/1067
- Making the Most of Andrew **[7:7]**
    - by Terry Gliedt
    - /article/2842
- Messages, A Multi-Media Mailer **[6:7]**
    - by Terry Gliedt
    - /article/2817
- Mobile Computing with Linux **[6:17]**
    - by Mark Fiuczynski
    - /article/2818
- Money Management in Linux **[26:10]**
    - by Curt Olson
    - /article/1265
- OpenGL Programming on Linux **[31:87]**
    - by Vincent Cojot
    - /article/5534
- Optimizing the Linux User Interface **[19:47]**
    - by Jeff Arnholt
    - /article/1091
- Porting DOS Applications to Linux **[17:28]**
    - by Alan Cox
    - /article/1138
- Qt GUI Toolkit **[31:32]**
    - by Eirik Eng
    - /article/0201
- Remote Network Commands **[9:33]**
    - by Jens Hartmann
    - /article/1002

- Report from the Front: Linux in Antarctica **[7:11]**
    - by Andrew Tridgell
    - /article/2828
- Review: Intelligent Multiport Serial Boards **[14:46]**
    - by Greg Hankins
    - /article/1097
- Review of Scilab: A Math and Graphics Package for Linux **[11:40]**
    - by Robert Dalrymple
    - /article/1101
- Review: xBase Products for Linux **[14:22]**
    - by Robert Broughton
    - /article/1083
- Samba: Unix Talking with PCs **[7:22]**
    - by Andrew Tridgell
    - /article/2858
- Scheduled Activity: cron and at **[26:48]**
    - by John Raithel
    - /article/1189
- Sendmail+IDA **[3:30]**
    - by Vince Skahan
    - /article/2771
- Setting up X11 **[15:24]**
    - by Greg Lehey
    - /article/1089
- Tar and Taper for Linux **[22:19]**
    - by Yusuf Nagree
    - /article/1216
- That First Gulp of Java **[30:17]**
    - by Brian Christeson, John D. Mitchell, and Todd Graham Lewis
    - /article/0148

- The Humble Beginnings of Linux **[11:11]**
  - by Randolph Bentson
  - /article/0041
- The Joy (and Agony) of SLIP **[6:30]**
  - by Warren Baird
  - /article/2820
- The LINCKS GPD **[15:17]**
  - by Martin Sjolin, by David G. Korn, Charles J. Northrup and Jeffery Korn
  - /article/1072
- The Pari Package on Linux: Solving Mathematics with Linux **[13:5]**
  - by Klaus-Peter Nischke
  - /article/1068
- Tutorial: Emacs for Programmers **[6:39]**
  - by Matt Welsh
  - /article/2821
- Understanding Red Hat Run Levels **[27:21]**
  - by Mark F. Komarinski
  - /article/1274
- Using Linux at SSC/*Linux Journal* **[20:27]**
  - by Kevin Pierce
  - /article/0086
- Using SmartWare Plus to Build the Integrated Office **[26:25]**
  - by Phil Hughes
  - /article/0115
- Using Tcl and Tk from Your C Programs **[10:26]**
  - by Matt Welsh
  - /article/0044
- Using TeX with xtem **[29:46]**
  - by Evelyn Mitchell
  - /article/1171

- Using grep **[18:42]**
    - by Eric Goebelbecker
    - /article/1149
- What is Java, Really? **[30:50]**
    - by Rudi Cilibrasi
    - /article/0171
- What is a Linux? **[8:59]**
    - by *LJ* Staff
    - /article/2735
- Wine **[4:14]**
    - by Bob Amstadt and Michael K. Johnson
    - /article/2788
- World Wide Web **[3:9]**
    - by Bernie Thompson
    - /article/2769
- Writing a Mouse-Sensitive Application **[17:14]**
    - by Alessandro Rubini
    - /article/1136
- Writing Manual Pages Using groff **[18:18]**
    - by Matt Welsh
    - /article/1158
- XFree86 Announces Formation of the XFree86 Project, Inc. **[2:1]**
    - by *LJ* Staff
    - /article/2749
- X Window System Programming with Tcl and Tk **[8:24]**
    - by Matt Welsh
    - /article/0008
- ncurses: Portable Screen Handling for Linux **[17:43]**
    - by Eric Raymond
    - /article/1124

- xfm 1.3: A File and Applications Manager **[15:50]**
    - by Robert Dalrymple
    - /article/1074

### From the Editor

- Linux 2000 **[2:2]**
    - by Phil Hughes
    - /article/2759
- Metamorphoses **[3:2]**
    - by Michael K. Johnson
    - /article/2777
- Unix Wars, Redux **[26:8]**
    - by Michael K. Johnson
    - /article/1272
- What is Success? **[21:10]**
    - by Michael K. Johnson
    - /article/0089

### From the Publisher

- Changes at *LJ* **[29:8]**
    - by Phil Hughes
    - /article/2198
- Freely Redistributable Software is Alive and Well **[25:10]**
    - by Phil Hughes
    - /article/0112
- Is This Any Way to Run a Railroad? **[31:8]**
    - by Phil Hughes
    - /article/0214
- The Politics of Freedom **[30:8]**
    - by Phil Hughes
    - /article/0147

- *WEBsmith* **[21:7]**
    - by Phil Hughes
    - /article/5544

## Kernel Korner

- An Introduction to Block Device Drivers **[9:11]**
    - by Michael K. Johnson
    - /article/2890
- Block Device Drivers: Optimization **[11:38]**
    - by Michael K. Johnson
    - /article/1013
- Block Device Drivers: Interrupts **[10:9]**
    - by Michael K. Johnson
    - /article/2885
- Device Drivers Concluded **[28:19]**
    - by Georg von Zezschwitz
    - /article/1287
- Dissecting Interrupts and Browsing DMA **[26:29]**
    - by Alessandro Rubini
    - /article/1222
- Dynamic Kernels: Modularized Device Drivers **[23:10]**
    - by Alessandro Rubini
    - /article/1219
- Dynamic Kernels: Discovery **[24:18]**
    - by Alessandro Rubini
    - /article/1220
- Introduction (Device Drivers) **[8:18]**
    - by Michael K. Johnson
    - /article/2890
- Linux on Alpha AXP **[21:38]**
    - by David A. Rusling
    - /article/1202

- Memory Allocation **[16:16]**
    - by Michael K. Johnson
    - /article/1133
- Network Buffers and Memory Management **[30:20]**
    - by Alan Cox
    - /article/1312
- Porting Linux to the DEC Alpha: The Kernel and Shell **[19:16]**
    - by Jim Paradis
    - /article/1178
- Porting Linux to the DEC Alpha: Infrastructure **[18:22]**
    - by Jim Paradis
    - /article/1044
- Porting Linux to the DEC Alpha: The User Environment **[20:13]**
    - by Jim Paradis
    - /article/1177
- System Calls **[17:12]**
    - by Michael K. Johnson
    - /article/1145
- The ELF Object File Format: Introduction **[12:14]**
    - by Eric Youngdale
    - /article/1059
- The ELF Object File Format by Dissection **[13:27]**
    - by Eric Youngdale
    - /article/1060
- The Linux Keyboard Driver **[14:5]**
    - by Andries Brouwer
    - /article/1080

## Linux Means Business

- MkLinux—Linux Comes to the Power Mac **[31:55]**
    - by Richard Kinne
    - /article/0203

- Sticking with Progress ... **[29:17]**
    - by Peter Struijk and Lydia Kinata
    - /article/1283
- Using Sendmail as a Multi-Platform Mail Router **[30:34]**
    - by Tom Lowery
    - /article/1294

### Linux System Administration

- Adding a New Disk to a Linux System **[20:52]**
    - by Æleen Frisch
    - /article/1169
- Anonymous FTP **[13:41]**
    - by Mark Komarinski
    - /article/1069
- Fixing Your Clock **[8:15]**
    - by Mark Komarinski
    - /article/0012
- How to Log Friends and Influence People **[11:35]**
    - by Mark Komarinski
    - /article/2926
- How to Move /home to a New Hard Drive on Your Linux System **[8:44]**
    - by *LJ* Staff
- Installing the Xaw3D Libraries **[15:54]**
    - by Mark Komarinski
    - /article/1113
- Linux System Administration **[2:26]**
    - by Mark Komarinski
    - /article/2805
- Making Your Own Filesystems **[3:20]**
    - by Mark Komarinski
    - /article/2779

- Maximizing Linux Security: Part 1 **[21:12]**
    - by Æleen Frisch
    - /article/1170
- Maximizing Linux Security: Part 2 **[22:10]**
    - by Æleen Frisch
    - /article/1206
- mtools **[5:17]**
    - by Mark Komarinski
    - /article/2805
- rm Your Way to Fun and Adventure **[9:14]**
    - by Mark Komarinski
    - /article/2787
- Setting up Services **[12:18]**
    - by Mark Komarinski
    - /article/1274
- The df Command **[1:35]**
    - by Phil Hughes
    - /article/2747
- Upgrading the Linux Kernel **[14:30]**
    - by Mark Komarinski
    - /article/1099
- Using LILO, the Linux Loader **[19:52]**
    - by Æleen Frisch
    - /article/1166

## News and Articles

- A New Project or a GNU Project? **[13:44]**
    - by Mark Bolzern
    - /article/1075
- An Introduction to Python **[21:21]**
    - by Jeff Bauer
    - /article/1121

- An Introduction to Rlab **[26:56]**
    - by Ian Searle
    - /article/1130
- Andy **[7:40]**
    - by Andy Tefft
    - /article/2864
- BBS **[12:54]**
    - by *LJ* Staff
    - /article/2789
- Basic FVWM Configuration, Part 1: Getting (fvwm) Started **[27:50]**
    - by John M. Fisk
    - /article/1198
- Basic RVWm Configuration, Part 2: Customizing the Desktop **[28:54]**
    - by John M. Fisk
    - /article/1236
- CD-ROMs and Linux **[7:33]**
    - by Jeff Tranter
    - /article/2851
- CVS: Version Control: Beyond RCS **[21:43]**
    - by Tom Morse
    - /article/1118
- Caldera and Corsair **[14:18]**
    - by *LJ* Staff
    - /article/1094
- Certifying Linux
    - by Heiko Eissfeldt
    - /article/0131
- Creating a Linux Firewall Using the TIS Firewall Toolkit **[25:36]**
    - by Benjamin Ewy
    - /article/1204

- DEC AXP Review **[30:53]**
    - by Bryan Phillippe
    - http:www.dcginc.com
    - /article/0184
- Dialog: An Introductory Tutorial **[5:24]**
    - by Jeff Tranter
    - /article/2807
- Ext2tools: Reading Linux Files from DOS **[23:43]**
    - by Robert Dalrymple
    - /article/1209
- Finding Files and More **[20:40]**
    - by Eric Goebelbecker
    - /article/1180
- Finding Linux Software **[24:46]**
    - by Erik Troan
    - /article/1098
- Fix /etc/gateway **[6:21]**
    - by Cor Bosman
    - /article/2826
- Geomview **[23:18]**
    - by Tim Jones
    - /article/2826
- Getting to Know gdb **[29:10]**
    - by Mike Loukides and Andy Oram
    - /article/1298
- Graphing with Gnuplot and Xmgr **[28:12]**
    - by Andy Vaught
    - /article/1218
- Hamming It Up on Linux: Using Ham Radios with Linux **[13:26]**
    - by Brian Lantz
    - /article/1071

- Harbor **[6:46]**
  - by Michael K. Johnson
  - /article/2832
- How to Build A Mac: The Executor Macintosh Emulator **[19:40]**
  - by Andreas Schiffler and David Moody
  - /article/1031
- How to Read 950 E-mail Messages Before Lunch **[22:42]**
  - by Jay D. Allen
  - /article/1196
- ICMAKE Part 1 **[1:12]**
  - by Frank Brokken and Karel Kubat
  - /article/2738
- ICMAKE Part 2 **[2:34]**
  - by Frank Brokken and Karel Kubat
  - /article/2767
- ICMAKE Part 3 **[3:24]**
  - by Frank Brokken and Karel Kubat
  - /article/2776
- ICMAKE Part 4 **[4:35]**
  - by Frank Brokken and Karel Kubat
  - /article/2794
- ILUG Shows Off **[29:38]**
  - by Shay Rojansky
  - /article/1314
- Indexing with Glimpse **[18:16]**
  - by Michael K. Johnson
  - /article/1164
- Installing Linux via NFS **[11:15]**
  - by Greg Hankins
  - /article/1021

- Introducing HyperNews **[27:58]**
    - by David Alan Black
    - /article/1281
- Introduction to the GNU C Library **[2:18]**
    - by Michael K. Johnson
    - /article/2754
- Java and Postgres95 **[31:77]**
    - by Bill Binko
    - /article/5535
- Let's Take Linux Seriously **[3:7]**
    - by Phil Hughes
    - /article/2774
- Linus Announces Code Freeze in Preparation for 1.2.x **[6:37]**
    - by Linus Torvalds
    - /article/2682
- Linus Torvalds Receives Award **[13:60]**
    - by *LJ* Staff
    - /article/2921
- Linus Torvalds at DECUS '94 **[4:20]**
    - by Bob Tadlock
    - /article/2791
- Linux Development Grant Fund **[9:42]**
    - by *LJ* Staff
    - /article/0037
- Linux Distributions **[2:22]**
    - by *LJ* Staff
    - /article/2755
- Linux Events: Two Views on Heidelberg **[5:32]**
    - by *LJ* Staff
    - /article/2810

- Linux at SCO Forum **[19:59]**
    - by Belinda Frazier
    - /article/2608
- Linux at the UW Computer Fair **[14:17]**
    - by *LJ* Staff
    - /article/0065
- Linux in the Real World—Linux in the U.S. Army, or How I Spent My Winter Vacation **[26:16]**
    - by Joe Klemmer
    - /article/1259
- Linux in the Real World—Remote Data Gathering with Linux **[11:18]**
    - by Grant Edwards
    - /article/2880
- Linux on Alpha: A Strategic Choice **[18:29]**
    - by Jon “maddog” Hall
    - /article/1150
- Linux on Low-End Hardware **[19:38]**
    - by Trenton B. Tuggle
    - /article/1144
- Linux on the Motorola 680x0 **[5:20]**
    - by Hamish Macdonald
    - /article/2806
- *LJ* Interviews Larry Gritz **[31:30]**
    - by Amy Wood
    - /article/0221
- Maceater: A True Story **[27:20]**
    - by Jonathan Gross
    - /article/1270
- Maximizing Linux Security: Part 1 **[21:12]**
    - by Æleen Frisch
    - /article/1170

- Maximizing Linux Security: Part 2 **[21:12]**
    - by Æleen Frisch
    - /article/1206
- Mobile-IP: Transparent Host Migration on the Internet **[28:10]**
    - by Ben Lancki, Abhijit Dixit and Vipul Gupta
    - /article/1271
- Mr. Torvalds Goes to Washington **[12:5]**
    - by Kurt Reisler
    - /article/1062
- My Next Pentium is a DEC Alpha **[30:12]**
    - by Bryan W. Headley
    - /article/0170
- Netsurfing With Linux: To Sail the Cyber Sea **[13:30]**
    - by Arthur Bebak
    - /article/1050
- Object Databases: Not Just for CAD/CAM Anymore **[27:39]**
    - by Gregory A. Meinke
    - /article/1234
- Onyx **[1:14]**
    - by Michael Kraehe
    - /article/2737
- Orest Zborowski **[15:14]**
    - by Phil Hughes
    - /article/0070
- Overview of the Debian GNU/Linux System **[6:59]**
    - by Ian Murdock
    - /article/2841
- Pentiums & Non-Pentiums **[11:34]**
    - by Phil Hughes
    - /article/0058

- Prototyping Algorithms in Perl **[16:39]**
    - by Jim Shapiro
    - /article/1103
- Putting Widgets in Their Place **[16:30]**
    - by Stephen Uhler
    - /article/1117
- Questions from the *Linux Journal* Booth at Open Systems World **[11:20]**
    - by Kim Johnson
    - /article/2881
- Reader Survey Results **[17:36]**
    - by *LJ* Staff
    - /article/2878
- Report from the Front: The Linux Review Group **[6:23]**
    - by Magnus Alvestad
    - /article/2828
- Report on Comdex '94 **[10:6]**
    - by Belinda Frazier and Phil Hughes
    - /article/0045
- Serving Two Masters **[27:45]**
    - by Michael K. Johnson
    - /article/1275
- Slackware 2.0 Released **[4:45]**
    - by Phil Hughes
    - /article/2795
- The Best Without X **[19:22]**
    - by Alessandro Rubini
    - /article/1090
- The Devil's in the Details **[25:52]**
    - by Georg von Zezschwitz and Alessandro Rubini
    - /article/1221

- The Java Developer's Kit **[31:18]**
    - by Arman Danesh
    - /article/0166
- The Linux FSSTD **[2:28]**
    - by Daniel Quinlan
    - /article/2756
- The Linux File System Standard **[15:45]**
    - by Garrett D'Amore
    - /article/1104
- The Linux-GGI Project **[31:60]**
    - by Steffen Seeger and Andreas Beck
    - /article/0160
- The Open Development of Debian **[3:29]**
    - by Ian Murdock
    - /article/2782
- The Quintessential Linux Benchmark **[21:51]**
    - by Wim van Dorst
    - /article/1120
- The Rough and Tumble World of the Linux-Based ISP **[25:22]**
    - by Todd Graham Lewis
    - /article/1240
- The Term Protocol **[8:39]**
    - by Liem Bahneman
    - /article/0010
- The Trade Shows: Comdex and Internet World **[16:14]**
    - by Randolph Bentson and Arnold Robbins
    - /article/1128
- The Trade Shows: Open Systems World/FedUNIX '95 and Comdex '95 **[22:63]**
    - by Kevin Pierce and *LJ* Staff
    - /article/0094

