Linux Journal Issue #92/December 2001

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

Linux System Administration: A User's Guide

Marcel Gagné

Issue #92, December 2001

An excerpt from our French chef's recently published book.

By the time you get this issue of *Linux Journal*, my new book, *Linux System Administration: A User's Guide*, will have hit the shelves. Because they are just so darn cool, the folks at *Linux Journal* have given me some space in this month's issue to give you a teaser from the book.

Being the kind of guy that I am, I ached over what to present while trying to keep it short. After all, there's so much to choose from. Setting up a scanner? A CD writer? Blocking unwanted access to your network? Printing? Backups? Sendmail tweaks? What? In the end, I decided to talk about something that doesn't get enough press in today's internet world, and that's good old-fashioned user security. What you are about to read is an excerpt from Chapter Seven of the book, “Users and Groups”. It is not the complete chapter, nor are all the pieces in order. “The chef has been sipping too much of his own wine”, you say? Not at all. In fact, like my alter ego, I wanted to use this space to give you a sample, a taste if you will, of what you might find in the pages of my new book. Think of it as nibbling from a buffet table. As Chef Marcel would say, *Bon appétit*!

Living in a Multiuser World

Linux is a multiuser operating system, meaning that one or more users can work on it at the same time. Every user is referenced by their user name. Each user name has a user ID (UID) associated with it and one or more groups. Like user names, group names are also represented by a numeric identifier, this time called a group ID (GID). A user's UID is unique as is a group's GID.

When it comes to your files and directories, security on a Linux system is defined by means of permissions, which directly relate to the user ID. Users are either administrative users or regular users. The chief administrative user is

called root. A user's ID is used to decide what commands can be executed and what files can be read from or written to.

Each user ID also has a password associated with it. That password can and should be changed on a regular basis.

When Not to Use the Root User

The short answer here is that you should never use the root user unless you absolutely have to. The danger lies in the fact that the root user is virtually omnipotent on the system. A mistake can have serious implications that can wipe out your entire system. Unless you absolutely have to, it is best to work as a nonadministrative user. There are other reasons as well.

The first is security. Because the root user has access to everything, it makes sense that only those who really need to have access are given the root password. The fewer people who have access to root, the better. Let me give you a few good reasons for jealously guarding root access: it makes it easier to maintain security, it decreases the risk of dangerous code and errors do not (usually) have global implications. Yes, it is still possible for a nonroot user to do great damage to a system, but the risk is much, much smaller.

Checking the Password File

As if you didn't already have enough things to do, here's yet another job. On a regular basis, you should run a report to identify accounts or logins that have gone dormant. This is a nice way of saying “people who are gone and who are not coming back anytime soon”.

An earlier chapter of the book explains the finger command, which displays information relating to the last time an account was used. Try typing this command to list each user ID and to check the last login time. Note that the single quotes (just before the sort command and just before the pipe symbol) are actually back quotes (or back ticks). The back quote is usually found with the tilde (~) just under the Esc key on most keyboards.

```
finger 'sort /etc/passwd | cut -f1 -d":"' | grep -i log | more
```

The output of this command looks something like this:

```
Login: aeinstein                Name: A. Einstein
Never logged in.
Login: guitux                   Name: Tux the Penguin
Last login Mon Jan  8 14:54 (EST) on tty2
Login: halt                     Name: halt
Never logged in.
Login: lp                       Name: lp
Never logged in.
Login: mail                     Name: mail
Never logged in.
```

```
Login: mgag                     Name: Marcel Gagné
Last login Wed Mar  7 17:29 (EST) on 1 from website
Login: named                    Name: Named
Never logged in.
Login: natika                   Name: Natika the Cat
```

Warning: you must use your good judgment (an absolute requirement for system administration) on this one. Some of these accounts—sync and lp, for instance—are system accounts. It only makes sense that no one will have ever logged in through them. On the other hand, Mr. Einstein (at the top of the list) has never logged in either and his is certainly not an administrative login. It could be that this is a new account (and it is) or that you created an account for a user that never was used. In the latter case, you should probably get rid of that account.

Now, I used that example to give you a feel for your command-line prowess. However, I should tell you that there is a cleaner way to do this. Your Linux system comes with a handy little command called lastlog that does just this sort of thing:

```
[root@scigate /root]# lastlog | more
Username  Port  From             Latest
root      tty1                   Wed Mar  7 17:18:40
                                          -0500 2001
bin                              **Never logged in**
daemon                           **Never logged in**
adm                              **Never logged in**
lp                               **Never logged in**
sync                             **Never logged in**
shutdown                         **Never logged in**
mgagne    1     scigate          Wed Mar  7 17:29:55
                                          -0500 2001
postgres                         **Never logged in**
www                              **Never logged in**
natika    8     localhost.locald Thu Dec  7 14:30:15
                                          -0500 2000
guitux    tty2                   Mon Jan  8 14:54:55
                                          -0500 2001
```

Geek trivia: you can't edit or modify this file, but the lastlog command information comes from the file /var/log/lastlog.

Here is another thing you should do. Every once in a while, run the command pwck. By default, it walks through your /etc/passwd and /etc/shadow files and does some basic integrity checks, such as making sure that the right number of fields are present and that each name is uniquely identified. For the group file, there is a companion command called grpck.