    - /article/2808

- X Forms: Review and Tutorial **[22:50]**
    - by Karel Kubat
    - /article/1188

- XF-Mail **[24:12]**
    - by John M. Fisk
    - /article/1199

## Novice to Novice

- DOS Emulation with DOSEMU **[13:34]**
    - by Dean Oisboid
    - /article/1070

- Databases **[17:37]**
    - by Dean Oisboid
    - /article/1146

- Games, Sound & Other Agonies **[15:6]**
    - by Dean Oisboid
    - /article/1086

- Interlude and Explorations: Spreadsheets and Text Editors **[16:6]**
    - by Dean Oisboid
    - /article/1132

- Keyboards, Consoles and VT Cruising **[31:12]**
    - by John M. Fisk
    - /article/0187

- Linux Installation and X-Windows **[12:26]**
    - by Dean Oisboid
    - /article/1057

- Serendipity **[18:26]**
    - by Dean Oisboid
    - /article/1165

## Practical Programming Hints

- Debugging Tcl Scripts, or Real Programmers Use puts **[18:12]**
  - by Stephen Uhler
  - /article/1159
- It's All a Matter of Timing **[20:18]**
  - by Stephen Uhler
  - /article/1186

## Product Reviews

- AX Graphical Display Server **[15:31]**
  - by Mark Ganter
  - /article/0031
- BRU—Backup and Restore Utility **[11:43]**
  - by Jon Freivald
  - /article/2884
- Caldera Network Desktop Preview 1 **[20:34]**
  - by Roger Scrafford
  - /article/1179
- Crisp Text Editor **[6:27]**
  - by Robert Broughton
  - /article/2829
- Doom **[8:22]**
  - by Michael K. Johnson
  - /article/0001
- IGEL Etherminal 3X **[19:35]**
  - by Michael K. Johnson
  - /article/1167
- InfoMagic Linux Developer's Resource **[12:25]**
  - by Caleb Epstein
  - /article/0060
- Metro X **[15:30]**
  - by Bogdan Urma
  - /article/1073

- Moo-Tiff Development Environment **[17:55]**
  - by Bogdan Urma
  - /article/1142
- Motif 1.2.3 Runtime and Development System for Linux **[6:12]**
  - by Dale A. Lutz
  - /article/2822
- Motif for Linux **[15:48]**
  - by Bogdan Urma
  - /article/1106
- SlickEdit **[14:41]**
  - by Jeff Bauer
  - /article/1095
- Unix Interactive Tools **[6:22]**
  - by Clarence Smith
  - /article/2827
- Xfig **[12:6]**
  - by Robert Dalrymple
  - /article/1043

## Programming Hints

- Introduction to make **[6:48]**
  - by Michael K. Johnson
  - /article/2839
- Programming the VT Interface **[3:35]**
  - by Michael K. Johnson
  - /article/2783
- Programming the VT Interface Part 2 **[4:31]**
  - by Michael K. Johnson
  - /article/2798
- Running Windows Applications NOW **[23:48]**
  - by Ron Bardarson
  - /article/5542

- *Linux Journal* at Unix Expo **[8:5]**
    - by Phil Hughes
    - /article/2869
- Linux/MIPS, Version 0.9.2 of Linux/68k, LSM Maintainer Change **[6:5]**
    - by *LJ* Staff
- Linux Version 2.0 **[28:8]**
    - by Michael K. Johnson
    - /article/1311
- Linux at DECUS **[15:10]**
    - by Michael K. Johnson
    - /article/2677
- Porting Linux to Other Platforms **[27:8]**
    - by Michael K. Johnson
- Recent Developments **[25:8]**
    - by Michael K. K. Johnson
    - /article/1263
- Some Breaking News **[3:6]**
    - by Michael K. Johnson
- The Desktop War **[23:8]**
    - by Michael K. Johnson
    - /article/1232
- Time Again for Reader Input **[11:10]**
    - by *LJ* Staff
    - /article/0069
- WYSIWYG Editor, BLADE, Linux at OSW **[17:10]**
    - by *LJ* Staff
- Windows 95 and Linux **[18:11]**
    - by Phil Hughes

## Take Command

- Keeping Track of Change with RCS **[25:63]**
    - by Michael K. Johnson
- The chmod Command **[21:27]**
    - by Eric Goebelbecker
    - /article/1190
- The rm Command **[11:46]**
    - by Phil Hughes
    - /article/0050
- apropos, whatis and makewhatis **[30:54]**
    - by David Bandel
    - /article/1329
- cpio: A Multipurpose Archive Tool **[23:44]**
    - by Eric Goebelbecker
    - /article/1213
- dtree **[29:61]**
    - by Phil Hughes
    - /article/0150
- etags **[31:66]**
    - by Dave Thomas
    - /article/0153

## What's GNU?

- Bash—the GNU Shell (Part 1) **[3:40]**
    - by Chet Ramey
    - /article/2784
- Bash—the GNU Shell (Part 2) **[4:41]**
    - by Chet Ramey
    - /article/2800
- Plan 9 **[11:31]**
    - by Arnold Robbins
    - /article/1012

- RCS: Revision Control System **[10:36]**
  - by Arnold Robbins
  - /article/2888
- Texinfo **[6:51]**
  - by Arnold Robbins
  - /article/0062
- The GNU Coding Standards **[16:47]**
  - by Arnold Robbins
  - /article/1135
- What's GNU? **[1:20]**
  - by Arnold Robbins
  - /article/2862
- What's GNU? **[2:14]**
  - by Arnold Robbins
  - /article/2888
- groff **[7:19]**
  - by Arnold Robbins
  - /article/2741

Archive Index Issue Table of Contents

Advanced search

# 1996 Readers' Choice Awards

**Various**

Issue #32, December 1996

*Linux Journal* Readers pick their favorite hardware and software.

Welcome to *Linux Journal*'s second annual Readers' Choice Awards.

In response to requests for more choices, last year's three categories have been expanded to ten. We've added several categories, including the hotly contested “Favorite CD-ROM Distribution”.

Our survey was conducted through the *Linux Journal* web site, where the voting was open for two weeks. The survey, though unscientific, gives us an idea of what our readers are using, what they like and what they'd like to see more of in *Linux Journal*.

Now, the results...

Favorite CD-ROM Distribution

Winner: Official Red Hat Linux

“Red Hat is second to none in tech support. I can usually expect e-mail response to a question within five minutes. Try getting that level of service from Microsoft for free!” --Chuck Milam

—Chuck Milam

Runner-up: Debian Linux

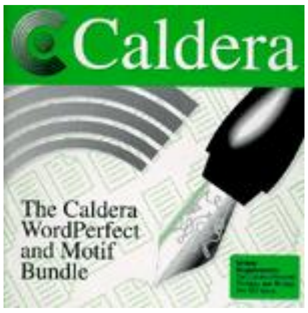
Favorite Text Editor

Winner: Emacs

Runner-up: vi/elvis

Favorite Word Processor or Text Processor

Winner: WordPerfect 6.0 for Linux

> 
>
> “Caldera has the RIGHT stuff.” --Ric Moore
>
> —Ric Moore

Runner-up: groff

Favorite Game

Winner: DOOM

> DOOM AND ULTIMATE DOOM
> “Are there any real games for Linux apart from DOOM?” --Sharon
>
> —Sharon

Runner-up: Quake

Favorite Development Tool

Winner: gcc

> “For me it is difficult to decide if I prefer gcc or Perl5.”
>
> —Jan “Yenya” Kasprzak

Runner-up: G++

Favorite Serial Interface Board

Winner: Cyclades

Runner-up: Comtrol Rocketport

Favorite Platform

Winner: Intel

> “I selected Intel as my favourite platform...Alpha and PPC chips have taken far too long to come out on inexpensive, reliable clone motherboards rather than expensive name brand stuff. I can't wait for the day when superior chips are as inexpensively ubiquitous as the x86 chips.”
>
> —Craig Sanders

Runner-up: Alpha

Favorite New Linux Book

**Winner:** *Linux System Administration Guide* by Lars Wirzenius Available at sunsite.uc.edu/mdw/linux.html#ldp

Runner-up: *Inside Linux* by Randolph Bentson Published by SSC

Favorite Overall Linux Book

**Winner:** *Linux Network Administrators Guide* by Olaf Kirch Published by SSC
Runner-up: *Running Linux* by Matt Welsh and Lar Kaufman Published by O'Reilly & Associates, Inc

More information about the products included in this article (or other Linux hardware and software) is available through www.ssc.com/linux/.

Archive Index Issue Table of Contents

Advanced search

# Pagesat High Speed News

**Rich Myers**

Issue #32, December 1996

Ready to get your news the high-tech way—with a satellite dish? Here's the scoop.

This article is a discussion of several aspects of my company's news system, namely the principles of operation and the hardware and software utilized, in order to guide the novice news seeker through a successful implementation of Pagesat's High Speed News feed. Our “wish” list for a news system included:

- A low-cost solution that would not impact local network traffic or chew up our Internet bandwidth
- A system that could be monitored and modified remotely
- A system that could support more than just a few simultaneous news readers
- A system that worked *now*

The news system that we have now set up is comprised of four major components:

- Pagesat High Speed News antennae and receiver
- News “receiver” machine connected to high-speed receiver, Ethernet and SLIP attached
- Master news machine, Ethernet attached
- Slave news machine, Ethernet attached

## Software

All three news machines run Slackware Linux 1.2.8 as the operating system. INN 1.4 processes the news feed. We are currently using the “hsdist2.0b.tar” software to decode data from the receiver. This software contains “Forward

Error Correction" code, which eliminates or drastically reduces data loss caused by less-than-perfect satellite reception.

### Hardware

We have three Intel-based machines. The receiver machine is a 486-33, 8MB RAM, 500M IDE HD. The master news machine is a little beefier: a P133 with 32MB RAM, 1GB SCSI for the OS, 4GB fast SCSI-II to retain the news hierarchy and data files, and another 4GB fast SCSI-II to contain INN and other toys. The relationship of news and the disk subsystem is simple—both should be big and fast. At the current rate, starting from scratch, my 4GB disk is full of news in 5.5 days. We use the Buslogic BT-456-C PCI SCSI controller because it is twice as fast as the 16-bit Adaptec 1542C. Having the Buslogic helps in particular on the daily expire, which went from 3+ hours to a mere 35 minutes. All the machines are Ethernet-attached, and one of the machines has a modem for remote control. The slave machine is just that—a slave. It's identical to the master in hardware configuration, and just receives everything that the master "feeds" it, which is everything. Why do we have it? In a pinch it can be configured as the "master", just in case some disaster strikes the master, and it can also be used as a primary or secondary news reader machine for all our clients.

### Principles of Operation

The number one rule is to keep operation straightforward and simple. Cron-managed batch jobs were our choice: I can't write C code, but I can write simple shell scripts. I wanted a little more monitoring capability, so I added extra processing to the Pagesat software. We accumulate news for half-hour bursts, then process it into the news system on the hour and the half hour. It currently takes, on average, fifteen minutes to process the previous half hour's data. At 15 and 45 minutes past the hour, we "feed" the slave, sending everything we have received to that point. We run a nightly expire on the master and slave to get rid of the old news and prepare for the next day. The "receiver" machine runs both the PSFRX and PSNEWS programs to receive the data and process it into the .gz data files. These files are stored on an NFS-mounted disk r/w to the master. The master copies the files at the specified intervals onto its disks and deletes them from the receiver disk. The master then processes the news into the system. With this configuration we can take down the master machine for whatever reason and continue to accumulate news on the receiver, processing it whenever the master comes on-line again.

### Why Does This Make Sense?

Three reasons: it's dirt cheap, it's efficient, it works. Add it up. The total cost for software is ZERO. Hardware costs are minimal, because PCs are a lot less expensive than workstations, and disk drive prices keep dropping every day. And RAM is really cheap these days—16MB for $100 or less.

## Silly Alternatives

Have you priced a leased line lately? The cost is maybe $200 a month for a 64KB circuit to pipe your news to you, a circuit that can't even support a full feed. Want to slurp two or more days of old news from your provider across your 28.8 AND try to surf AND do anything else at the same time?

## Ready to Install the Software

If you don't already have your Linux system(s) up and running, do it now. Run down to your local software house and pick up a CD-ROM containing the latest version of Linux (I like Slackware), follow the instructions and get it installed. It's easy—all it takes is a little time. Take the extra time needed to customize your kernel in order to save RAM. Next, get X-Windows up and running, so that you can monitor several things simultaneously. Make sure your TCP/IP is working, be it LAN or SLIP/PPP, to allow posting capability. Now you're ready to set up the news system. We chose to obtain a source code version of INN off the Internet rather than use the distributed version. Key files worth reading are most notably the FAQs in /usr/lib/news/tools.linux, and the README files in the base directory. These files explore the configuration options and operating procedures.

Now it's time to build your news repository. First, fetch the latest “active” file from ftp.pagesat.net. Then write a simple script to strip out and retain the newsgroup name, and append “00000000 000000001 y” to each entry to reset the news article counters. Make your modified file the “active” file. Now run /usr/lib/news/bin/makehistory and watch a lot of your disk space be consumed by the directory structure being built to house the news data. Next, you will need to edit some of the INN control files in /usr/lib/news. The following examples are excerpts from our working files, with explanations. Feel free to copy and/or modify to suit your configurations.

## File 1: control.ctl

```
## mail notification to root for all control
## functions, and create new newsgroups.
all:*:*:mail
checkgroups:*:*:mail
ihave:*:*:mail
sendme:*:*:mail
newgroup:*:*:doit=mail
rmgroup:*:*:mail
sendsys:*:*:mail
senduuname:*:*:mail
version:*:*:mail
```

### File 2: expire.ctl

```
## expire control and junk after 1 day, keep
## 2 newsgroups for 90 days, keep biz.pagesat
## forever, expire all other news after 3 days.
/remember/:1
control:A:1:1:1
junk:A:1:1:1
*:A:3:3:3
news.software.nntp:A:90:90:90
comp.os.linux*:A:90:90:90
biz.pagesat:A:never:never:never
```

### File 3: hosts.nntp

```
newsfeed.webworks.net:
localhost.webworks.net:
```

### File 4: inn.conf

```
## our org, server and domain... please use your own.
organization:   Webworks Internet Services
server:         newsfeed.webworks.net
domain:         webworks.net
```

### File 5: newsfeeds

```
##  feed this machine and slave everything., output
##  posts to slave and pagesat.
## exclude some posting from pagesat
ME:*::
slave:*:Tf,Wnm:
pagesat/jolt.pagesat.net,news.pagesat.net,pagesat.net,\
   pubxfer.news.psi.net,psinntp,unknowna:*,\
   !junk*,!local*,!control*:Tf,Wnm:
```

### File 6: nnrp.access

```
## allow/disallow  newsreader/nntp acess<\n>
*:: -no- : -no- :!*
*.webworks.net:Read Post:::*
```

### File 7: nntpsend.ctl

```
## the FQDN of all the machine names that we intend to feed
slave:slave.webworks.net:1m:-t300
pagesat:news.pagesat.net:1m:-t300
```

### Ready to Install the Pagesat Antenna

Once again, the main thing to remember is to follow the directions. Read the documentation that comes with the dish and receiver. Grab a compass and protractor, an extension cord and the tools necessary to assemble the dish. Don't forget a beer or two, a lawn chair and a friend with two hands to help. Go out into your yard and plug everything together. Then, using your compass and protractor, aim the dish in the general direction of the satellite to obtain a tone signal. This tone will help you orient the dish to the proper location, so that you can decide where to mount it permanently—a position that should be free of current and future obstructions. Once you get the antenna mounted, attach it to your computer, and start up PSFRX -v to see if you're pointing at the correct satellite. If you are, you should see a series of dots representing data blocks—it isn't a continuous flow, so be patient. If you see other characters like C and S, which represent errors, try re-aiming the dish a little, twisting the LNB for proper polarization. You really need a friend within earshot to fine-tune the aiming of the dish. If you're getting data, you're aimed at the right place. Now you can re-attach the receiver next to the dish for fine-tuning. Using the tone and meter, you can really zero in on the satellite. Once done, go back into the house, and re-attach the receiver to your PC: you're ready to start receiving the news!

## Feeding Time

Start up INN with /usr/lib/news/etc/rc.news and the PSFRX and PSNEWS programs. You should start seeing news batches filling up your spool queue. Set **cron** to run the PROCESSSATNEWS PROGRAM each half hour or so. Pick a time and set **cron** to run **news.daily** once a day. Watch your disk fill up with hundreds of megabytes of news daily. Now you're a member of the INN crowd; pick your favorite newsgroups (e.g., biz.pagesat, comp.os.linux.* and news.*) and start learning more about how to tailor your system to be exactly as you want it.