How Crackers Crack Your Passwords

The reason for a good password goes right back to my description of the password file earlier in the book, specifically as it relates to the password field in nonshadow files. Here's a quick reminder of the format:

```
root:2IsjW45pb4L56:0:0:root:/root:/bin/bash
```

The password field (field 2) is encoded by virtue of a hashing algorithm. If you are curious as to the gory details, type <@cxb>man crypt<@$p> and you'll find everything you ever wanted to know about encoding passwords. The short form is this: that strange password is actually a coded version of your password based on a two-character, randomly generated salt. This salt is then used to seed the hashing routine to generate the final group of characters.

The term hashing represents a technique for taking a string of characters (a person's last name, for instance) and generating a unique key (ideally) for easy retrieval of the information from a database. What you are doing is encoding the normal text into a shorter, (usually) numeric representation.

Password crackers figure out passwords by using that salt to generate passwords against every word in the dictionary. While this sounds pretty complex, it's not. A simple program calls the crypt routine, runs the hash on a word and then compares it to the password entry in the /etc/passwd file. If it matches, bingo! They have your password. If it doesn't, they move on to the next word. On a reasonably punchy system, it doesn't take all that long for crackers to work their way through every password in the book.

Don't believe me? Take a look at the output in Figure 1 from a little program called Nutcracker, a freeware tool that does the kind of brute-force password checking I was talking about.

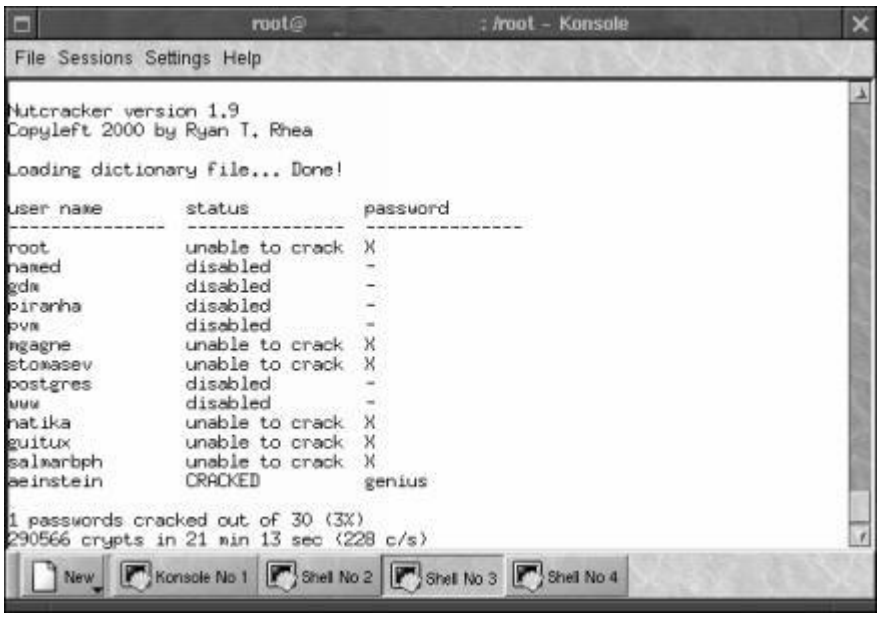


Figure 1. Why Dictionary Wofrds Make Bad Passwords

As you can see in Figure 1, picking something you'll remember easily because it is a common word is a bad choice for a password.

I Logged in from Where?

Have a look at what happens when I log in to a machine. Everything looks normal. I have a login name, a request for my password. I enter the password and *voilà*, I am in. But hold on—read that little one-line message that appears after I enter the password:

```
login: mgagne
Password:
Last login: Mon Jan  8 16:00:39 from energize
```

What the heck is “energize”? Energize is the hostname of the computer from which I last logged in apparently, except I don't have a system called energize. Furthermore, let's pretend that I don't know anyone with that system and I always log in from the same place. The only explanation is that somebody from a system called energize logged in to the server with my login name and password.

This is just a hypothetical situation, but it does illustrate one other habit that you should consider training your users to adopt. If they are logging in from the same PC day in and day out, that message should never change. If they do not recognize the hostname in the last login message, they should make it a policy to alert you.

Security isn't just the domain of the system administrator. After all, you've got plenty on your hands. Any help is appreciated. You need to get the users involved. Let them know that system security is their business as well as yours.

Resources

email: mggagne@salmar.com

Marcel Gagné (mggagne@salmar.com) is president of Salmar Consulting, Inc., a systems integration and network consulting firm and the author of Linux System Administration: A User's Guide, published by Addison-Wesley.

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

Building an E-mail Virus Detection System for Your Network

Dave Jones

Issue #92, December 2001

The best way to stop the spread of e-mail-based viruses is not to let them into your system.

One of the most threatening and irritating problems I deal with as a network administrator is viruses. It seems that the virus threat has increased tenfold over the last couple of years. I have found it increasingly unnerving to see what the newer viruses are capable of doing. It's not even their ability that is most disturbing, it's the ease with which they get into our systems. Recently I spent two hours cleaning the MTX and Navidad viruses off of a user's machine.

The usual first approach that administrators take in virus prevention is to install a desktop virus protection program. That is wise, but it seems to me the most fail-safe way to protect a corporate network from viruses is to prevent their entrance into the system in the first place. By far the most common entry point for viruses, especially macro viruses, is corporate e-mail systems. However, this is usually the most neglected piece in any virus detection solution. The e-mail virus software on the market today is often application-specific, expensive or both (not to mention unreliable). Being a mid-sized business, we did not have it in our budget this year to buy an antivirus suite off the shelf, so I turned to Linux and open source. I did find a couple of interesting projects on the Internet that might meet our needs, but I decided to write my own instead. I wanted our system to be extremely easy to follow and easy to extend without a user having to know C or be a Perl guru. Also, I wanted the system to take advantage of the power of the utilities that every base Linux installation usually has. All of this would ensure that another could administer the system in my absence and that it would be portable.

The basic outline of the system consists of using Bash scripts, metamail, grep, the Obtuse Systems' smtpd product, Samba and a command-line virus scanner.

A flowchart-style diagram can be found in Figure 1. The Obtuse Systems' SMTP store and forward package is freely available at www.obtuse.com/smtpd.html. The current version as of this writing is version 2.0. The virus scanner I chose was McAfee Virus Scan for UNIX/Linux, but there are quite a few others to choose from. Some are free and some are not. Do make sure you choose one that sets exit status codes based on what it finds and that is well supported with frequent signature updates.

The system can be set up on an existing Linux firewall or a separate machine, if you do not already have a Linux firewall in place. If you choose to set up a separate machine as the e-mail firewall, it doesn't have to be very powerful. A 200MHz 586 with 32MB of RAM would be plenty. Our network is attached to the Internet via SDSL and is protected by a Mandrake Linux machine running IP masquerade. This design made it easy to set up the system on our current firewall machine. The internal e-mail system used is not important as long as it speaks SMTP or ESMTP. In our case, we use Novell's Groupwise product. All SMTP traffic (port 25) should be redirected from the SMTP port on the firewall to the machine you have set up as your e-mail firewall on the inside (or to the firewall itself in our case). Now let's move on to the actual setup.

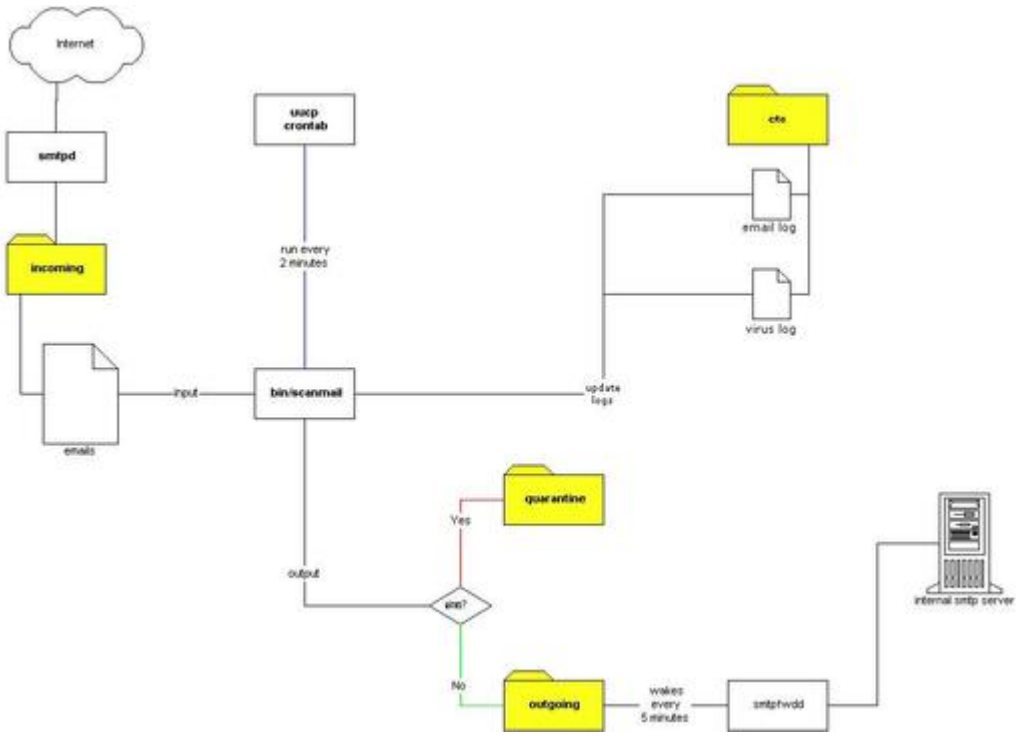


Figure 1. Setup of Network Traffic and Firewall

Our first step is to set up the filesystem hierarchy. A diagram of this can be found in Figure 2. We will set up our system in a directory named /var/spool/smtpd. If you have an average e-mail volume of over 25,000 e-mails a day, I suggest partitioning a separate hard drive dedicated to the e-mail firewall and mounting it in this directory. The base directory will be /var/spool/smtpd. In this

directory we will make five subdirectories named incoming, outgoing, etc, bin and quarantine. To start, become root and issue this command:

```
mkdir -p /var/spool/smtpd/ {etc,bin,incoming,outgoing,quarantine}
```