**Rich Myers** (rich@webworks.com) cut his programming teeth on IBM iron using 370 Assembler language back in 1980, when mainframes were “in”. In the last half of the 80s when the dinosaurs were starting to die off and PCs were sprouting everywhere, he was manager of a network of SUN workstations. Then came Corporate Takeovers, with countless changes to the almighty LANs and WANs, and late nights and weekends keeping up with the Joneses. And throughout all this, he bought not even one winning lottery ticket. Which brings us to now—a piece of Internet Pie for me, please.

# V—A Free C++ GUI Framework for X

PhD. Bruce E. Wampler,

Issue #32, December 1996

Dr. Wampler introdces us to V, an excellent GUI framework for writing applications that work with Linux and X Windows.

I love Linux as a development platform—it has all the Unix tools I've been using for years. Linux has been a superior development platform in every way except for development of graphical user interfaces (GUIs) with X. For some time, programmers using Microsoft Windows have had the option of using either Microsoft's MFC or Borland's OWL to develop C++ GUI applications. However, Linux has caught up in the graphics area as some very good freeware C++ GUI frameworks are now available for X and Linux. This article is about V, the GUI framework I've written.

I first developed V as a tutorial example for a C++ book I'm writing. My tutorial has turned into a very complete, easy-to-use C++ GUI framework suitable for developing almost any GUI application. It works with X and Linux, and uses the standard X Athena toolkit, so that everything remains free. V uses some customized versions of the free Athena 3D widget set, so V applications have a very handsome 3D look. V also works with Microsoft Windows, so applications you build on Linux with V are easily ported to MS-Windows, as well as other X platforms. V is distributed under the GNU Library General Public License, so you can freely use V to develop either other freeware or proprietary applications (as long as you make the source of V available to the end user).

Two other good frameworks are wxWindows and YACL. While these two programs have their strong points, I think V is much simpler to use and has applications that are more user friendly. For more details, a comparison of V and the other frameworks is posted on my web page (http://www.cs.unm.edu/~wampler/). All three programs provide much of the functionality of commercial frameworks such as MFC or OWL.

## Life Without a Framework

If you don't use a GUI library or framework, you are stuck using the native X toolkits to develop your GUI, either by using X directly (very hard) or using the Xt Athena or Motif widget sets (just a bit easier). While you can develop very nice interfaces using these toolkits directly, there is a steep learning curve. A good programmer needs at least a month of steady effort to be able to write programs using either Xt Athena or Motif, and the applications are limited to X platforms. One of the problems with the native X toolkits is that they are designed to be totally flexible, and to handle almost any interface design. This very generality makes them hard to learn and to use. For most applications, you simply don't need that much flexibility.

V, on the other hand, has been designed from the start with ease of programming in mind. I have received reports from new V users saying they were able to get their first V applications up and running in just four or five hours from downloading the V distribution to ending with a running application. Within a week, programmers are able to fully use almost every feature of V. Even the inexperienced programmers in my classes at the University of New Mexico have learned to program with V in a week.

The cost for this ease of programming is rather small. For one thing, you must use V's definition of the GUI elements. When you use V, you end up with interfaces that conform closely to standard Motif or Microsoft Windows applications. Also, you lose some of the low level access and flexibility you have when using the low-level toolkits. For the vast majority of applications, this loss is not significant. V does provides hooks for getting at some of the low-level details, but the only use I've heard made of them is to interface with OpenGL and Mesa.

## A Standard V Application

Figure 1 shows the V Icon Editor included with the V distribution. The Icon Editor is representative of a typical V application. The main user interface element is the *command window*. The command window has a menu bar that is the standard drop-down menu found on most Motif and Windows applications. Below the menu bar are one or more tool bars that allow you to make the most commonly used commands readily available to the user. The current trend is to place buttons with small icons on the tool bar; with V you can use any standard control object on the tool bar. Under the tool bar is the drawing canvas where your application shows data, such as text information, graphical information, or, in this case, the icon to edit. The drawing canvas can have scroll bars, and the user interacts with the data displayed on the canvas using the mouse or keyboard. Below the drawing canvas is a status bar, used to show useful information, like the mouse position.

Dialogs are an important part of most GUI programs. Figure 2 shows the color selection modeless dialog used with the V Icon Editor. V supports both modal and modeless dialogs with a complete set of controls.

I hope you agree that the appearance of this V application is as good as any other X application you've seen on Linux. Icon Editor also compiles in Microsoft Windows without a single **#ifdef** and will look just like a standard Windows application.

## V C++ Objects

V is a C++ framework, and therefore, consists of several C++ base classes you can use to derive your own classes. The core class is **vApp**, which is used as the main foundation of the rest of the application. A minimal application has a command window derived from the **vCommandWindow** class containing at least a menu and a drawing canvas. The command window can also contain tool and status bars. When the user selects menu items or activates command objects on the tool bar, V sends messages to your derived **vCommandWindow** object which your program then interprets. Your program usually shows its output on a canvas object, derived from a **vCanvas** class. Mouse input is handled by sending messages to your derived canvas class.

While it is possible to build an application using only the **vApp**, **vCommandWindow**, and **vCanvas** classes, most applications will also use dialogs. The dialog controls supported by V currently include:

- standard push button
- push button with icon and color
- toggle buttons
- labels
- icons
- check box
- radio button
- combo box
- list
- spinners
- slider
- progress bar
- text in
- multi-line text
- frame (box)
- toggle frames (similar to tabs)

The layout of controls in a dialog is determined by specifying a position for each control relative to other controls in the dialog. For example, you might specify one control is to the right and below another control. All dialog controls can also be used on the tool bar of a command window.

V also includes several standard dialogs that implement common operations. One of the most useful is the file open dialog, which lets the user select files and directories interactively. Since no standard X file open dialog exists, V implements its own. The MS-Windows version of V uses the standard Windows file dialog. Other standard dialogs include font selection, yes/no response and a message box. Using standard dialogs helps ensure that your application will match the look and feel of the native platform.

## Building Your First V Application

The first step to using V is obtaining the latest release. The primary source for V is my web page, http://www.cs.unm.edu/~wampler/. V is also available for anonymous ftp from ftp://ftp.cs.unm.edu/pub/wampler. Unpack the distribution, decide where you want to keep the source and libraries, modify the makefile to reflect this decision and execute make.

After you've built V, read the documentation. You can either print it, or use xdvi or ghostscript to view the provided dvi or ps files. You then typically start your own application by modifying one of the examples provided. Most new V users find it very easy to modify the example to get a good start on their own. The entire process of downloading, building, reading the manual and building a first custom V application has been done by several current V users (already familiar with the Internet, Linux and C++) in just four or five hours.

## Summary

Since its release to the world in February 1996, V has been in use by my software engineering class and received considerable interest and positive response from the user community. If you've put off writing a GUI application because X programming is just too hard to learn, V may be the solution you need. If you've been putting off learning C++, again V may be just what you need. V is a very elegant, small and well-designed C++ class library. GUIs are one of the most naturally object-oriented applications around and are a good way to get started with object-oriented programming.

**Bruce E. Wampler, PhD** (wampler@cs.unm.edu) has been involved in the PC industry since its earliest days. In fact, he founded two successful software companies, Aspen Software and Reference Software International, and was the principal designer and author of the well-known grammar checker Grammatik.

He is currently an Adjunct Professor of Computer Science at the University of New Mexico.

Archive Index Issue Table of Contents

Advanced search

# A Brief Introduction to XTide

David Flater

Issue #32, December 1996

This article introduces a program that produces output in text mode, graphics and Java. Besides all that, XTide is both useful and fun.

XTide is free software for predicting tides. If you live on the water, then you're already convinced that it's the coolest program ever written. But if you're stuck inland, like 99% of the human population, you may still find it to be an interesting and possibly educational toy. In fact, I originally wrote it to ease the pain of long hours in a land-locked, windowless office.

XTide can provide tide predictions in a wide variety of graphic and text formats. It can also act as an on-screen tide clock, showing the current time and water level at the location of your choice.

It was developed entirely under Linux, but all flavors of Unix are supported. Ports to Macintosh, DOS, MS-Windows and OS/2 also exist, supported by their respective maintainers.

Since XTide's release in Summer 1995, web sites offering tide predictions have appeared everywhere. It is probably a safe bet that the majority of XTide users have used it only through a web page. Dean Pentcheff's page (http://tbone.biol.sc.edu/tide/) is an example, offering tide predictions for many locations world-wide. However, if you look around, you will find that many coastal cities have their own XTide pages offering local tide information. [One example of this is our own page for Pacific Beach, Washington at http://www.a42.com/grays/PB/. —Pub]

XTide is released under the GNU Public License and comes with no warranty. It is an excellent tool for visualizing tide predictions, but don't ever depend on it for navigation or use it where an error could be dangerous. XTide's predictions are only as good as its database of tide information, and that was built up almost entirely by scavenging the Net. Always verify the output against predictions published by NOAA or a similar agency before taking any risks.

## Installing XTide

The primary site for the XTide source distribution is universe.digex.net/~dave/files/. Only HTTP access is permitted (no anonymous FTP). However, that web page contains full instructions for downloading via HTTP and even provides a program, http_get, to make it easier.

You will need to get at least two files: an XTide source distribution and a harmonics file. The harmonics file is the database of tide information that XTide uses to predict the tides. The default harmonics file, *harmonics*, contains data for many U.S. and British Columbia locations, as well as a fair number of assorted locations world-wide. There is another harmonics file, *harmonics.canadian*, which contains data for over 1,000 Canadian locations.

Generally, there will be two source distributions available: a stable version and a development version. The development version will have “dev” in the file name. XTide follows an open development process, much like the Linux kernel, with every new feature immediately available to those brave enough to run it.

While XTide installation is not fully automatic, there is nothing especially difficult about it. The following is taken from the XTide README, simplified by the assumption that we have a typical Linux system:

- Copy the harmonics file to the place that you have decided to put it (e.g., /etc/harmonics), gunzip it, and chmod 644.
- Edit the #defines in config.h to specify the location and name of the harmonics file and the default location for which to show tides. The location you specify must be in the harmonics file. NOTE: The environment variables LOCATION and HFILE override the values specified in config.h, so you might just want to set these in /etc/profile instead.
- xmkmf; make depend; make.
- Copy the xtide executable into /usr/X11R6/bin.
- Copy xtide.man to /usr/man/man1/xtide.1 and chmod 644.
- Copy xtide.xpm into /usr/include/X11/pixmaps (mode 644) and change your window manager config file to use that icon for windows whose names start with “XTide”. The syntax to do this in fvwm (system.fvwmrc) is:

```
Style "XTide*"      Icon xtide.xpm
```

## Running XTide

XTide is controlled almost entirely by command line switches. Pull-down menus and other user-interface niceties may be added in future versions; thus far it's been a challenge just keeping up with the more urgent feature requests.

XTide has way too many command line switches. Everyone seems to want different things from a tide program, and I have attempted to keep everyone happy. As a result, the default modes are very simple and boring, and you need to supply the switches to turn on any fancy features you may want. The XTide README can help with this; you can also type **xtide -help** to get a listing of the switches.

The default mode for XTide is tide clock, with no fancy options enabled. To get it to look like the picture, you need at least the following switches:

```
  -now    Show current time in tide clock.
    -hinc Label the tide height tick marks with
          default increment.
    -tinc Label the time tick marks with default
           increment.
    -hairy        "Hairy" tide clock mode,
          showing slice of tide graph.
```

In order to get a different location and/or the time zone for that location, you will need these:

```
  -list   List all supported locations.
    -location     Specify the location for which to predict tides
    -loctz        Use their time zone, not mine.
```

<f“Courier”>-location random<f$> will cause XTide to choose a location at random.

If you are not interested in keeping a tide clock on your desktop, the next most interesting mode will probably be graph mode. For that, you will need these switches:

```
  -graph  Specify graph mode.
  -gstart Specify the starting time for graphs and
          listings.
```

The format for the time stamps given to -gstart is *YYYY:MM:DD:HH:MM*. For example, half past midnight, June 1, 1995, is 1995:06:01:00:30.

By generating tide graphs for various locations, you will quickly get a feel for the different kinds of tides different places have. Many people are surprised to find out that not all places have two high tides and two low tides each day. Some places, like Bayou Rigaud, Louisiana, have only one high and one low tide per day; other places, like Brunswick, Maine, can have four of each! The latter are known as quarter-diurnal tides; the former (diurnal tides) are also interesting in that they will sometimes *flatline*, showing almost no tidal activity for 24 hours or more.

XTide supports many other modes, including simple ASCII text listings of tides, tide calendars, PPM and GIF output, and even Java-based tide graphs. It also has many options of special interest to folks on the water, such as to apply offsets for secondary tide locations, or to find the times at which the water goes above or below a specific tide level. Most of these options are explained in greater detail in the XTide README.

## Learning More

An illustrated version of the XTide README can be accessed at universe.digex.net/~dave/xtide/. It contains examples of almost every kind of output that XTide can generate and includes full instructions and a FAQ.

You can learn a lot about tides and tide prediction by reading the National Ocean Service's *Tide and Current Glossary*. An old version is preserved at universe.digex.net/~dave/xtide/tidegloss.html for the purpose of providing definitions for the technical terms used in the XTide README. The latest version, currently accessible at www-ceob.nos.noaa.gov/tidegloss.html, has been separated into many smaller web pages for easier browsing.

The canonical reference for tide prediction is the *Manual of Harmonic Analysis and Prediction of Tides*, Special Publication No. 98, Revised (1940) Edition, United States Government Printing Office, 1941. However, much of the traditional lore on tide prediction is not digestible unless you like swimming through pages of equations. Probably the easiest introduction to the subject for programmers is to read the source for the Java applets provided in the XTide distribution. These were written to be as small and simple as possible, and you can easily see where the tides are generated.

Although tide prediction is almost a definition of the term *niche market*, XTide has attracted an amazing number of users, and I hope that it will continue to serve their needs for years to come.

**David Flater** (dave@universe.digex.net) is a Computer Scientist (actual job title) living in the vicinity of Washington, D.C. He escaped grad school two years ago with a Ph.D. in Computer Science and is still trying to regain his sense of humor. All things considered, he'd rather be John Carmack.

Archive Index Issue Table of Contents

Advanced search

# Letters to the Editor

**Various**

Issue #32, December 1996

Readers sound off.

## More FreeBSD Coverage

I am a subscriber to *LJ* and, in fact, have every issue. I hesitate to say this, because it sounds like so much bullshit, but I have always been impressed by how quickly *LJ* came up to speed with quality articles. As a Unix applications programmer, I read several periodicals, and I find your nuts-and-bolts approach refreshing, compared to *DDJ*'s articles. However, I am writing (if that is what this form of communication is properly called) to ask for an improvement in *LJ*. Would your editorial staff please give serious consideration to including articles on FreeBSD? Especially since it poses no threat to the sales of Linux. Furthermore, I think it would increase your sales, at least with some of the programmers that I know.

—ursa@cris.com

## Plenty of Linux

An informal survey made back in *LJ*'s early days suggested the FreeBSD community wasn't interested in a magazine devoted to FreeBSD, which SSC was contemplating. (Perhaps things have changed?)

As far as giving time to FreeBSD within the pages of *Linux Journal*, we do have some ideas we're working on. Right now, though, we've got more material about Linux than we can print.

## Korn Shell

The July '96 issue of *LJ* presents the new Korn shell (ksh93). What is not mentioned (and is not widely known) is users not interested in commercial support can get Linux, Sun, etc. binaries for free (src is not available). This includes not only the ksh binary but also shared libraries and the Tksh extension for Tcl/Tk. Just check the URL www.research.att.com/orgs/ssr/book/reuse. By the way, the book that URL refers to (*Practical Reusable Unix Software*, Krishnamurthy, Wiley 1995), is a gold mine, describing and providing the source code for gems such as the dot tool. Dot is a directed graph layout tool, and now I can't live without it. Recommended.

—Alexandre Valente Sousa avs@daimi.aau.dk

## “Getting to Know gdb”

Mike Loukides and Andy Oram point out in their “Getting to Know gdb” article (September '96) that gdb is capable of setting hardware watchpoints on “only a few workstations”. Not only does the i386 (and beyond) have these capabilities, but current versions of Linux (1.2.1+) and gdb (4.14+) are able to use this feature.

The 386 supports up to four simultaneous hardware watchpoints which can trigger when a memory location is “accessed” (read or written). In Linux gdb, these are the “rwatch” and “awatch” commands. Both commands take an expression to be watched. As the authors point out, this feature is great for watching a memory location that is being mysteriously trashed. The advantage with hardware watchpoints is your program runs at full speed instead of being slowly single-stepped.

The one tricky point in dealing with these watchpoints is they are disabled automatically if program control transfers outside the scope of the expression. For example, if “i” is an automatic variable (a “local” variable), then when a subroutine is called, the meaning of “i” is lost. The workaround is to watch a raw memory location:

```
(gdb) print &i
$1 = (int *) 0xbffffd2c
(gdb) awatch *((int *) 0xbffffd2c)
Hardware access(read/write) /
watchpoint 3: *(int *) 3221224748
(gdb) cont
Continuing.
Hardware access(read/write) /
watchpoint 3: *(int *) 3221224748
123      *foo = 0;
(gdb)
```

—Andy Vaught andy@maxwell.la.asu.edu

## Yay *LJ*!

I couldn't believe it—not one, not two, but three (!) issues of *Linux Journal* in my mail box Wednesday.