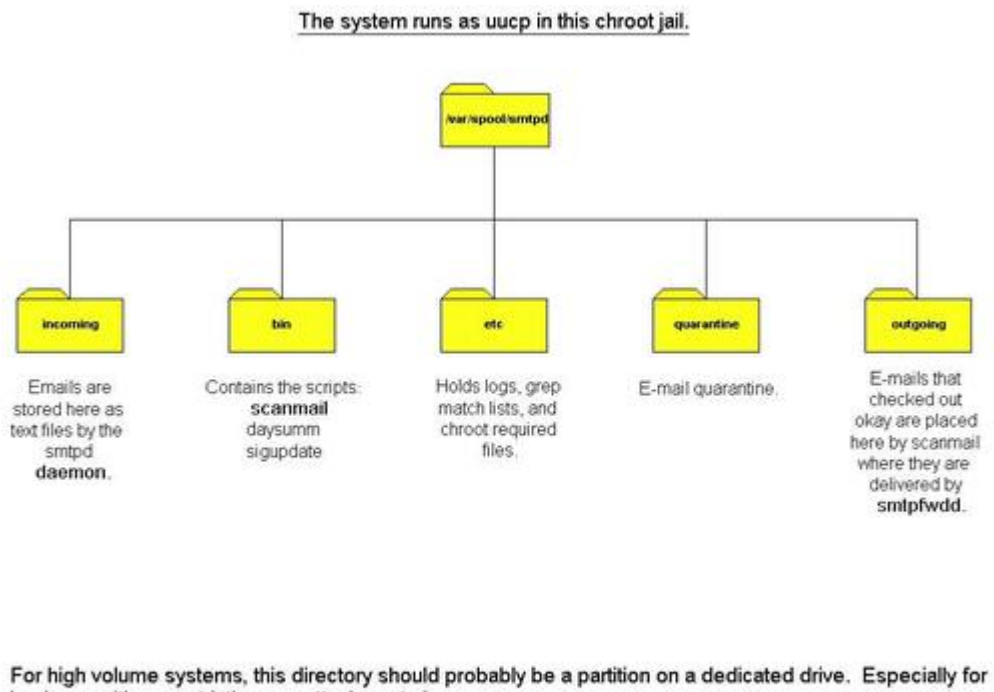


Figure 2. Filesystem Hierarchy

Next, change the permissions on the whole hierarchy to be accessible only by the uucp user, since all the programs will be run as this user. These commands will do the trick:

```
chown -R uucp.uucp /var/spool/smtpd
chmod -R 700 /var/spool/smtpd
```

Now we will set up the first component of the system. You will need to grab the smtpd package from the Obtuse Systems' web site mentioned earlier. Switch to the download directory and unpack the tarball by issuing the command:

```
tar -zxvf smtpd-2.0.tar.gz
```

Now change to the smtpd-2.0 directory and edit the Makefile to reflect the following changes:

```
SPOOLDIR = /var/spool/smtpd
SPOOLSUBDIR = incoming
POLL_TIME = 300
PARANOID_SMTP = 1
JUNIPER_SUPPORT = 0
CHECK_IDENT = 0
```

What we want is for smtpd to store mail in the incoming subdirectory and smtpfwdd to read mail from the outgoing directory. To make this possible we have to insert a line into the file smtpfwdd.c. Insert the following two lines at line number 75:

```
// Pull mail from the outgoing subdir.
#define SPOOLSUBDIR "outgoing"
```

Finish up by compiling and installing the package with these commands:

```
make
make install
```

Next, we need to populate the /var/spool/smtpd/etc directory with some files to allow smtpd to operate correctly. Copy the resolv.conf file from /etc to the /var/spool/smtpd/etc directory, and then copy the file localtime from /etc to here. You also should copy the antirelay_check_rules_example file from the smtpd-2.0 distribution directory to /var/spool/smtpd/etc and rename it smtpd_check_rules. You can look on the Obtuse Systems' web site for instructions on how to create check rules if you need this. I would at least put in an antirelay rule to start. To have the smtpd program start automatically, place an entry in your /etc/inetd.conf file similar to this:

```
smtp    stream tcp nowait root /usr/local/sbin/smtpd    smptd
```

This line should replace any other SMTP entries that may be there. We will need to start smtpfwdd manually from /etc/rc.d/rc.local (or whatever your local rc file is). So go ahead and add an entry like this to your startup file:

```
### Start the smtpfwdd forwarding daemon
/usr/local/sbin/smtpfwdd
```

Finally, you will need to shut down any other mail transfer agent (MTA) that might be running. These would include Postfix, sendmail, qmail and the like. On Red Hat systems you can do this simply by running the setup utility, unchecking any MTAs from the system services menu and rebooting. Be aware that some MTAs run as children of other processes, such as Postfix, and simply can't be killed directly. If you can reboot the machine, then go ahead and do that now. If not, then you can issue these commands to get smtpd and smtpfwdd up and running:

```
kill -HUP inetdpid
/usr/local/sbin/smtpfwdd
```

where *inetdpid* is the Process ID for inetd.

Once the smtpd and smtpfwdd dæmons are running, you can test the setup by launching a Telnet session on port 25 (the smtp port) of the e-mail firewall, like so:

```
telnet email.firewall.com 25
```

where *email.firewall.com* is the hostname of your e-mail firewall.

You should get a prompt that says

```
220 email.firewall.com SMTP ready,
Who are you gonna pretend to be today?
```

If you get any other prompt, then you probably forgot to shut down a running MTA on the server. A **ps -e** should bear this out.

Let's look at what we have so far. At this point you should have a machine with the smtpd dæmon running and accepting e-mail. All received e-mail is being stored in the /var/spool/smtpd/incoming directory as simple ASCII text files. To test this out, try issuing these commands:

```
telnet email.firewall.com 25
helo firewall.com
mail from: joe@firewall.com
rcpt to:
data
This is a test.
.
quit
```

where *you@domain.com* is your e-mail address. Be sure to end your message body with a period on a line all by itself. This tells the server you are through sending text.