I don't know how to thank you for the happy hours I have wiled away in the past two days. I spent several hours just reading ads.

Not since the old days of *Byte* magazine can I remember enjoying a magazine's ads so much.

—Dwight Johnson djohnson@olympus.net

## Anonymous Frenchmen

Bonjour,

Thanks for your very efficient dtree [article].

As an anonymous subscriber, I encourage [you] to continue this way: staying with the technical features is, for now, the best means to sustain the Linux movement.

Merci beaucoup.

Cordialement,

—J-F Bardou

## Useless Distributions Article

I just purchased a copy of the September 1996 issue of *Linux Journal*. I was extremely disappointed that you are excluding any mention of bugs from your review articles. Rely on word of mouth? Then why the devil did I just pay $4.00 for your silly journal!! I know you don't want to offend any potential advertisers, but this is ridiculous. Until you can be bothered to report on what these packages are REALLY like, your journal does not have any value.

—Tim Gawne tgawne@icare.opt.uab.edu

## Texas OODB

I just finished reading your article on the Texas OODB in July's *Linux Journal*.

One free system you did not mention was Exodus (from the now famous OO7 benchmark debacle with Object Store). I have been using Exodus for about two years with its E (C++ extension) for persistence. It is exceptional.

I note from its archive (cs.uws.edu) the latest version of E has been removed (based on gcc-2.5.8), maybe because of several bugs in the release (which I tripped over and have since fixed). The stable release is based on gcc-2.3.3.

I have just completed the port of E-2.5.8 to Solaris 2.4 & 5 and should be releasing it, along with the fixes, back to the archive shortly.

By the way, Exodus does run under Linux and does support a very extensive multi-user, multi-server caching system called “sm”. sm also supports a two phase commit and checkpointing.

I agree about OODB speed differences—they make systems like “Manacle” look like the dinosaurs they are.

—Stephen Dennis dennis@rover.research.canon.com.au

## Israeli Linux Users Group

I want to give you an update on the Israeli Linux Users Group. First, we now have our own domain. The official web site is www.linux.org.il, and a full-scale sunsite-mirroring FTP site is on the way. We are also considering becoming an official non-profit organization, which might be interesting for readers. Subscription to our mailing list can be done at www.linux.org.il/Linux-il-sub.html or by sending mail to majordomo@linux.org.il with the subject line empty and the body containing “subscribe linux-il [address]”.

Finally, I just want to ask that any feedback and comments sent to you as a result of the article also be forwarded to me. I'll make sure the entire group receives them.

—Shay Rojansky roji@cs.huji.ac.il

## Thanks

Thanks to everyone there for a wonderful source of information and advertisements specific to Linux. I receive many magazines each month, but the copy of *LJ* I pick up at the newsstand each month (until now) is by far my favorite. Keep up the good work! (And if there are any technical employment opportunities at *LJ*, I would love to hear about them.)

I would like to know as much as possible about the projected availability of the back issues. I want to get one of each of the issues still available, so it would help to know which to order first. I don't want to miss one by waiting too long.

Thanks in advance.

—Walter Seefeld wseefeld@memphisonline.com

## Article Wish List

I'm writing to congratulate you on yet another fine issue of the *Linux Journal*. Your journal is the only one I read from cover to cover any more. Here are a few opinions and comments I'd like to pass on:

- Please convey to Michael Johnson he did a marvelous job in getting *LJ* started. I always enjoyed his technical articles, and I wish him well in his new position with Red Hat.
- I think your mix of newbie to “not so newbie” articles varies a little between issues but, on the whole, is about right (for me anyway).
- I think your “What is Linux” section should highlight the contributions of the FSF, BSD, MIT and others a little more prominently. Although their software wasn't written to support the Linux project (but it does now), Linux would be much the poorer without them (it doesn't diminish Linux in any way to provide this acknowledgement).
- Please continue to run as many articles as possible on introductions to programming tools under Linux. The “Getting to Know gdb” article was a good example of this type of article.
- I liked your interview with Linus Torvalds. It would be very interesting to see more articles in this vein. I suggest you interview additional Linux luminaries such as Alan Cox, Werner Almesberger, Matt Welsh, Lars Wirzenius, Remy Card, Rick Faith, Leonard Zubkoff, Bjorne Ekwall, Al Longyear, Donald Becker, and many others too numerous to mention.
- I'd also be quite interested in getting to know a little more about Linux entrepreneurs like Phil Hughes, Mark Bolzern, Erik Troan and many others.
- While I'm on the topic of interviews, a good complement to the Linux Distributions review article would be short bios on the founders of the distributions. (I'm particularly fascinated by the Debian project, its philosophy, and its contributors.)
- Interviews with Unix luminaries like Richard Stallman, Larry Wall and others would also be very illuminating.
- An article on the use of Linux for real-time applications would be interesting. There are a number of groups working on this, and perhaps you could convince their coordinators to contribute an article or two.
- An article surveying free and non-free database software would be very useful.

Well, that's enough opinions and suggestions for one letter! Keep up the good work!

—Nick Busigin nick@xwing.org

## Modules and Mr. Crow

I think your treatment of modules both in your article about 2.0 and in Mr. Crow's article about them is totally wrong. You focus only on memory savings. Mr. Crow even goes so far as to explain that for drivers used permanently, it is better to compile them in the kernel due to the 2K loss per module when they are page-aligned—besides, you lose twelve pages due to kerneld.

Let me answer this first. If you have 20 modules loaded (an unrealistically high number) you lose 40K—add the kerneld and you are still under 100K. Well, what's the matter? This is Linux, not stinking DOS. We have no 640K barrier, and 100K will not cause a noticeable difference in speed (unless you have only 4MB of RAM, but today this is rare).

Consider a beginner using 1.2: he must choose from dozens of boot diskettes the one able to support his hardware. Despite this, the kernel has so many unnecessary drivers that it is 1MB too big (and that WILL make an important difference in speed). And despite being so large, this kernel lacks things he wants, like sound support. So our beginner (still barely able to copy a file) is confronted with the task of compiling a new kernel. It is not so difficult, as we know, but this has an undesirable effect: Linux gets the reputation of being a “hackers only” OS—you can't put it in the hands of a person without some computing experience.

Now consider a (future) 2.0 distribution: only one boot image (well, perhaps half a dozen, if you want to optimize for Pentiums). All the drivers are modules. At installation time, the user answers some questions about his hardware, and the installation procedure builds the config files for kerneld and /etc/rc.d/conf.modules for “permanent modules”. The user reboots and he is running.

Your kernel is perhaps **slightly** suboptimal, but recompiling is no longer a requisite. That means handling drivers in Linux becomes a lot easier than editing CONFIG.SYS or AUTOEXEC.BAT. Add to this the new package managers, some configuration tools, and a good file manager—with a little hand holding, I now have a hope to rescue my 15-year-old niece from the clutches of the Evil Empire.

—Jean Francois Martinez jfm@sidney.remcomp.fr

Archive Index Issue Table of Contents

Advanced search

# From the Editor

**Gary Moore**

Issue #32, December 1996

Live free or die.

Hi. I'm the new Editor of *Linux Journal*. If that gives you a sense of deja vu, you likely read a similar note in the September issue. I can only assume someone made *LJ*'s five-day Editor an offer he couldn't refuse.

Similarly, I didn't see how I could say no when Phil Hughes, our Publisher, asked if I'd like to leave my ho-hum job at a state university and become Editor of a publication dedicated to nurturing and promoting the use of Linux. The little operating system that could is now a formidable force, and new product announcements are becoming the rule rather than the exception. The good press Linux is receiving in major media is exciting and impressive. And who would've thought Linux would do so well that the Santa Cruz Operation would start giving their Unix away?

Linux is still free and open, unlike the other Unix-like operating systems. The ability for users to contribute to the development of Linux creates a positive and vibrant scene. Because Linux actually works—and works well—it is truly useful instead of just fun.

Talented people devoting their time to Linux development make it possible for others at many levels of computing savvy to use Linux systems effectively. In many situations, users can just “plug and play” now; Linux is no longer just for hackers. Of course, Linux still has the free tools and open design to accommodate anyone who *does* want to start playing around with code.

So, I had to take this job, really.

I've actually been working with Specialized Systems Consultants, the company that publishes *Linux Journal*, for a couple of years now. I decided I needed to learn about this Linux thing I kept hearing about, joined a mailing list, found out

*Linux Journal* needed some data entry done in the evening... moved on to copy editing, and now here I am.

My computing experience includes: playing with a Cyber on an old TI thermal printing terminal with a 300-baud acoustic coupler, calling up every BBS I could find back in the early '80s, trying to load programs on cassette tape for my TRS-80 Model 1, learning Turbo Pascal and CP/M (a functional OS in under 7K!) on a Kaypro II, taking computer science classes with VMS on old Heath 19 terminals, spending countless hours with Macs and Windoze machines, and, lately, trying everything I can find on my Linux machines. I may have a talent for programming, but I've put more energy into writing and editing, so I need to rely on the people actually making things happen on Linux systems—programmers and developers—for content. For you, the reader, this means well-written articles by people who know what they're talking about.

What I and the rest of the staff of *Linux Journal* want is simple: world domination. Short of that, we want to be the best source of information about Linux and to help promote its use. We think we can do that with how-to texts, programming tips, hardware and software reviews, examples of Linux in the "real world", and articles for the beginner. Is there something you need or want to see? Please let me know (info@linuxjournal.com).

—Gary Moore

# SCO Gives It Away: More SCO vs. Linux

**Gary Moore**

Issue #32, December 1996

Is the Santa Cruz Operation concerned about the growth of Linux? Perhpas. And they have people like Mark Horton to thank for that.

There is a new vendor attempting to enter the “Free Unix” market. You may have heard of them. And Microsoft owns part of them. We're talking about SCO, current owner of what was AT&T Unix. A press release we received says they are offering a free copy of SCO Open Server. Well, actually, it is a license for SCO OpenServer—if you actually want the software you have to pay $19 for the media.

Is this a threat to the Linux community? Not likely. If anything, it is a response on the part of SCO to the threat of Linux to their market.

For those who think it is a threat to Linux, the license is for a single-user version of the product. And, in spite of the fact that they make it out as a return to AT&T's policy to offer Unix free to universities, they don't mention that AT&T included source code. SCO does not.

Sure, there will be some takers on what they offer. It will give people a chance to get a mini-Unix to play with. Then they will be ready to graduate to Linux.

## Mark Horton Passes Away

Mark Horton, one of the Linux community's strongest proponents, has died. In September, his friend, Victoria Welch ( vikki@seastar.org), had this to say about Mark:

> The Linux world suffered the loss of one of its greats on September 9, 1996. Mark A. Horton (from GA Tech) passed away in his sleep. Mark had been in poor health for some time, but of late, he seemed to be making a comeback. His passing blindsided those of us close to him.

I first encountered Mark in the version 0.9x days of Linux. Having reached the end of my rope with SCO and a number of other Unices, I decided to give this “Linux toy” a try. About all I had to lose was some time and if it worked, I'd save $3,000 on another version of the “OS” I was using at the time for another upgrade that “didn't have any problems”. So I boldly stripped off what I was using and installed Linux from my pile of floppies (remember those days?). The rest was history—I was hooked. There were various problems with it at the time, but I was elated. There was no one to tell me, “No, we don't have *that* problem” and I had the actual source code so I could fix it. For a Unix junkie, it was Unix nirvana!

It was at this time I met Mark. He was able to sort out the problems I had and either suggest fixes or workarounds or point me in the direction to go to solve the problem. Corresponding with him via e-mail was a pleasure; his enthusiasm, wit, and gentleness I came to love and now miss deeply. Over time, e-mail led to phone conversations, frequently in the early morning hours. I couldn't get angry with him for the time at which he chose to call since the conversations were always interesting and fun!

Sometime later I separated from my husband and returned home to Atlanta. In the process of job searching, I was in the Decatur area and stopped in to say hello to Mark, and we hit it off wonderfully. He was even more charming in person than via “e-media”. One thing led to another, and I ended up working for him doing technical support for InfoMagic—at subsistence wages—but I was having fun and learning more about Unix than I ever thought possible.

I thought I was a pretty “hot shot” Unix guru, but working with Mark was a humbling experience—I probably learned ten times what I had known. Mark was a stern taskmaster, and I am a far more competent Unix person because of it. Never have I had such a crash course, nor more fun doing it. He would plow through the support calls, usually faster than I could get them out of the queue. Tireless and enthusiastic, he was having fun, and I've never seen anyone draw so much satisfaction from helping others. It was common to field 50+ calls a day—he returned them, and I spent seemingly forever trying to sort out the mumblings from the less-than-optimal voice mail system while trying to keep an ear on what he was doing. His dedication to Linux was amazing to me, even as monomaniacal as I am. He often started work at 0600, taking calls from around the world until long past midnight, and then spent time researching the tough questions or just talking Linux with whoever called. He would take calls from the other side of the planet and help people rather than tell them to call back during normal business hours.

> He was supposed to have limits: support on installation only. But too many people had other questions, and he kept almost everyone satisfied. If anyone made any attempt to solve a problem on his own, Mark bent over backwards to get him through it. Numerous times, we got calls from people who expected to stick the CD-ROM in the drive and immediately be running Linux; even when they were rude, obnoxious, and threatening, this rarely ruffled his feathers (although he did occasionally have some choice comments after a call). Many of the angry callers hung up happy and satisfied, and some would call or send e-mail back saying they had figured things out, apologizing for their attitude earlier. Mark was a master wordsmith with the patience of a saint.
>
> When Mark wasn't directly supporting Linux, he was researching, talking with those writing the packages and elements of the system, discussing improvements, and occasionally writing patches to help out where he could. When not doing all that, he was talking up Linux and making converts; many of the most recalcitrant eventually decided to give it a try and thanked him for it.
>
> Around this time, his health was deteriorating, and believe it or not, it stemmed from physical problems and not the lack of rest—he got energy out of what he did. He ended up in the hospital for a week or two, and I ended up taking him a laptop, modem, and manuals. He hated being there and was working hard to get well enough to “get back to it”.
>
> We quit doing direct support at that time and went over to e-mail support. Three hundred messages per day (plus phone calls from people to whom he'd given his phone number)--it was overwhelming just trying to categorize them and get boilerplate responses together. The categorization was 90% of our time and the other 90% was spent generating the boilerplate. There was simply not enough time in the day to even have hope. We finally admitted it was impossible and basically burned out. Although he burned out, it didn't dampen Mark's enthusiasm or interest. He still tried to help, and we spent many nights discussing the “this-n-thats”, ramifications and solutions, until the sun came up and we broke for breakfast before crashing for a bit of well-earned rest. We worked our arses off, but it was the kind of work you love—we knew Linux was “getting there”.
>
> Some of the more interesting stories I'm sure would be denied—you wouldn't believe some of the places (big places) using Linux internally. My favorite was a call with a particularly knotty network problem (as I recall) which I couldn't solve. I was going to have Mark call the guy back since he was, as usual, buried. The guy offered to hold, and I told him it might be a while and to just give me his number and I'd have Mark get

back to him. He said something along the lines of, “We don't get to surface often, er ah, I mean we don't mind waiting for a bit, we'll just wait.” I had Mark put his other call on hold and take the call. (He would never talk about that one.) Other places I know are using it wouldn't be doing so if Linux wasn't a truly viable and reliable resource. There are thousands of other stories from “the little folk” (us), who found new wonder in computing because of Linux. I could go on, but with most of y'all who read this, I'm “preaching to the choir”.

Mark was never able to keep a “spare” machine—anything he built became part of the critical path. He was an inspiration as he was one of the few people I know who had many machines and actually knew them better than I did. This had a positive side for me; working hard at it, I did keep a spare machine and got to try all the new goodies he was sent, and then we'd hash it all over. If he wanted to play with it, he would come over or play via the network. After the initial experience with him, I was willing to relinquish my claim to being a guru beyond SCO and Slackware, but with his stretching of my limits, I no longer have any qualms about jumping into anything to do with Unix or any Linux distribution. Anything I could say about this extending of my capability, knowledge and self-confidence would not do it justice.

Looking back over the e-mail I have received since I posted news of his passing to comp.os.linux.announce, I am convinced I was but one of many to have this honor and luck.

If it had to do with Linux, Mark was interested. He experimented with various distributions of Linux, including those for the Mac and anything else it might run on. His interest was boundless and his enthusiasm infectious. His quick wit and intellect were a joy to behold in operation.

When I found him on September 7, he appeared to be resting peacefully and was, as usual, surrounded by piles of manuals.

As his best friend and being as close to him as I was, the job of sorting out his machines and turning off mailing lists so his parents could wrap up his affairs left me be amazed all over again at Mark. As well as I knew him, I was only peripherally aware of all the things he was into. The sheer scope of his studies and experimentation with Linux became obvious, if a bit mind-boggling. He was a true hacker in the old definition of the word—research for the joy of knowledge, not intrusion or destruction.

For those of us who knew him, the loss is acute. I'll miss the night-long technical discussions, his curiosity,

> and the joy he radiated doing the things he did so well. My education will not end here, as my relationship with him taught me to stretch my own limits—one of the true joys of Linux is there is no end to learning!
>
> If there is one thing to regret, it's not writing this while he was still alive. He wasn't a high profile Linux personality, but he put more working copies of Linux out there than anyone else I am aware of. He took the most ignorant (“ignorance can be cured, stupidity is terminal”) beginners and gave them a start and undoubtedly infected a high percentage of them with the joy he found with Linux.
>
> Perhaps this learning experience will encourage others out there to spend a few minutes writing about those who deserve credit for Linux being where it is today. I know Mark would have “poo-poo-ed” something like this, as he didn't feel he was really doing anything special, but that seems to be a trait of the truly exceptional.
>
> While I feel a deep sense of loss, I can only try to honor a request he made repeatedly when the subject of “if anything ever happens to me...” came up; he was consistent and firm that there be no moroseness over his passing. A number of his friends are planning to get together to have a “going away” party for Mark and to hoist a beverage of choice to the honor of one of the greats—joy, not sadness, was one of his strongest desires. I'll do my best to honor this but the passing of my best friend and greatest mentor in my life does not make it easy.
>
> At the party I intend to recount some of the things I mentioned here and to raise a glass to the kindest, most loving, most sharing, overall nicest, funniest and most dedicated person I have ever known. Hope you are at peace and in no pain. To know you will be one of the high points in my life; cheers, friend, until we meet again.
>
> Remember the good guys and say what is in your heart while you still can.
>
> I'll close with a quote from a Spider Robinson book that I have found so true with Mark: “Joy shared is multiplied, pain shared is divided.”

# A Beginner's Guide to Compiling Source Code

Larry Ayers

Issue #32, December 1996

If you've been hesitant to compile Linux source code, hesitate no more—this article gives complete instructions for your first foray.

One of the first things a newcomer to Linux often does is search the Internet for interesting and useful programs to run, quickly discovering that many programs are available only in the form of a source-code tree (a form that can be intimidating if one isn't a programmer). For this reason, the new Linux user needs to become comfortable with the compilation process, as it truly is a necessary skill for anyone running Linux.

I'm not a programmer myself, but I do have a basic knowledge of how source code becomes an executable file. Possibly my non-programming status will enable me to bring up information which might seem “too obvious for words” to the experienced programmer. A good introduction to the subject is chapter six of *Running Linux* by Matt Welsh and Lar Kaufman (O'Reilly, 1995).

## Compilation Software

The GNU programming utilities, including the gcc compiler, the make program, the linker and a slew of related tools (many of which you don't necessarily need to know about) are an integral part of most Linux distributions. The Slackware distribution has a menu-driven installation during which you are given the option of having the GNU programming tools installed. If you elected not to install these packages, you will have to start up the pkgtool utility and have them copied to your hard disk.

There are other free compilers out there, but it is advisable to stick with the GNU tools, as they are well-maintained and of high quality. Most Linux software seems to be written with gcc in mind as well, and the less editing of the supplied Makefiles you have to do the better off you'll be.

Applications written in the popular Tcl/Tk programming languages don't generally use the GNU tools; if they do, the C-language components are subsidiary to the Tcl/Tk components. You need to have the Tcl and Tk libraries and executables installed on your system in order to install the source for this type of application. These applications aren't compiled in the usual sense. Installation consists of copying the the Tcl and Tk files to directories specified in the makefile. These programs are completely dependent on their ability to access an existing Tcl/Tk installed base of files, one of the most important of which is the Tk “wish” executable.

As recently as a couple of months ago, it was difficult to maintain a current Tcl/Tk installation; development was rapid, binaries weren't always available and the packages could be difficult to compile successfully. Some of the newer applications required the beta libraries to function. The situation has stabilized recently with the release of the non-beta Tcl-7.5 and Tk-4.1 in both binary and source versions. For these programs most users are better off installing the binaries since, in my experience, they can be difficult to compile from source.

Note that even if you have a reasonably current Linux distribution, the Tcl/Tk versions included may very well be out of date. If you want to run the latest versions of such exemplary applications as TkDesk and TkMan it is well worthwhile to upgrade your Tcl/Tk files.

## Obtaining Source Code

FTP sites can't really be called user-friendly or inviting to newcomers. The file names are often cryptic, and navigating through seemingly infinite levels of a directory tree can be frustrating, especially if you don't know where the file is located. These sites aren't designed for casual browsing, but the maintainers of the large archive sites (e.g., ftp://sunsite.unc.edu and its mirrors) keep the various index files, sometimes in HTML format, which list the available files with brief descriptions. Usually a file called NEW exists which lists recent arrivals. The complete index files can be very large, but are worth downloading in order to be able to use a text editor with a good search facility to search for keywords or file names which interest you.

In general, a file called filename.tar.gz will usually be a source code directory tree, tarred and gzipped. The binary distributions usually have a name patterned as filename.BIN.tar.gz, or filename.BIN-ELF.tar.gz.

Usenet postings in the various Linux newsgroups often contain locations of various packages.

I recommend using NcFtp as an FTP client.This well-written interface to the command-line FTP program has many convenient features, such as remembering every site you visit in a bookmarks file, including the directory you were last in. This feature meshes well with NcFtp's “reget” function, which allows resumption of interrupted file transfers at the point the connection was broken.

Another handy resource is a recent release of a CD-ROM containing a snapshot of one of the major Linux FTP archive sites. Several companies market these CD-ROMs and they are reasonably priced. Linux software changes so quickly that the files on even the most recent of these CD-ROMs will probably be back-level by a version or two, but if you have a sudden desire to compile Xemacs or the Andrew User Interface System, a CD-ROM will save you a lengthy download.

## Dealing with *.tgz Files

NcFtp can be easily configured to deposit downloaded files in /usr/local/src, or wherever you like. Once you have your file it must be unzipped and untarred. Setting up an alias in your ~/.bashrc file (if you use bash) simplifies this process. As an example, the line

```
alias tgz='tar xvzf'
```

in .bashrc will allow you to expand an archive by typing **tgz filename.tar.gz**. The great majority of archive files will create a subdirectory, then expand the files and subdirectories of the archive inside of it. Every now and then you will run across one which expands right into the current directory. You can either list the contents of the archive first (tar tzvf filename.tar.gz), or create a directory and move the files into it (if you've already expanded it).

There are filemanagers available that can expedite these processes. The Midnight Commander text-mode utility can treat *.tgz files as virtual directories, allowing you to dive into them and inspect the contents (read the readme files) without actually expanding the archive.

The Tcl/Tk file/desktop manager TkDesk has right-mouse-button menus specific to archive files; they allow you to list the contents in an editor window and extract to the current directory or the root directory.

One way or another you'll end up with a directory tree containing the source code. There is a useful Unix convention by which files that should be read before attempting the compilation have names in all-caps, e.g., README or CHANGES. There is usually an important file named INSTALL that should be read closely. Since capitalized file names are displayed at the top of a directory listing, these files are easy to find.

## Three Types of Source

Source code packages can be broadly categorized into three types: programs which include a Configure script, programs using imake, and programs with a default makefile.

We'll start with the easiest type, the first mentioned above. Configure scripts are marvelous constructs—basically, they are shell programs that wander at will throughout your Linux system, checking for the presence of various libraries and header files. The scripts use this information to build a proto-makefile, converting it into a Makefile customized to your system. I've found that programs using these scripts compile very easily. After the script does its work, it's usually just a matter of typing **make**, then **make install** when the process is complete.

Many source packages use the **imake** program, usually via its shell script interface **xmkmf**. These packages will have an Imakefile and a makefile.in. The compilation begins with the invocation of **imake**, usually by typing **xmkmf**. Imake is a C preprocessor; it generates your Makefile using makefile.in as a template and information stored in various templates and macro files, usually located in /usr/X11R6/lib/X11/config. Fortunately, you don't need to understand how this works to use it. The Imakefile is the only file you should need to modify, usually just by setting preferred installation paths and the locations of some vital libraries.

The simplest, but most problematic, type of source package has a default Makefile included. This Makefile will have to be carefully edited in order to ensure that the proper libraries are included. Sometimes, especially if the source was written for a machine very like yours, these packages will compile with minimal Makefile-editing. But they can also bomb, so spend a little time trying a few things, and know when the time spent isn't worth the dubious results.

Luckily, the most useful and popular programs tend to compile easily, as a result of more people being involved and submitting bug-reports. Many free programs are available that have makefiles for a wide variety of operating systems, including OS/2, Windows NT—even DOS.

## The Compilation Process

Much can be learned by watching the status messages from your compiler scroll down the screen. Each *.c file in the source directory is first compiled into an object file (*.o). In this process the human-readable ASCII text source file is converted into binary format. This phase is the most time-consuming. If you're still watching near the end of the process, you will see a dramatic flurry of activity as the object files are linked together, and the shared libraries the finished executable needs in order to run are linked as well. Then the process abruptly stops. gcc doesn't tell you that it compiled the executable successfully. It will, however, tell you if there were errors and it can't finish the compile. It's always a kick to do a quick **ls** after the compile, to verify that there actually is a shiny, new, never-before-executed program waiting to be tried.

In my experience, most fatal errors involve the library-linking step. A little common sense helps here; make sure you have the library, and that it is in a location known about by ldd, the library loader. Sometimes the problem is a missing symbolic link, linking a lib in a nonstandard location to one of the normal library directories. If the version of the library you have is outdated or wrong, gcc will say so in the error message.

The Xpm libraries can be a source of compilation problems. There are quite a few versions out there, and some programs are picky about accepting them. If you upgrade to a newer Xpm lib in order to get something to compile, don't be surprised if some of your older X applications stop working. I have yet to figure out a way to have more than one version of Xpm active at the same time. As finicky as Xpm is, it has become a vital part of many X programs. I'm beginning to realize what motivated people to create coherent, upgradeable Linux distributions.

## A Few Hints

In several instances I've failed miserably with a source distribution, then weeks or months later downloaded a later version and had it compile cleanly. Perhaps I updated a library the Makefile was looking for, or perhaps the author made a change in the source which fortuitously caused the program to be compatible with my system. In other words, it's worthwhile to try again later when you initially have a problem.

Another situation I've found myself in: after several edits of the Makefile and perhaps a few header files I'm getting more and more compiler errors. Nothing seems to work and I can't seem to make any headway. This is an ideal time to delete the entire directory tree and reinstall it from the archive file. Sometimes a completely fresh start helps.

One compiler flag to watch out for in the Makefile is -g (as in **gcc -g**). The GNU programs often have this flag, which instructs the compiler to add bulky debugging code to the executable. This is needed if you plan to use a debugger on the program. I don't even have a debugger installed, so I routinely remove that flag. The strip utility will remove this debugging code, often reducing an executable to half its original size.

Virtual consoles are tailor-made for compiling. Once you've set a lengthy compilation in motion, just switch to another console and start something else. I like to shut down X-Windows while compiling, as gcc uses all of the processor cycles it can get. The more resources that are available, the faster your program will compile.

## Conclusion

So what do you gain from learning to compile programs?

- The range of software available to you is considerably increased.
- I believe there is an advantage to using an executable tuned to your system and configuration.
- You have the opportunity to specify compiler flags, like **>\#140>O2**, to optimize the code. Sometimes there are compile-time options that can be set or unset in the Makefile.
- Functions or subroutines in the program you know you will never need can be left out of the executable.
- Source code is often the only form in which successive builds are available in beta-testing scenarios.
- Often more complete documentation will be included with source code than with a binary distribution.
- It is interesting to get glimpses into the way programs are put together. Often source files are heavily commented, because the programmer might want to explain sections of code to present or future collaborators in the project.

**Larry Ayers** (layers@vax2.rain.gen.mo.us) lives on a small farm in norther Missouri, where he is currently engaged in building a timber-frame house for his family. He operates a portable band-saw mill, does general woodworking, plays the fiddle and searches for rare prairie plants, as well as growing shiitake mushrooms. He is also struggling with configuring a Usenet news server for his local ISP.

# dd

**Sam Chessman**

Issue #32, December 1996

It can strip headers, extract parts of binary files and write into the middle of floppy disks; it is used by the Linux kernel Makefiles to make boot images.

The **dd** command is one of the original Unix utilities and should be in everyone's tool box. It can strip headers, extract parts of binary files and write into the middle of floppy disks; it is used by the Linux kernel Makefiles to make boot images. It can be used to copy and convert magnetic tape formats, convert between ASCII and EBCDIC, swap bytes, and force to upper and lowercase.

For blocked I/O, the dd command has no competition in the standard tool set. One could write a custom utility to do specific I/O or formatting but, as dd is already available almost everywhere, it makes sense to use it.

Like most well-behaved commands, dd reads from its standard input and writes to its standard output, unless a command line specification has been given. This allows dd to be used in pipes, and remotely with the **rsh** remote shell command.

Unlike most commands, dd uses a keyword=value format for its parameters. This was reputedly modeled after IBM System/360 JCL, which had an elaborate DD “Dataset Definition” specification for I/O devices. A complete listing of all keywords is available from GNU dd with

```
dd --help
```

Some people believe dd means “Destroy Disk” or “Delete Data” because if it is misused, a partition or output file can be trashed very quickly. Since dd is the tool used to write disk headers, boot records, and similar system data areas, misuse of dd has probably trashed many hard disks and file systems.

In essence, dd copies and optionally converts data. It uses an input buffer, conversion buffer if conversion is specified, and an output buffer. Reads are issued to the input file or device for the size of the input buffer, optional conversions are applied, and writes are issued for the size of the output buffer. This allows I/O requests to be tailored to the requirements of a task. Output to standard error reports the number of full and short blocks read and written.

## Example 1

A typical task for dd is copying a floppy disk. As the common geometry of a 3.5" floppy is 18 sectors per track, two heads and 80 cylinders, an optimized dd command to read a floppy is:

### Example 1a: Copying from a 3.5" floppy:dd bs=2x80x18b if=/dev/fd0 of=/tmp/floppy.image1+0 records in1+0 records out

The **18b** specifies 18 sectors of 512 bytes, the 2x multiplies the sector size by the number of heads, and the 80x is for the cylinders—a total of 1474560 bytes. This issues a single 1474560-byte read request to /dev/fd0 and a single 1474560 write request to /tmp/floppy.image, whereas a corresponding **cp** command:

```
cp /dev/fd0 /tmp/floppy.image
```

issues 360 reads and writes of 4096 bytes. While this may seem insignificant on a 1.44MB file, when larger amounts of data are involved, reducing the number of system calls and improving performance can be significant.

This example also shows the factor capability in the GNU dd number specification. This has been around since before the Programmers Work Bench and, while not documented in the GNU dd man page, is present in the source and works just fine, thank you.

To finish copying a floppy, the original needs to be ejected, a new diskette inserted, and another dd command issued to write to the diskette:

**Example 1b: Copying** to a 3.5" floppy**dd bs=2x80x18b < /tmp/floppy.image > /dev/fd01+0 records in1+0 records out**

Here is shown the stdin/stdout usage, in which respect dd is like most other utilities.

## Example 2

The original need for dd came with the 1/2" tapes used to exchange data with other systems and boot and install Unix on the PDP/11. Those days are gone, but the 9-track format lives. To access the venerable 9-track, 1/2" tape, dd is superior. With modern SCSI tape devices, blocking and unblocking are no longer a necessity, as the hardware reads and writes 512-byte data blocks.

However, the 9-track 1/2" tape format allows for variable length blocking and can be impossible to read with the cp command. The dd command allows for the exact specification of input and output block sizes, and can even read variable length block sizes, by specifying an input buffer size larger than any of the blocks on the tape. Short blocks are read, and dd happily copies those to the output file without complaint, simply reporting on the number of complete and short blocks encountered.

Then there are the EBCDIC datasets transferred from such systems as MVS, which are almost always 80-character blank-padded Hollerith Card Images! No problem for dd, which will convert these to newline-terminated variable record length ASCII. Making the format is just as easy and dd again is the right tool for the job.

**Example 2: Converting EBCDIC 80-character fixed-length record to ASCII variable-length newline-terminated recorddd bs=10240 cbs=80 conv=ascii,unblock if=/dev/st0 of=ascii.out40+0 records in38+1 records out**

The fixed record length is specified by the cbs=80 parameter, and the input and output block sizes are set with bs=10240. The EBCDIC-to-ASCII conversion and fixed-to-variable record length conversion are enabled with the conv=ascii,noblock parameter.

Notice the output record count is smaller than the input record count. This is due to the padding spaces eliminated from the output file and replaced with newline characters.

## Example 3

Sometimes data arrives from sources in unusual formats. For example, every time I read a tape made on an SGI machine, the bytes are swapped. The dd command takes this in stride, swapping the bytes as required. The ability to use dd in a pipe with rsh means that the tape device on any *nix system is accessible, given the proper rlogin setup.

**Example 3: Byte Swapping with Remote Access of Magnet Tape:rsh sgi.with.tape dd bs=256b if=/dev/rmt0 conv=swab | tar xvf -**

The dd runs on the SGI and swaps the bytes before writing to the tar command running on the local host.

### Example 4

Murphy's Law was postulated long before digital computers, but it seems it was specifically targeted for them. When you need to read a floppy or tape, it is the only copy in the universe and you have a deadline past due, that is when you will have a bad spot on the magnetic media, and your data will be unreadable. To the rescue comes dd, which can read all the good data around the bad spot and continue after the error is encountered. Sometimes this is all that is needed to recover the important data.