If all went well, you should now see a text file in the /var/spool/smtpd/incoming directory containing the contents of our SMTP session. Now move the file into the /var/spool/smtpd/outgoing directory. In a few minutes the file should disappear, and you should receive the e-mail in your main mailbox. **smtpd** saves e-mails as text files named smtpd*??????*, where *??????* is a randomly generated message ID. **smtpfwdd** reads these text files and forwards them to the destination server. After a file is successfully forwarded, it is deleted.

So how does the mail get from incoming to outgoing? Here is where our e-mail scanning script will come into play. The script will search through each e-mail file in the incoming directory one by one and look for viral content. If the file doesn't contain a virus, it is moved to the outgoing directory. If it does, then it gets moved to the quarantine directory. It's as simple as that. But before the mail scanning script will work, we need to have a functional piece of virus detection software installed. Let's focus on that now.

What we need is a command-line based virus scanner. I chose McAfee Virus Scan for UNIX/Linux, so that is what I describe here. The McAfee product has a intuitive list of exit status codes, so it is easy to integrate into shell scripts. You can get this product by going to www.nai.com and using the e-store. After you

install the product you will need to make sure it is executable by uucp. To do this, go to the /usr/local/uvscan directory and execute this command:

```
chmod -R 755 *
```

Now test it out and make sure it works. Become uucp and issue this command:

```
/usr/local/uvscan/uvscan --version
```

If the test is successful, then we can move on to updating the virus signature files. The signature files (or definition files as they are sometimes called) are the lifeblood of any virus scanner. Therefore, we will want to automate the process of updating them on a frequent basis. It is important to check for these updates at least biweekly, since new versions usually are put out at least three times a month. To do this we must first create a Bash script called sigupdate and place it in the /var/spool/smtpd/bin directory. As always, make sure the script is owned and executable by the uucp user. The contents of the script are shown in Listing 1 and should be easy to follow. The sigupdate file should be set to run once a week, preferably at a low-volume time of day. We will do this later by inserting a crontab entry for it. You also will need to create a file in the uucp user's home directory called .netrc. Edit this file to look as follows:

```
machine ftp.nai.com
login anonymous
password admin@domain.com
macdef init
cd pub/antivirus/datfiles/4.x
bin
prompt
mget dat-*.tar
close
bye
```

Listing 1. sigupdate Script

The .netrc file controls sessions for predefined FTP hosts. This is the best way to make our FTP session automatic. You can reference the FTP man page for help with the netrc syntax. Go ahead and run sigupdate as soon as it, and .netrc, are ready. After the update finishes, execute this command:

```
/usr/local/uvscan/uvscan --version
```

Look at the virus data file creation date; it should be recent, usually no more than a month old. If it is not, you will need to check that sigupdate is running properly from the command line.

Now let's move on to the main mail scanning script. We will name it scanmail, and it will be stored in the /var/spool/smtpd/bin directory. This script will do all of the direct manipulation of the e-mail text files created by smtpd. Create the file and make it owned and executable by uucp. You will find the complete script in Listing 2 [available at ftp.linuxjournal.com/pub/lj/listings/

issue92/4882.tgz]. It is heavily commented so I will just give you an overview here.

Starting on line 19, scanmail first switches to the incoming directory and stores the e-mail filenames in an array variable. It then loops through each filename and uses grep to search for specific patterns within the files. Two passes are made through each e-mail. The first pass greps for attachments that are obviously bad. The patterns for this pass are stored in a file named matches.bad, which we will create later. If grep hits a match, that file is quarantined and an e-mail is sent to the administrator containing a timestamp, the name of the file, who it came from and who was to receive it.

If no match was made, a second pass is made on the e-mail. This time grep uses a file named matches.doc to search for attachments that could contain macro viruses or embedded viruses. If a match is made this time, the attachment is decoded into a dynamically generated temporary directory using the metamail program. The temporary directory is created by using the name of the e-mail with a "_d" concatenated onto the end. The contents of the temp directory are then scanned with our command-line virus scanner. If a virus is detected here (by checking the exit status of the scanner), scanmail quarantines the e-mail and the attachment(s) and sends the administrator an e-mail alert. A courteous e-mail also is sent to the sender saying that they may need to check their system for viruses and giving the name of the virus detected.

At this point, if no viruses have been found, the e-mail is moved over to the outgoing directory where it is delivered to the internal e-mail server by smtpfwdd. **smtpfwdd** scans the outgoing directory for mail to deliver every five minutes.

The next thing to do is set up our filename lists that scanmail will use to search for suspicious attachments. **scanmail** uses the grep tool to search through files. We will take advantage of the -f switch to make grep pull a list of patterns from a specific text file. The layout of the text file is simple, only one pattern per line. **grep** will then match any pattern listed in the file. Switch to the /var/spool/ smtpd/etc directory and create two files called matches.bad and matches.doc. In the matches.bad file we list all the filename patterns that we definitely don't want to come into our network without being inspected by the administrator.

The matches.doc file, on the other hand, should contain filename patterns for documents that may contain embedded viruses, such as Word documents or spreadsheets. When creating these files, use the form **filename=.*\.exe** for each line. This is so you will not get false alarms because of random strings in the mime encoding that happen to match what grep is scouting. Also, be sure not to have any blank lines in the file because grep will interpret a blank line as a

pattern to be searched for and will match all e-mails. Vim is a good editor to use for creating these files since you easily can see any blank lines. You can find the contents of the files I use in Listing 3.