**Example 4: Error Handlingi:dd bs=265b conv=noerror if=/dev/st0 of=/tmp/ bad.tape.image**

### Example 5

The Linux kernel Makefiles use dd to build the boot image. In the Alpha Makefile /usr/src/linux/arch/alpha/boot/Makefile, the srmboot target issues the command:

**Example 5. Kernel Image Makefile:dd if=bootimage of=$(BOOTDEV) bs=512 seek=1 skip=1**

This skips the first 512 bytes of the input bootimage file (skip=1) and writes starting at the second sector of the $(BOOTDEV) device (seek=1). A typical use of dd is to skip executable headers and begin writing in the middle of a device, skipping volume and partition data. As this can cause your disk to lose file system data, please test and use these applications with care.

### Credits

The dd command has been around since the 1970s, ported to many systems, rewritten many times, and tested by time as a useful tool. The current Linux version is GNU dd GNU fileutils 3.12, written by Paul Rubin, David MacKenzie, and Stuart Kemp, Copyright © 1985, 1990, 1991 Free Software Foundation, Inc.

GNU dd is found in the fileutils collection, with the current version at the URL ftp://prep.ai.mit.edu/pub/gnu/fileutils-3.12.tar.gz or a mirror near you.

Other major versions include SYSV and BSD, with the BSD source version 5.16 4/28/93 derived from software contributed to Berkeley by Keith Muller of the University of California, San Diego and Lance Visser of Convex Computer Corporation, Copyright © 1991 The Regents of the University of California.

**Sam Chessman (SSC3)** (chessman@wauug.erols.com)

# Best of Technical Support

**Various**

Issue #32, December 1996

Our experts answer your technical questions.

## Tuning the Kernel to Recognize RAM

How do I get Linux to recognize more than 64MB of RAM? I presume I may need to tune the kernel. How do I do this? —Edward Longstrom

## Forcing the Issue from LILO

*The reason Linux does not recognize more than 64MB of RAM is actually related to limitations in BIOS. You can force the issue from LILO with an argument of **mem=??M**, where **??** is the amount of physical RAM in the machine and **M** stands for Megabytes. To make this automatic, add that line to the block defining the specifics for each boot (image) configuration. —Dan Lark, SuperNet of Las Cruces, Inc dan@netsteps.com*

## Kernel Doesn't Recognize CD-ROM

I have a Toshiba XM-series CD-ROM that is not recognized by the kernel. I thought it was a standard IDE/ATAPI drive. What could be causing this? —Scott Herscher

## Command Line Parameters

*First, make sure your CD-ROM is connected to a primary or secondary IDE interface. Kernel 1.2.13 will not see tertiary interfaces. Then you may need to give it command line parameters to have the kernel find it. Here is a chart:Primary Interface - Master: hdaPrimary Interface - Slave: hdbSecondary Interface - Master: hdcSecondary Interface - Slave: hddUse the "hd" parameter for your actual device, based on the chart above, to boot with a command like:*

```
LILO boot: linux hdd=cdrom
```

*If you don't know how the CD-ROM is connected, it is safe to try them all. You can then add this as an append line to your* ***lilo.conf****. An example would be:*

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.0.12
        label=linux
        root=/dev/hda2
        read-only
        append="hdd=cdrom"
```

*Make sure to run* ***/sbin/lilo*** *after editing the file. —Donnie Barnes, Red Hat Software* *redhat@redhat.com*

### Netscape 2.02 and Linux

Are there any FAQs about setting up Netscape 2.02 with Linux? After you unzip Netscape where do you put the files? When I set up Netscape 2.02 it gives me the error message **cannot find lib.so.4**. Any ideas? —Marc A. Krushelnyski

### Upgrade Recommended

*First of all, I recommend using 3.0. You can get an ELF version that won't take up nearly as much memory and won't cause the missing library problem you mention. Second, the README that comes with Netscape tells you where to put the files. I'd put the Netscape binary in* ***/usr/local/bin****, then put the zip file in the recommended location. —Donnie Barnes, Red Hat Software* *redhat@redhat.com*

### PPP Error Message

I installed PPP support from the control panel, but when I want to access the pppd in /usr/sbin I get the message **THERE IS NO PPP SUPPORT IN THIS SYSTEM**. I tried to install **ppp-2.2.0f.tar.gz** to see if that would help, but it didn't. I checked the **/proc/net/dev** file with a cat command but all I saw was a column of:

```
s1
s2
s3
```

and so on with many zeros. What must be done to correct this problem? —Dominik Barth

## Installation Error?

*You either didn't install **ppp-2.2.0f** properly, or you didn't compile PPP support into the kernel. A standard distribution kernel should have PPP compiled in, so I would guess your pppd install went wrong.Most likely, you either didn't **make install**, didn't do **make install** as root, or your old pppd binaries live in a different place from your new pppd binaries and the path for the old binaries comes before the newer one in your PATH environment variable. —Bert Vermeulen bert@cnct.com*

## Simple Dialup Program

I cannot set up pppd to connect to my ISP. Chat script is simple in principle but in reality it is impossible to decode and make work. Is there a simple dialup program, preferably in X-Windows, that will solve this problem? —Ken Kim

## Making Your Own Chat Script

*Chat is actually easy to understand. My advice: get rid of those 50-line ppp shell scripts and make your own chat script instead. The man page for chat is very good.The easiest way to do it is to just log in to your ISP manually (e.g., with minicom) and write down the login sequence exactly. Watch out for uppercase/ lowercase mistakes! —Bert Vermeulen bert@cnct.com*

## Linux Won't Recognize Drive

How do I get Linux to recognize my 2.0GB Maxtor drive? I have a 486dx66 with 8MB of RAM and a 1993 AMI BIOS that doesn't recognize the larger drives requiring me to use On-Track for DOS. Would a new BIOS solve the problem? —Scott Sharpe

## Getting off On-Track

*Your problem in this case is the On-Track software and DOS itself. DOS suffers from a lack of foresight. Any disk is made up of a combination of cylinders, heads and sectors per track. For an operating system to be able to operate a drive, it must know these values, called the "geometry" of the drive.DOS was originally written quite some time ago. The function that it uses to determine the disk geometry returns the number of cylinders in a value 10 bits wide. As anybody up to speed with their binary knows, this allows a value from 0-1023.As drives have grown, these values have also risen. Modern hard drives, those that are generally above 500MB in size, have more than 1024 cylinders, causing problems with DOS.There are two solutions. One is compatible with Linux, and the other is not. The most common today is to use Logical Block*

*Addressing (LBA). This is typically provided by the computer's BIOS as a feature to handle larger drives. What it does is double the number of heads and halve the number of cylinders “logically” so that DOS can operate on a drive physically outside its capability. The BIOS handles the translation on the fly, and DOS is none the wiser.Linux will operate quite well with this scheme. It simply does not use that function, and instead determines the geometry itself. However, this is not compatible with the other method, that of using disk management software to handle the problem.Disk management software works in much the same way that the LBA mode does, but it relies on a piece of software loaded in memory when the computer boots. Since Linux (or any other operating system besides DOS) will not load this driver, it cannot read a disk controlled by the manager software.Your best solution is to replace your BIOS, or buy a new motherboard with a newer BIOS. Either solution is probably just as good. Since a BIOS upgrade is a rare event, companies typically charge high prices for them. You may be able to find a 486/80 or something even faster for approximately the same amount of money it would cost you to upgrade your BIOS. —Chad Robinson, BRT Technical Services Corporation chadr@brttech.com*

### Portmap Zombie Processes

My portmap process keeps generating zombie processes, but I can find no reason for it—any ideas? —Chris Kolosiwsky

### Upgrade Should Solve Problem

*Upgrade to a later portmap RPM from: ftp://ftp.redhat.com/pub/redhat/redhat-3.0.3/i386/updates/RPMS. —Donnie Barnes, Red Hat Software redhat@redhat.com*

# IMEC/NIT

**Erwin Glassee**

**Rudi Cartuyvels**

Issue #32, December 1996

IMEC/NIT is taking its Unix applications to its customers on a portable Linux box.

Our research group at the InterUniversity MicroElectronics Center (IMEC—more information about IMEC can be found on www.imec.be) in Leuven, Belgium is working on the simulation of microelectronic fabrication processes and microelectronic devices. Somewhat misleadingly, this is usually called Technology Computer Aided Design (TCAD). Simulation is gaining importance in the microelectronics industry as processes and devices are becoming more complex. Real prototype fabrication currently takes too much time and money in the development cycle for new technologies.

However, producing a good fabrication plan for a new semiconductor technology is difficult. Numerous parameters in the plan, such as furnace temperatures, process duration, implant doses of impurities and so on, influence the resulting semiconductor microstructure. Also, the electrical characteristics of the diodes, transistors and capacitors depend on the geometry and impurity of the contents of the microstructure. The engineer is responsible for coming up with a fabrication plan, such that all the electrical characteristics of the devices not only meet their specified values but also exhibit a minimal sensitivity towards the uncontrollable fluctuations on the numerous process parameters. Trial and error methods, combined with physical insight, were once the only tools available for engineers.

In order to help the engineer on this laborious and grave task, we developed a software package called NORMAN/DEBORA. Input to NORMAN/DEBORA is a template design for the fabrication plan, which contains a likely range for some of its parameters. The package then fires off simulations of some well-chosen fabrication plans and constructs mathematical models for the electrical characteristics.

The engineer can then interpret the template fabrication plan using these models. She will find the critical parameters and will see how the electrical characteristics are linked, whether the specified values are feasible and, of course, which parameter values could be used to achieve this. The engineer can judge the sensitivity of the electrical characteristics when parameter variations occur, as in real fabrication. This will determine the robustness of the fabrication plan. Finally, NORMAN/DEBORA can perform an optimization of the fabrication plan, taking into account specified electrical characteristics and parameter sensitivity.

## From the Laboratory to the Marketplace

Most engineering software currently runs on the more powerful Unix systems of various breeds, and so does NORMAN/DEBORA. When the product was leaving the research phase, demonstrations were held for potential customers. At first, we envisaged the microelectronics industry as the sole user. However, the general concept of iterative simulations, plan evaluation and optimization could also be applied to other fields of engineering, such as injection molding of plastics and mechanical analysis of dynamics, acoustics and vibrations.

Finding a place in the sun in the market for engineering software proved to be tougher than we expected. We visited customers, gave demonstrations and prepared successful examples on the customers' particular types of engineering projects, using their engineering software. For easy travelling, and to avoid installation procedures, we decided to install the software on a portable machine. Linux was the most attractive operating system, because it installs on a small, inexpensive portable PC. Since Linux itself is free, using it would be extremely cost-effective, with no compromises in the required functionality.

- Linux was hooked up to our network of Unix workstations. In order to do this, we needed the PCMCIA driver package in addition to the common distributions. Currently, we are using FTP for transferring the source code, and remote login for working on the machine. To integrate the machine even further, nfs could be used, but it would require reconfiguration when the portable is not connected.
- We can run the simulators from our portable machine using the remote execution facility of NORMAN/DEBORA. However, most companies don't allow us to connect our portable to their network.
- For porting, Linux has a POSIX-compliant programming interface and a good compiler for the C language readily available. As some older and

mathematical modules were written in FORTRAN, we needed f2c to complete the porting since there was no full compiler for FORTRAN available at that time.

- Porting our graphical user interface required an X server for our graphical hardware, which needed to be selected with care. As the user interface had been developed with Tcl/Tk, this step also went smoothly. We are using gnuplot to generate graphs and pbmtoxbm from the netpbm graphics format conversion library to display these as X bitmaps. As some companies don't want to have free software installed on their machines, we intend to replace this graphical interface, using widgets from the Motif library, which is available for Linux at a low price.

## Conclusions

The time spent in configuring Linux on a portable and porting our software to it has certainly proven worthwhile. The machine has been used to give demonstrations at customers' sites with great ease, including two tours in the US and Canada. We are also working on engineering projects at customer sites and using the machine for a three day training course, held at the customer's site Our group members regularly take the machine home for the weekend when it's not being used by anyone else! Some of our customers are: Philips in the Netherlands, OKI in Japan, Kodak and Xerox in the US. Outside the field of microelectronic General Motors in the US is also using NORMAN/DEBORA.

The marketing and further commercial development of the package is handled by Numerical Integration Technologies (NIT), the leading company in the field of software for acoustic engineering.

As it hardly does any marketing itself, Linux usually enters a commercial organization through its supporters. A lot of them are software engineers, and they can have a hard time convincing their managers Linux is a viable and even superior solution. Therefore we want to thank the many programmers who have been developing the system and its application software. We intend to continue porting our software to the system at NIT.

**Erwin Glassee** has been a Unix software engineer for two years. He formerly worked at IMEC, and is now working at NIT.

**Rudi Cartuyvels** works at IMEC.

# New Products

**Amy Kukuk**

Issue #32, December 1996

Visual SlickEdit Version 2.0, C++ Version of Andrew User Interface, JClass LiveTable Pro & LiveTable Applet and more.

## Visual SlickEdit Version 2.0

MicroEdge, Inc. has announced release 2.0 of Visual SlickEdit, the editor for programmers. New features include API apprentice, which reduces complicated API calls down to filling out a dialog box, a C/C++/Java code beautifier, difference editing, selective display (code folding), code block selections and hex editing. SlickEdit is available for Linux at a price of $195.

Contact: MicroEdge Inc., P.O. Box 18038, Raleigh, NC 27619, Phone: 919-831-0600, Fax: 919-831-0101, E-mail: sales@slickedit.com, www.slickedit.com/.

## C++ Version of Andrew User Interface

Carnegie Mellon University's Andrew Consortium has released the first C++ version of its Andrew User Interface System (Andrew7.4), an integrated suite of compound document applications with which Linux users can create documents containing combinations of text, pictures, spreadsheets and other embedded objects. Andrew is available free of charge, and has been released in binary form to simplify its installation. The entire suite of Andrew7.4 applications can be found at www.cs.cmu.edu/~AUIS. A comprehensive user's guide is available in print format for $25.

Contact: The Andrew Consortium, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, Phone: 412-268-6710, E-mail: zanger@cs.cmu.edu, www.cs.cmu.edu/.

## JClass LiveTable Pro & LiveTable Applet

KL Group Inc., a leading provider of Motif GUI components, announced the release of JClass LiveTable Pro and JClass LiveTable Applet, Java table components that enable developers to build interactive tables and forms for Java and WWW applications. LiveTable Pro is a complete Java class library and applet that provides the building blocks for creating dynamic forms and spreadsheets. LiveTable Applet enables designers to bring HTML tables to life by adding scrolling view, on-the-fly sorting, and in-table searching capabilities. LiveTable Applet is available for $99, and LiveTable Pro is $399. Both can be downloaded from the KL Group's web site at www.klg.com/.

Contact: KL Group Inc., 260 King Street East, Third Floor, Toronto, Ontario, Canada M5A 1K3, Phone: 800-663-4723, Fax: 416-594-1919, E-mail: info@klg.com , www.klg.com/.

## FairCom Web Server

FairCom Corporation announced the release of its new FairCom Database Servers and c-tree Plus File Handler for use as a web back end. These new servers utilize the same heterogeneous network support offered by all FairCom Servers. These servers are available for many operating systems including Linux. The FairCom Web Server offers multi-protocol interaction between the client and server processes giving transparent file access to the end user. Web servers start at $445 and are licensed on a per server machine basis. OEM distribution and source are available.

Contact: FairCom Corporation, 4006 W. Broadway, Columbia, MO 65203, Phone: 573-445-6833, Fax: 573-445-9698, E-mail: faircom@faircom.com, www.faircom.com/.

## New Release of ObjectSpace C++ Class Libraries

ObjectSpace Inc. announced the second generation of its C++ Component Series including Systems<ToolKit>, STL<ToolKit> and Web <ToolKit> products. Version 2 of the C++ Series consist of 10 C++ class libraries including the most portable version of the ANSI/ISO Standard C++ Library available today. ObjectSpace class libraries come with full source code, extensive on-line or printed documentation and hundreds of examples. Full technical support is available. For pricing use the following contact information.

Contact: ObjectSpace, Inc., 14881 Quorum Drive, Suite 400, Dallas, TX 75240, Phone: 214-934-2496, Fax: 214-663-9100, E-mail: scarrol@objectspace.com , www.objectspace.com.

## CocoBase Java DataBase Access System

Thought Inc. announced CocoBase, a new family of Java based DataBase Access modules for creating and maintaining Java to relational database mapping using Java JDBC. CocoBase creates an internal catalog which can be dynamically modified without recompiling any application. A single map can span multiple RDBMS tables, not only for lookups, but also for updates, inserts and deletes. The CocoBase product, CocoMass, includes CocoPowder for object data retrieval and storage and CocoButter for easy GUI administration of objects to RDBMS mappings, is available for $4,000. Optionally, CocoNibs designed for remote access to CocoPowder using RMI and CocoBeans designed for remote access using CORBA can be added for $995 each.

Contact: Thought Inc., 2222 Leavenworth St., Suite 304, Phone: 415-928-4229, FAX: 415-567-9945, E-mail: info@thoughtinc.com, www.thoughtinc.com

## InTerNet-LINK

Illimite, Inc. announced the availability of the InTerNet-LINK providing small-to-medium size organizations with inexpensive desktop Internet and/or Intranet Access. InTerNet-LINK is a self contained server built on the Linux operating system. It is designed to require a minimum of effort to install and maintain. InTerNet-Link lists for $3,995 for a 5-8 user system to $5,495 for the 25-50 user model.

Contact: Illimite, Inc., P.O. Box 18529, Rochester, NY 14618, Phone: 716-271-7128, Fax: 716-461-0431, E-mail: morcote@eznet.net, www.shopex.com/link/.

## Linux Advanced GUI Toolkit

Microline Software has announced a free Linux advanced GUI toolkit. This toolkit is based on the Microline Widget Library for Motif and contains all the same features that are listed on the Microline web site at www.mlsoft.com/xml/xml.html. Software can be download free from ftp://ftp.best.com/pub/mlsoft/motif.

Contact: Microline Software, 1095 East Duane Ave., Suite207, Sunnyvale, CA 94086, Phone: 408-245-5116, Fax: 408-245-5126, E-mail: info@mlsoft.com , www.mlsoft.com/.

## Introduction to Programming Java Applets

MindQ Publishing Inc.'s Introduction to Programming Java Applets is a multimedia CD-ROM tutorial designed to teach newcomers to experts using animation, audio, video and hypertext. CD-ROM includes a JDK (Java Developer's Kit). This program retails for $49.95.

Contact: MindQ Publishing, Inc., 450 Springpark Place #1200, Herndon, VA 22070-5243, Phone: 703-708-9380, Fax: 703-708-9381, E-mail: MikKass@aol.com , www.mindq.com/.

### SOLID Server 2.1 for Linux

Solid Information Technology Ltd., announced the availability of the Linux version SOLID Server, a relational database management system (RDMS) with native support for SQL and ODBC. It is designed in strict adherence to standards, it is extremely compact, and it is easy to install and administer with an automatic backup feature and error recovery. A single-user desktop version is priced at $99, and multi-user versions are $199 per user.

Contact: Solid Information Technology Ltd., Huovitie 3, FIN-00400 Helsinki, Finland, Phone: +358-0-477-4730, E-mail: martin.mickos@solidtech.com , www.solidtech.com/.

### COS/Relay

Open Systems Management (OSM) announced Cos/Relay, an automated software distribution and installation package. COS/Relay automates the process of software installation across a distributed network from just one console. Comprehensive audit information is automatically collected, recording details on the software installations. COS/Relay is available for Linux, and is a part of the COSMOS suite of applications. It can be installed as a stand alone product for $200 per license (quantity discounts available).

Contact: Open Systems Management Inc., 1111 Third Avenue, Suite 2500, Seattle, WA 98101, Phone: 206-583-8373, Fax: 206-292-4965, E-mail: mike@osminc.com.

### Linux Serial I/O Solutions

Digi International Inc., a provider of connectivity solutions, announced that its broad range of serial I/O solutions now fully supports the latest Linux kernel operating system, version 1.2.13 and 2.0. Digi products that now support Linux include the PC/Xe, PC/Xr and PC/Xem line of asynchronous serial port boards, as well as the Digi RightSwitch, an internal switch for Ethernet server connections. Driver for PC/Xi, PC/Xe and RightSwitch are built into Linux 2.0 kernel. Drivers can be downloaded from ftp//ftp:dgii.com/drivers/linux/digiRel. 100g.tar.gz.

Contact: Digi International, 11001 Bren Road East, Minnetonka, MN 55343, Phone: 800-344-4273, Fax: 612-912-4952, E-mail: sales@dgii.com , www.dgii.com/.

### CRiSP v5—Multiplatform Professional Editor

Vital Inc. announced CRiSPv5, Multiplatform Programmer's Editor with Hypertext help, visual templates and dynamic colorization on all platforms. Many new features have been added, and old ones enhanced. Complete documentation is now available on-line. CRiSP v5 is available for Linux at a price of $149.99.

Contact: Vital Inc., 4190 Candlewyck Drive, Plano, TX 75024, Phone: 214-612-2684, Fax: 214-612-3326, E-mail: sales@vital.com , www.vital.com/.

### Cross-Platform Web Agent-Trudger

Vital Inc. announced Trudger, a dynamic multi-platform Internet web agent that makes web surfing possible without a live connection. A preview version can be downloaded from Vital's web site. Trudger uses asynchronous replication to stop an in-progress download with an ability to resume it later, and maximizes transfer rates by parallelizing the download of related links. Trudger is available for Linux at a price of $59.99.

Contact: Vital Inc., 4190 Candlewyck Drive, Plano, TX 75024, Phone: 214-612-2684, Fax: 214-612-3326, E-mail: sales@vital.com, www.vital.com/.

### NExS version 1.2 for Linux

X Engineering Software Systems announced the availability of the NExS 1.2 spreadsheet for Linux. NExS, the Network Extensible Spreadsheet, is a full-featured, graphical spreadsheet developed specifically for Unix and the X Window System. NExS has more than 200 built-in business and scientific functions, and can import and export data in a wide variety of formats (including HTML). In addition the conNExions application programming interface (API) gives external processes complete control of NExS spreadsheets. Demonstration copies can be downloaded from www.xess.com/. NExS is priced at $49 for a single-user, stand-alone spreadsheet, $149 with the added conNExions API and $249 for a floating license.

Contact: X Engineering Software Systems, 2608 Sweetgum Dr., Apex, NC 27502, Phone: 919-387-0076, Fax: 919-387-1302, E-mail: devb@xess.com, www.xess.com/.

## Networking Switch for Access 8 PCs

Network Technologies Inc., announced the introduction of the SC-2X8-6M6M15V-A, a two user, eight PC keyboard, monitor and mouse switch. It can be used to maintain and update networks, to provide technical backup to computer presentations, or to allow an additional point of access to multiple computers. The price for this unit is $3,630.

Contact: Network Technologies Inc., 1275 Danner Drive, Aurora, Ohio 44202-8054, Phone: 216-562-1999, Fax: 216-562-1999.

## Acu4GL for ODBC

Acucobol, Inc. announced the release of Acu4GL for ODBC, a product that interfaces seamlessly from COBOL to Open Database Connectivity (ODBC) compliant data sources. This interface gives developers the ability to communicate with many different database formats without having in-depth knowledge about them, and without writing customer database queries. Acu4GL works behind the scenes to satisfy each request of the executing COBOL application by generating the appropriate ODBC calls. For pricing, use the following contact information.

Contact: Acucobol Inc., 7950 Silverton Avenue, Suite 201, San Diego, California 92126, Phone: 800-COBOL-85, Fax: 619-566-3071, E-mail: info@acucobol.com, www.acucobol.com

## InfoMagic Workgroup Server

The InfoMagic Workgroup Server provides high-performance file and printing services to PC and Macintosh clients using the Linux operating system. It is the first Linux distribution designed specifically for servers. Based on networking software created by the Internet community and already in use at hundreds of companies, universities and organizations worldwide, the InfoMagic Workgroup Server provides simple graphical tools for system administration and set-up. A Unix novice can set up a sophisticated server environment in a couple of hours. The InfoMagic Workgroup Server is available unsupported for $75, and supported for $499.

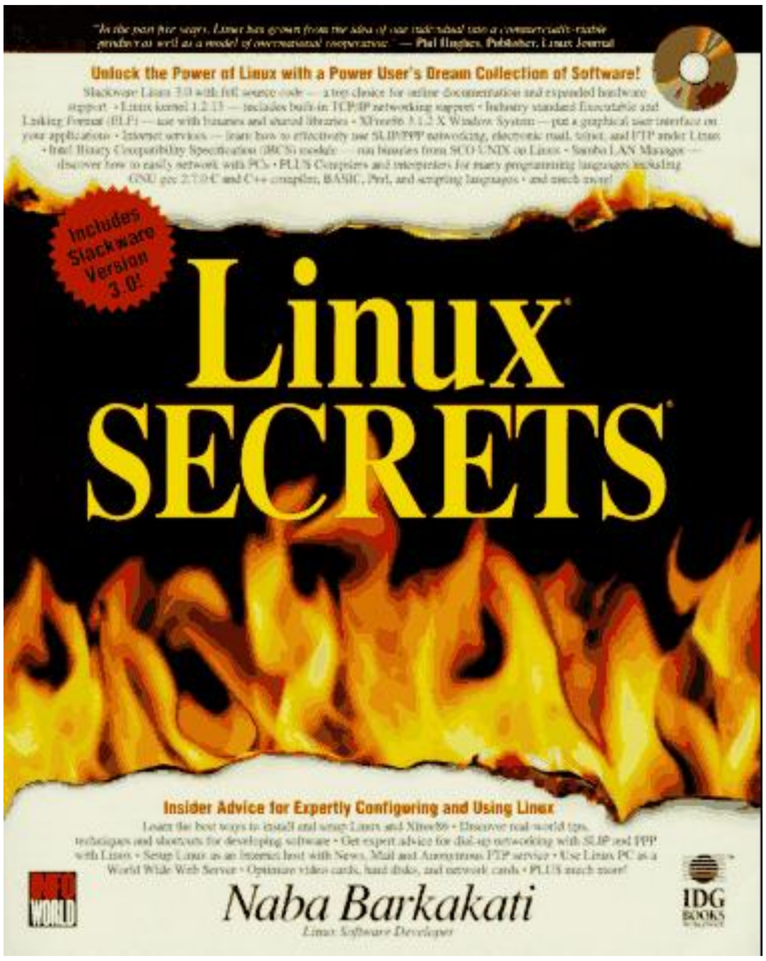
Contact: InfoMagic, 11950 N. Hwy 89, Flagstaff, AZ 86004, Phone: 800-800-6613 (US & Canada) or 520-526-9565, FAX: 520-526-9573, E-Mail: Orders@InfoMagic.com , www.infomagic.com/.

# Linux SECRETS

David Black

Issue #32, December 1996

What I looked for in this book, among other things, was information that can actually be used.


Linux SECRETS
Unlock the Power of Linux with a Power User's Dream Collection of Software!
Includes Slackware Version 3.0!
Insider Advice for Expertly Configuring and Using Linux
Naba Barkakati
IDG BOOKS

**Author:** Naba Barkakati

**Publisher:** IDG Books, 1996

**ISBN:** 1-56884-798-X

**Price:** $49.99 (includes CD)

**Reviewer:** David Alan Black

“Naba Barkakati,” says I when I received the review copy of *Linux SECRETS*, “That's the guy who wrote *The Microsoft Macro Assembler Bible*.” I'm a low-level guy from way back, and in my DOS assembler programming days, Naba's books always stood out for their readability, seriousness, and thoroughness. *Linux SECRETS* continues this tradition.

I'd be bluffing to generalize too much about “what to look for in a mass-market Linux book”, because this is the first one I've owned. So I'll be specific. What I looked for in this book, among other things, was information that can actually be used. In other words, I kept it close at hand in order to see whether or not I ever actually referred to it. The answer is that I did—not every day, but enough to continue to keep it close.

I also tested the book's clarity and utility by using its instructions for a project I'd never done before—namely, writing a **man** page. The instructions for this neither are, nor pretend to be, a **groff** tutorial; but they make sense and they work. (You can judge for yourselves—see Figure 1.) It's always been my impression that Naba has a gift for explaining, and this book confirms that impression.

*Linux SECRETS* covers an enormous amount of material. The table of contents alone is a full twenty pages long, with chapter titles such as Installing Linux; Secrets of X under Linux; Secrets of Linux Commands; Scripting with Tcl/Tk; Running a Business with Linux; Networks; Modems; Printers, and Text Processing. The book's stated goal is to serve as “a practical guide that not only gets you going with the installation and setup of Linux, but also shows you how to use Linux for a specific task, such as an Internet host or a software development platform.”

The book views different topics at levels of magnification; it explores some topics in telescope-like detail; for others, it uses binoculars; and sometimes it looks through those devices from the other end.

The topics presented in the most detail include installation and other setup-related procedures. My experience on both sides of the virtual help desk suggests that no single written source will ever get everyone from blissful ignorance to Linux Nirvana with no problems and no outside help; but anyone who studiously follows Naba's installation chapter will see that it comes very close. Along the way there is much supplemental information, such as remarks on the 16/32-bit OS distinction, sidebars about the BIOS and DOS, and so forth.