Listing 3. grep Pattern Files

The other script we will use is called daysumm and resides in /var/spool/smtpd/bin. Create this file as shown in Listing 4. **daysumm** sends a report of the day's e-mail activity to the administrator. It shows how many e-mails were received that day, how many viruses were detected and which viruses they were. This file should be set up in cron to run at 11:59 **P.M.** each evening. The daysumm script relies on the virus.$date and email.$date files in /var/spool/smtpd/etc. These are text files that are created dynamically, updated by scanmail and are date-dependent. For this reason, daysumm must be run before midnight or the timestamp will change and the wrong files will be read.

Listing 4. daysumm Script

If you will notice, each time we send an e-mail from the firewall itself, such as from a script, we follow it up with the command **sendmail -q**. The -q switch tells sendmail to start up, check for any outgoing mail, send it and then exit. It effectively flushes all local outbound sendmail queues. This is necessary since there is no MTA running on the machine anymore. The smtpd package is not an MTA but a store-and-forward agent only. It can be thought of as a dedicated mail relay program. Without this command, you would never get any mail that originated on the e-mail firewall itself.

We should now automate the whole process via the cron dæmon. We will do this by way of the uucp user's personal crontab file. Make sure you are logged on as uucp and issue the command **crontab -e**. This will open up uucp's cron table for editing. Make the following entries for the scanmail, daysumm and sigupdate scripts:

```
MAILTO=""
# Run the scanmail script every two minutes
*/2 * * * * /var/spool/smtpd/bin/scanmail
# Run the daysumm script every night at 11:59PM
59 23 * * * /var/spool/smtpd/bin/daysumm
# Run sigupdate every Thursday at 4:00PM
0 16 * * 4 /var/spool/smtpd/bin/sigupdate
```

You can adjust the runtimes as necessary to suit your needs. The main consideration as to how often to run scanmail is your average daily e-mail volume. If you have a daily volume of over 10,000, I would set the interval to every two minutes and set smtpfwdd to run every five minutes. This is so you don't send huge batches of e-mail to your internal server all at once. If you have a volume of around 1,000 a day or less, then scanmail could be run every ten

minutes and smtpfwdd could scan every 20 minutes. Also, don't leave out the **MAILTO=""** entry at the top of the crontab. This is so crond will not send the uucp user any mail about completed cron jobs. E-mail every two minutes will add up quickly, and uucp never checks its own mail.

I also have a Samba share set up to allow me access to the /var/spool/smtpd hierarchy from my Windows machine. This is helpful to administrators that mainly use Windows. It keeps me from having to open up an ssh session each time I need to check a virus warning message. You can do that by adding an entry such as this to your /etc/smb.conf file:

```
[mail-gate]
   Comment = Email firewall directories
   Path = /var/spool/smtpd
   Valid users =
   Admin users =
   Browseable = No
   Read Only = No
```

where *you* is your Samba user name.

The final step is to divert all incoming SMTP connections from your firewall to your new scanning server. If your firewall uses IP masquerade, this is an easy task. Simply execute the command:

```
ipmasqadm portfw -a -P tcp -L firewall 25 -R destination 25
```

where *firewall* is the address of the firewall and *destination* is the address of your new scanning server.

If you have decided to run the e-mail firewall directly on your existing firewall, then you don't need to make any changes. If you have another firewall product, you will need to read the documentation to figure out how to do port redirection. It should not be too much trouble, but you may need to contact the maker of your product to find out. Be sure to put this redirection command in your startup scripts so that they will persist through rebooting.

If everything went well, you should now have an operational e-mail-virus firewall. Try sending yourself some test messages from a free web e-mail account to make sure everything is in working order. I periodically send myself macro-virus infected documents to make sure the system is still functioning correctly. I can think of many ways to expand this basic system to do other things. Especially promising is the daysumm script, which could be enhanced greatly. I currently am working on a CGI script that will report current statistics, such as average daily volume, in an intranet page. That is only one of a hundred ways this system could help you manage and safeguard your e-mail system with little or no impact on the company budget. If anyone comes up with some cool ways to expand the system, please let me know. I'd love to hear about it.

email: davejones@pearcebevill.com

Dave Jones has been a network administrator in Birmingham, Alabama for three years. When he is not in front of the computer, he can be found enjoying a pipe of tobacco or watching X-Files with his wife and daughter. He can be reached at davidashleyjones@hotmail.com.

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

Guard Against Data Loss with Mondo Rescue

Hugo Rabson

Issue #92, December 2001

Hugo shows how to use Mondo to restore your system in the event of catastrophic data loss.

Every computer user has lost data at some point. Unfortunately, the experience rarely teaches the user to backup his or her data regularly. Mondo Rescue creates one or more bootable Rescue CDs based on your filesystem. In the event of catastrophic data loss, you will be able to restore your entire system. Mondo supports Lin/Win dual-boot systems, RAID, ReiserFS, ext2, ext3, XFS, JFS and VFAT.

I wrote Mondo because no other program did what I wanted, nor was any program as easy to use. I back up, wipe, format and restore my own system at least three times a week using Mondo's latest release. Mondo is the only backup regime that I use.

Background

Mondo is a thriving open-source project; it can restore your OS and data from bare metal, and data loss has not been reported since May 2000. The project is coming along nicely.