While we're on the subject of installation: the book and CD-ROM are Slackware-oriented. If you have reason to care—for instance, if you're starting out and have decided in advance to use a different distribution—you should be aware that the only distribution on the CD-ROM is Slackware 3.0, and much of the treatment of installation, setup, and upgrading assumes a Slackware-based system. You should also keep in mind that a great deal of the material in the book is distribution-independent.

Also covered in considerable detail is the topic of hardware. There's actually a chapter called “Computers”, which forms a bridge between a general look at hardware architecture and Linux-specific compatibility concerns. Input devices, CD-ROMs and sound cards, video hardware, disk drives, printers, and modems all receive their own chapters. As always, these chapters include information about how things work generally, as well as detailed discussion of how they work under Linux.

Shifting from telescope to binoculars, we find coverage of the things you can actually *do* with Linux, centering around Part IV, “Using Linux for Fun and Profit”. Almost half of this chapter is devoted to networking (setting up a server, running an ISP, and so on); and while the binoculars perhaps get picked up backwards here and there, overall this section contributes much—not only in practical advice, but in making the point that Linux has real sweep as a platform for many projects.

At the sketchier end, the book includes sections on such topics as vi, shell programming, emacs, Perl, and Tcl/Tk scripting, as well as a “Linux Applications Roundup” which introduces several other editors and applications (gzip, Workbone, Ghostscript, etc.). Much of this material, of course, is second nature to many of us, but I have found some of these short sections very handy for getting an overview of a topic and deciding whether I want to pursue it. In this respect, the less detailed sections can be very helpful.

Overall, the lesson learned from breaking contents down by level of magnification is that while no one is likely to need or use everything in the book, many people at varying levels of expertise with Linux will find a good fit between their needs and the book's coverage. I'll just add that the author provides guidance on where to go for further information, both interspersed throughout the text and listed in a “Linux Resources” appendix. (Of the various resource subheadings, “Magazine” is the only one that appears in the singular!) Even if some of the pointers to resources remind us how quickly things change (e.g., the inclusion of comp.os.linux.help), it helps keep you in touch with the world of Linux.

Linux will always be (I hope) too dynamic a thing to be captured fully on paper. But while we watch it take off, explode, spread, and do various other non-static things, it's nice to have written works that are lucid, informative, and in touch with the spirit of the project.

**David Alan Black** (dblack@icarus.shu.edu) (dblack@candle.superlink.net) found many people visiting his home computer's HyperNews installation after he wrote his last *LJ* article, and scrambled to make it functional. He welcomes e-mail about Linux, *The Avengers*, and various other topics.

# The OPEN LOOK and XView CD-ROM

**James Waldby**

Issue #32, December 1996

This CD-ROM is aimed at developers and users of OPEN LOOK software or the XView toolkit for the X Window System.

**From:** Darwin Open Systems

**Price:** $30.00

**Reviewer:** James Waldby

OPEN LOOK is a graphical-user-interface specification. For example, the Sun OpenWindows windowing environment is based on OPEN LOOK specifications, while olwm and olvwm are OPEN LOOK window managers often included with OpenWindows implementations. XView is an X11 toolkit for developing programs that meet OPEN LOOK specifications. The “Linux GUI Development mini-HOWTO” refers to XView as “the poor man's object-oriented toolkit for building OPEN LOOK applications for X” but also says, “The XView toolkit provides extensive attribute-value pair combinations, convenience routines and object class hierarchies...” I comment briefly on XView vs. Motif later in this review.

The CD I reviewed was released in mid-1995 by the Darwin Open Systems company in Ontario, Canada. Priced at USD $30, the CD contains about 570 megabytes of files, mostly uncompressed. The principal directories are: src/X11R6/contrib, src/X11R6/xc, src/xview, book, bin.Linux, NeWS, bin.sun and t viewers. Contents of some directories are of use only on Suns (bin.sun, of course; a set of indices for SearchIt; AnswerGarden/xinfo files), but most files either appear in two forms (Sun OS and Linux) or aren't OS-specific. A lamentable exception: acroread-msw.exe and acroread.sun Acrobat/PDF viewers are supplied on the CD—but no Acrobat viewer for Linux is supplied.

One page accompanying the CD describes licensing terms for its contents, and a second page tells how to mount the CD on your system and “get started” with it. You can get a quick overview of much of the disk using an HTML browser and the provided TOUR script. The script will use Arena (provided on the CD) if you like, or any browser you already have. I used both Netscape and Arena to navigate around the disk.

The CD contains text and figures of three books, licensed for personal use. The books are volumes 7A *(XView Programming Manual)* and 7B *XView Reference Manual)* of the O'Reilly & Associates Guides to the X Window System series, and an analog of volume 3 *X Window System User's Guide)* tailored toward XView instead of Motif or Athena widgets and windows. Each book appears in two forms: as single files suitable for Acrobat/PDF viewing, and as chapter-by-chapter PostScript files for viewing or printing. It is these three books that make the CD unique, rather than the software on the disk; most of the software is freely or inexpensively available on Internet or other CDs. Still, having the software collected and organized on one CD adds value because that makes it easier for programmers to study and run several XView-based applications quickly.

Documentation on CD has its virtues, the most important in my view being findability (I often misplace books), transportability (much lighter than books), and machine readability (for searching and copying). But on my system (Linux 1.3.81, 486DX2-66 with 32MB RAM, 17" monitor) ghostview seems clumsy and slow, so I'm not inclined to recommend the PostScript files on this CD as an online substitute for printed manuals. Instead, I'd recommend the CD to persons learning about X11 or XView because it provides a broad set of example X and XView programs. Now, if all those .ps files were .html's ...

The src/X11R6/xc/doc directory subdivides into directories with many megabytes of troff and PostScript files for X library descriptions, X specifications, and X man pages. For example, xlib.PS.Z is a 490-page document about xlib, widgets.PS.Z is 146 pages about Athena widgets, and so forth.

The src/xview directory leads to source and executable versions of about 50 XView application programs in categories such as general, accessories, graphical, networking, administrative, games, editors, programming tools, and GUI program skeletons. Applications include (for example) workman, xrolo, xvnews, ftptool, and mahjongg.

Many of the HTML pages for the CD tour let you try out applications easily. For example, the page with URL file://localhost/CD/src/xview/pan3.0/index.html lets you run “pan” (post a note) by clicking on “Try it!”. There should also to be a link on these pages, to make it easy to view source code with your browser; but there isn't, so those pages without “Try it!” links feel like dead ends.

I noticed glitches and unexpected results in some tour links; for example, catcher and contool (listed among the XView example programs) are not actually on the CD; their pages say that copyright prevented inclusion and to use FTP to get a copy. The catcher page has a clickable link for catcher.tar.Z from rain.com, but the contool page merely says “use archie or your favorite Net Searching tool to find a copy.” For psview, the tour futilely refers to src/misc/psview instead of src/motif/psview. The workman “Try it out” button is displayed by $$Try it out.$$. The TOUR script determines it is running on a Linux system by detecting file /vmLinux which didn't exist on my system; perhaps it should say “elif [ -f /vmLinux -o -f /vmlinuz ]” instead of just “elif [ -f /vmLinux ]”. Of course, minor problems like these are easily dealt with by us Linux experts and need not deter anyone from buying the CD.

The HOWTO document that refers to XView as “the poor man's object-oriented toolkit for building OPEN LOOK applications” also says “Motif has become the standard user interface for X Window System applications.” I don't know enough about Motif or OPEN LOOK to compare their technical qualities, but at a local bookstore saw a dozen Motif titles in stock vs. two XView or OPEN LOOK titles. This is consistent with Ian Darwin's comment (in article 14.03 of the O'Reilly book, *X User Tools*, by Mui and Quercia) that “OPEN LOOK has essentially lost the commercial GUI wars to Motif.” But he concludes positively that “olwm and the XView toolkit will live forever, ensconced on tens of thousands of CD-ROMS and on Linux and other ... distributions.”

**James Waldby** is a Computer Science graduate student at the University of Illinois at Urbana-Champaign

# Running Linux Companion CD-ROM

**Josh Turiel**

Issue #32, December 1996

O'Reilly & Associates intends to update the Companion CD-ROM on a fairly frequent basis (every six months or so), in order to stay fairly current with the state of Linux development.

**Publisher:** O'Reilly & Associates and Red Hat Software

**ISBN:** 1-56592-212-3

**Price:** $24.95

**Reviewer:** Josh Turiel

When Matt Welsh and Lar Kaufman wrote their excellent *Running Linux* last year, one of the things I noticed about it was that, unlike most Linux books to that date, *Running Linux* was not really written with any one distribution's idiosyncrasies in mind. Also, most Linux boxes typically have a CD-ROM with whatever distribution they favor (historically the Slackware *du jour*).

*Running Linux* had none. With the *Running Linux Companion CD-ROM*, a separate 100-page volume with a bound-in copy of Red Hat Linux 2.1, O'Reilly enters the Linux distribution business. Unlike most, Red Hat's Marc Ewing has written the guide, aided by Donnie Barnes and Erik Troan, also of Red Hat.

The included copy of Red Hat Linux 2.1 is precisely the same as any other version the reader may have encountered—in fact, as I am writing this, the current release is 3.03, so this one is even a little out-of-date. However, the point of the *Running Linux* package is not to have a bleeding-edge distribution for all the hard-core Linux enthusiasts to pound on; it is to give readers a combination of several assets:

- The O'Reilly brand name, a staple to Unix sysadmins and network pros everywhere
- A book oriented to the user who wants to become familiar with Linux
- A software distribution for those who want to put theory into practice (backed with the first two assets)
- The CD-ROMs in the package contain the Red hat 2.1 installation and a Sunsite archive of freely distributable software (not all in the RPM format)

The manual seems to have a definite bent towards the user with previous network experience—or possibly those who are already familiar with the Linux information in the original *Running Linux*. It's interesting at times, since a section like the Xconfigurator description seems to be written with novice-level users in mind (with instructions explaining what kind of graphics chip your video card probably has, among other relatively basic features), while the section immediately following on Sendmail opens with the following paragraph:

> A default *sendmail.cf* file will be installed in */etc*. The default configuration should work for most SMTP-only sites. It will not work for UUCP sites; you will need to generate a new *sendmail.cf* if you need to use UUCP mail transfers.

My question here is not so much a criticism as a comment. A trend I have noticed in many Linux books is to present instructions like these for the more complex software packages. Remember, O'Reilly's own definitive *Sendmail* reference is over 800 pages long, so obviously all the needed information doesn't fit here. But it would be handy to at least explain some of the relevant terms with bullet points or something similar. Even though the ideal use of this set is with the *Running Linux* volume, having this information in a slightly less elliptical form would be helpful in making Linux less threatening for the first-timer.

That aside, there is excellent documentation provided on *glint*, the Red Hat package manager for X, and the Red Hat FAQ at the end of the book is very helpful (particularly the clearly written information on the Trident 9400 series chip-set). One thing that was not completely clear from the documentation is the support arrangements for the *Companion CD-ROM*. Although its authorship by Red Hat may lead you to believe otherwise, support on this product is not available from Red Hat directly without the purchase of a support agreement. Support on the *Companion CD-ROM* is available on the Internet (through Red Hat's mailing lists and the comp.os.linux.* hierarchies), and O'Reilly provides installation support via e-mail.

O'Reilly & Associates intends to update the *Companion CD-ROM* on a fairly frequent basis (every six months or so), in order to stay fairly current with the state of Linux development. As a first step into the actual software distribution business, this is a fairly sound strategy. They've associated themselves with the distribution that is arguably the most “commercial-ready”, provided very solid reference material, and split the product into both a book (with relatively infrequent updates) and a CD product that can be updated more often. Combined with the new attention to Linux appearing in O'Reilly's newest editions of their indispensable Unix guides, the *Running Linux Companion CD-ROM* is a product that can only help in the ongoing penetration of the market by Linux.

**Josh Turiel** (josht@janeshouse.com) His web page is www.janeshouse.com.

Archive Index Issue Table of Contents

Advanced search

# Linux Kernel Internals

Phil Hughes

Issue #32, December 1996

The translation job is excellent and the content of the book fulfills expectations based on the book title.

**Authors:** Beck, Böhme, Dziadzka, Kunitz, Magnus, Verworner

**Publisher:** Addison-Wesley

**ISBN:** 0-201-87741-4

**Price:** $45.14 (includes CD)

**Reviewer:** Phil Hughes

*Linux Kernel Internals* is an English translation of a book originally written in German and published in early 1994. I was immediately concerned since I have read literal translations from German to English that were very hard to understand.

Fortunately, my fears were unfounded. The translation job is excellent and the content of the book fulfills expectations based on the book title.

If you intend to write kernel code, or a kernel module, or just want to know how the kernel of a Linux system works, this book is an excellent source of information. Also, if you want to know how to build a kernel and understand what you are doing, understand file systems, networking, or even just how a system boots, this book will answer your questions.

As I was reading the book to write this review, I found myself slowing down to carefully understand all the Linux kernel structures. The information is there, and by reading the book cover to cover, you will learn all about the kernel.

On the other hand, if you just want to know about some specific feature—like how timer interrupts work or how to debug a kernel module—that information is included as well, and presented in an easy-to-find, easy-to-understand fashion. But, not too easy. That is, if you don't know C, you are not going to understand most of the book. The information presented is at the code level with snippets of code scattered throughout the book.

After a brief introduction covering Linux in general and compiling the kernel, the book covers important data structures, like the process table, followed by inodes, memory management and timers. Next, how signals, interrupts, booting, the scheduler and system calls work are presented. There is even a section on how to implement a new system call.

The next five chapters deal with specific pieces of the system: memory management, inter-process communication, the file system, device drivers and network implementation. This information is presented in enough detail to clarify the process and enable the reader to write compatible code. The chapter on the file system includes the proc and ext2 file systems.

The final chapter in the main part of the book is on modules, describing what modules are and how they are implemented. The text presents an example module (the PCMCIA card handler), and explains how to debug modules.

The book ends with five appendices. The first details the system calls in much the same way as Section 2 of the man pages, but the calls are sorted by area (processes, file system, etc.) rather than alphabetically. It also tells you where the file with the code to implement the function is located.

The second appendix discusses what the authors call “kernel-related commands”. These include free, ps, rdev, top, init, shutdown, strace and mount as well as network, serial and parallel interface configuration.

The third appendix discusses the proc file system. Many people don't realize how important the proc file system is. This short (ten page) chapter explains it and the important information that can be secured from it.

The final appendix explains the boot process, including the MS-DOS boot sector and partition table, and discusses LILO with an explanation of the LILO startup errors (e.g., when you just get LI and the system dies).

The final appendix presents “useful kernel functions”. This is an explanation of functions, such as printk(), available to users writing kernel code.

## What's Wrong with the Book?

Very little. I think it does an excellent job of covering the material. It doesn't attempt to cover subjects outside its primary focus, and I find that a plus.

The only shortcoming I see is that it covers version 1.2 kernels, not 2.0. Of course, at this time, it would be very difficult to have a book that covers 2.0. Addison-Wesley may decide to update the book before the next printing.

Bottom line, I highly recommend this book for anyone who is serious about writing kernel code or who wants to know what is in the Linux kernel.

Phil Hughes is the publisher of *Linux Journal*.

Archive Index Issue Table of Contents

Advanced search

# Consultants Directory

> This is a collection of all the consultant listings printed in *LJ* 1996. For listings which changed during that period, we used the version most recently printed. The contact information is left as it was printed, and may be out of date.

**ACAY Network Computing Pty Ltd**
Australian-based consulting firm specializing in: Turnkey Internet solutions, firewall configuration and administration, Internet connectivity, installation and support for CISCO routers and Linux.

Address:
Suite 4/77 Albert Avenue, Chatswood, NSW, 2067, Australia
+61-2-411-7340, FAX: +61-2-411-7325
sales@acay.com.au
http://www.acay.com.au

**Aegis Information Systems, Inc.**
Specializing in: System Integration, Installation, Administration, Programming, and Networking on multiple Operating System platforms.

Address:
PO Box 730, Hicksville, New York 11802-0730
800-AEGIS-00, FAX: 800-AIS-1216
info@aegisinfosys.com
http://www.aegisinfosys.com/

**American Group Workflow Automation**
Certified Microsoft Professional, LanServer, Netware and UnixWare Engineer on staff. Caldera Business Partner, firewalls, pre-configured systems, world-wide travel and/or consulting. MS-Windows with Linux.

Address:
West Coast: PO Box 77551, Seattle, WA 98177-0551
206-363-0459
East Coast: 3422 Old Capitol Trail, Suite 1068, Wilmington, DE 19808-6192
302-996-3204
amergrp@amer-grp.com
http://www.amer-grp.com

**Bitbybit Information Systems**
Development, consulting, installation, scheduling systems, database interoperability.

Address:
Radex Complex, Kluyverweg 2A, 2629 HT Delft, The Netherlands
+31-(0)-15-2682569, FAX: +31-(0)-15-2682530
info@bitbybit-is.nl

**Celestial Systems Design**
General Unix consulting, Internet connectivity, Linux, and Caldera Network Desktop sales, installation and support.

Address:
60 Pine Ave W #407, Montréal, Quebec, Canada H2W 1R2
514-282-1218, FAX 514-282-1218
cdsi@consultan.com

**CIBER*NET**
General Unix/Linux consulting, network connectivity, support, porting and web development.

Address:
Derqui 47, 5501 Godoy Cruz, Mendoza, Argentina
22-2492
afernand@planet.losandes.com.ar

**Cosmos Engineering**
Linux consulting, installation and system administration. Internet connectivity and WWW programming. Netware and Windows NT integration.

Address:
213-930-2540, FAX: 213-930-1393
76244.2406@compuserv.com

**Ian T. Zimmerman**
Linux consulting.

Address:
PO Box 13445, Berkeley, CA 94712
510-528-0800-x19
itz@rahul.net

**InfoMagic, Inc.**
Technical Support; Installation & Setup; Network Configuration; Remote System Administration; Internet Connectivity.

Address:
PO Box 30370, Flagstaff, AZ 86003-0370

602-526-9852, FAX: 602-526-9573
support@infomagic.com

**Insync Design**
Software engineering in C/C++, project management, scientific programming, virtual teamwork.

Address:
10131 S East Torch Lake Dr, Alden MI 49612
616-331-6688, FAX: 616-331-6608
insync@ix.netcom.com

**Internet Systems and Services, Inc.**
Linux/Unix large system integration & design, TCP/IP network management, global routing & Internet information services.

Address:
Washington, DC-NY area,
703-222-4243
bass@silkroad.com
http://www.silkroad.com/

**Kimbrell Consulting**
Product/Project Manager specializing in Unix/Linux/SunOS/Solaris/AIX/HPUX installation, management, porting/software development including: graphics adaptor device drivers, web server configuration, web page development.

Address:
321 Regatta Ct, Austin, TX 78734
kimbrell@bga.com

**Linux Consulting / Lu & Lu**
Linux installation, administration, programming, and networking with IBM RS/6000, HP-UX, SunOS, and Linux.

Address:
Houston, TX and Baltimore, MD
713-466-3696, FAX: 713-466-3654
fanlu@informix.com
plu@condor.cs.jhu.edu

**Linux Consulting / Scott Barker**
Linux installation, system administration, network administration, internet connectivity and technical support.

Address:
Calgary, AB, Canada
403-285-0696, 403-285-1399
sbarker@galileo.cuug.ab.ca

**LOD Communications, Inc**
Linux, SunOS, Solaris technical support/troubleshooting. System installation, configuration. Internet consulting: installation, configuration for networking hardware/software. WWW server, virtual domain configuration. Unix Security consulting.

Address:
1095 Ocala Road, Tallahassee, FL 32304
800-446-7420
support@lod.com
http://www.lod.com/

**Media Consultores**
Linux Intranet and Internet solutions, including Web page design and database integration.

Address:
Rua Jose Regio 176-Mindelo, 4480 Cila do Conde, Portugal
351-52-671-591, FAX: 351-52-672-431
http://www.clubenet.com/media/index.html/

**Perlin & Associates**
General Unix consulting, Internet connectivity, Linux installation, support, porting.

Address:
1902 N 44th St, Seattle, WA 98103
206-634-0186
davep@nanosoft.com

**R.J. Matter & Associates**
Barcode printing solutions for Linux/UNIX. Royalty-free C source code and binaries for Epson and HP Series II compatible printers.

Address:
PO Box 9042, Highland, IN 46322-9042
219-845-5247
71021.2654@compuserve.com

**RTX Services/William Wallace**
Tcl/Tk GUI development, real-time, C/C++ software development.

Address:
101 Longmeadow Dr, Coppell, TX 75109
214-462-7237
rtxserv@metronet.com
http://www.metronet.com/~rtserv/

**Spano Net Solutions**
Network solutions including configuration, WWW, security, remote

system administration, upkeep, planning and general Unix consulting. Reasonable rates, high quality customer service. Free estimates.

Address:
846 E Walnut #268, Grapevine, TX 76051
817-421-4649
jeff@dfw.net

**Systems Enhancements Consulting**
Free technical support on most Operating Systems; Linux installation; system administration, network administration, remote system administration, internet connectivity, web server configuration and integration solutions.

Address:
PO Box 298, 3128 Walton Blvd, Rochester Hills, MI 48309
810-373-7518, FAX: 818-617-9818
mlhendri@oakland.edu

**tummy.com, ltd.**
Linux consulting and software development.

Address:
Suite 807, 300 South 16th Street, Omaha NE 68102
402-344-4426, FAX: 402-341-7119
xvscan@tummy.com
http://www.tummy.com/

**VirtuMall, Inc.**
Full-service interactive and WWW Programming, Consulting, and Development firm. Develops high-end CGI Scripting, Graphic Design, and Interactive features for WWW sites of all needs.

Address:
930 Massachusetts Ave, Cambridge, MA 02139
800-862-5596, 617-497-8006, FAX: 617-492-0486
comments@virtumall.com

**William F. Rousseau**
Unix/Linux and TCP/IP network consulting, C/C++ programming, web pages, and CGI scripts.

Address:
San Francisco Bay Area
510-455-8008, FAX: 510-455-8008
rousseau@aimnet.com

**Zei Software**
Experienced senior project managers. Linux/Unix/Critical business software development; C, C++, Motif, Sybase, Internet connectivity.

Address:
2713 Route 23, Newfoundland, NJ 07435
201-208-8800, FAX: 201-208-1888
art@zei.com

Archive Index Issue Table of Contents

Advanced search