In the beginning, Mondo was created to back up Windows-only systems and provide disaster recovery facilities for Windows users. While running a small computer store near Nashville, Tennessee, I realized that many of the people that bought computers from us were bringing them back to have the disks reformatted and the applications re-installed. Was this because they did not know how to maintain their own PCs? Was our hardware to blame? Either way, we needed a way to restore a fresh copy of Windows onto a typical PC, unattended, in 10-15 minutes. We were using Norton Ghost at the time, but Ghost didn't let us restore selectively; it was all or nothing. Also, Ghost tended to crash when copying Linux partitions.

Our idea was to run Linux on a CD but have Windows archived in big tarballs on said CD. Well, to cut a long and tortuous story short, it worked. I even managed to write code that would create a Windows-bootable VFAT partition.

In March 2001, an employee of Hewlett-Packard (Grenoble) became interested in the project. Bruno Cornec, a skilled programmer in the server division, began to look at the code. He started to play with it and improved it in numerous ways. Between September 2000 (when the project was shut down) and March 2001, two new versions of Mondo were released. Both were made possible almost entirely by Bruno Cornec, Maciej Kulasa and other contributors.

In June 2001, I built a new computer and got to work. By July 4, I had fixed all the major bugs, and Mondo v1.00 was released (no jokes about Independence Day, please).

How It Works

Mondo is just a bit shell script that wraps around two other tools: Mindi and afio.

Mindi, aka Mindi-Linux, is a mini-distribution that generates boot disks from your kernel, modules, tools and libraries. It is more likely than a generic boot disk to be binary-compatible with your existing software because its tools are copied straight from your hard disk. It also can generate an El Torito 2.88MB boot disk image. Mondo uses Mindi to create a 2.88MB boot “disk” and associated data “disks”, which are incorporated in each Mondo CD. Whichever modules were loaded at backup time are reloaded at boot time. So, in theory you will boot into almost the same environment you were in when you backed up.

Mondo-Archive works something like this:

1. It squeezes your files into several tarballs, each 5-10MB in size.
2. It puts those tarballs in a directory.
3. It puts an El Torito boot floppy image and associated data files in that directory.
4. It points mkisofs at that directory and pipes its output to cdrecord, which causes a CD to be burned that contains a copy of that directory and its contents.
5. It repeats the above steps for *N* CDs, *N* being the number of CDs it takes to accommodate all your files.

Kernel Configuration

Mindi v0.38 comes with a 2.4.7 fail-safe kernel, in case your kernel does not support all the features required by a boot disk. This is not Mondo being picky; this is Linux being awkward. Some kernels are just not suitable for boot disks. Novice users should stick with their default kernel and make Mindi use its own kernel by saying no when asked if they want to make a boot disk with their own kernel.

Advanced users should make sure that their kernel includes the following, built-in: CD-ROM support, IDE CD support, IDE support, initrd and RAM disk support, floppy disk drive support, stable loopfs support (which means it really needs to be 2.2.17 or later, or 2.4.5-ac10 or later) and ISO9660 support.

In case your kernel does not support these features, you should probably use Mindi's own kernel until you are confident enough to build your own kernel. If you find that Mindi's kernel does not support something you need (e.g., XFS) then please let me know. I will probably add the feature to the next release of Mindi.

Getting and Installing Mondo

To install the program, go to www.microwerks.net/~hugo and download Mondo and Mindi. The latter is part of the former but was forked because Mindi also creates standalone boot disks based on your kernel, modules, tools and libraries. Installation instructions for both tools are provided on the Download web page.

RPM users have it easy; they simply need to download Mindi's RPM to /tmp, download Mondo's RPM to /tmp and then type the following:

```
rpm -Uvh /tmp/mondo-1.13-1.i386.rpm /tmp/mindi-0.39-1.i386.rpm
```

Tarball users have a slightly harder time; they must download Mindi's tarball to /tmp, download Mondo's tarball to /tmp and then type the following:

```
cd /tmp
tar -zxvf mindi-0.39.tgz
cd mindi-0.39
./install.sh
cd ..
tar -zxvf mondo-1.13.tgz
cd mondo-1.12
./install.sh
```

Other Tools Needed by Mondo

Some distributions lack certain crucial packages. The packages most often missing are afio, cdrecord, bzip2, libnewt0.50, libslang1 and mkisofs. Some

users may have to create a gawk-to-awk shortcut. You can find these tools on the web site of the vendor of your distribution.

Make a Test CD

Making a test CD is a good idea because the new user can try it and not mess up his or her system. First, be sure that Linux knows how to use your CD writer. Then, run **mondo-archive**.

To find your CD writer, type

```
dmesg | grep CD
```

If your CD writer is an IDE device, it will show up here as /dev/hd*X*, *X* being a letter between a and h. If SCSI emulation is properly configured, you will see your CD writer listed when you type

```
cdrecord -scanbus
```

If your CD writer is properly installed, you will see

```
0,0,0 --- JoeCamel 4x CD Writer
```

or something similar. The *0,0,0* number to the left of the device description is the SCSI device where the writer can be found. Write this number down.

If you want your rescue CD to include certain special programs, e.g., your copy of BRU, add the file and its config files to /usr/share/mindi/deplist.txt by hand. Mindi will find the libraries and add them for you.

Run Mindi to create some boot disks just to make sure Mindi works properly on your system. Type

```
cd /usr/share/mindi
./mindi
```

If your kernel is too large (more than about 900KB), you cannot make boot floppies, although you still can make a bootable CD image. The easiest way to test Mindi in either case is to press N to “Create boot floppies?” and Y to “Create iso image?” Then use **cdrecord** to make a bootable CD-R or CD-RW. Type

```
cd /root/images/mindi
```

Then choose one of the following calls to write the CD, depending on whether the disk in the drive is a CD-R or a CD-RW. Please replace *x,x,x* with your writer's SCSI node. For CD-RW, type

```
cdrecord blank=fast dev=x,x,x speed=2 mindi.iso
```

and for CD-R, type

```
cdrecord dev=x,x,x speed=2 mindi.iso
```

Close all applications; reboot from the CD instead of the hard disk. (You may have to edit your BIOS settings to make your computer try to boot from the CD before the hard drive.) If your computer boots okay from the CD then you know Mondo also will generate a bootable rescue CD reliably. Of course, an ideal rescue CD will use your own kernel. I recommend that you use your own kernel if possible, to minimize the risk that the boot CD won't support your hardware or filesystems, etc.

Finally, to do a complete backup, type

```
cd /home
mondo-archive --burn-cds 2 0,0,0 --comp-level 9
```

The 2 indicates that you are writing at 2x speed. If you are burning to CD-RWs, type

```
mondo-archive --burn-cds 2 0,0,0 cdrw --comp-level 9
```

After running the command, insert a blank CD-R(W) into the drive and leave the PC running. That's all.

I always choose the maximum compression level (9) because I start Mondo and then go to work. When I come home, I insert the second CD-RW and wait a half an hour. That is a day's backup.

The default compression level is 3. If you are in a hurry, use **--comp-level 1** to speed up the backup process. You will use more CDs that way, but it should take less time to run.

If Mondo does not find a CD in the drive when it tries to write files to the CD, it will pause with a Retry/Fail/Abort message. If you insert a CD and choose Retry, it will retry as if nothing had gone wrong. If you choose Abort, the program will stop. If you choose Fail, the program will skip that CD but continue the backup process. Mostly, you should choose Retry.

If there are specific paths that you do not wish to backup, you may exclude them with

```
--exclude-paths /foo /bar /xanadu
```

If you want to include only certain paths, use **--bkpath /home**. So, if you want to backup only your home and boot directories but exclude the communal MP3 folder, use this:

```
mondo-archive --burn-cds 2 0,0,0 cdrw --exclude-paths /home/MP3s /home/WAVs /home/secret
```

If you do not want to burn the CDs immediately but would rather create ISO disk images to be burned later, do this:

```
mondo-archive --isodir /root --bkpath /home /boot --exclude-paths /home/MP3s /home/WAVs /home/secret
```

This will create 1.ISO, 2.ISO, etc., and save the files to the /root directory.

Before running Mondo-Archive, be sure to add some files to /usr/share /mindi/deplist.txt, run **mount** to make sure you have mounted the partitions that you want be backed up and run **df** to determine backup size/compression/CDs needed.

You can speed up the compare process by switching to another terminal after booting and running ide-opt. Use Alt-left cursor key and type **ide-opt**. This enables DMA and other good stuff.

To compare the backup against your live filesystem, please boot from the CD and choose compare mode (type **compare** and then press Enter). Check /tmp/mondo-restore.log after the compare cycle to see which files do not match. Aside from the initial teething troubles you might encounter with making boot disks from your kernel (some kernels are not appropriate for boot disks and have to be recompiled), you are likely to discover that Mondo is quite boring. It does what it says it does. It squeezes all your files onto your CD-R(W)s, and it restores them again if necessary. It partitions your drives, formats them, restores the data and runs LILO to set up your boot sector.

Bare-Bones Restore

Imagine that your hard drives happen to be wiped, deliberately or accidentally. Or, imagine that you want to clone your existing operating system. In either case, you want to run in Nuke Mode. After booting from your Mondo CD, type **nuke**, press Enter and watch the show. That's it.

If you want to see exactly what Mondo is doing while it is restoring, press Alt-A and type

```
tail -f /tmp/mondo-restore.log
```

to monitor its progress in detail.

Selective Restore

If you want to restore only some files, or if you do not want to prep/format your drives, then you should boot into interactive mode. Type **interactive** and then press Enter at boot time. You will be asked to say yes/no to a range of

questions: do you want to partition your devices? Do you want to format them? Do you want to restore everything? Do you want to restore something? Do you want to run LILO to set up your boot sectors?

Interactive mode is for people who have lost a subset of data from their live filesystem or perhaps who have lost some data from their latest backup and want to restore a subset of data from an earlier backup.

Using Expert Mode

When you have backed up your system, booted from the CD in compare mode and verified the archives, you are in a position to experiment with your system. You could move partitions around, resize them, enable or disable RAID, play with other boot loaders, etc. That is what Mondo is really good for. The greatest threat to a typical Linux installation, in my opinion, is a careless root user.

If you want to try some of these tricks, boot from your Mondo CD and choose expert mode. This will drop you to a shell prompt. Edit the mountlist file. Then, type **mondo-restore** and choose Interactively from the pop-up menu. The mountlist is a text file, /tmp/mountlist.txt, located on the RAM disk after you boot from the Mondo CD. It lists the various partitions that will be created, their sizes, their mountpoints and their formats. To change the size or layout of your partitions, just edit that file with **pico /tmp/mountlist.txt** (or use your favorite editor). Save and close by pressing Ctrl-X and then Enter.

Here is a sample mountlist. The sizes are in kilobytes, so count the zeros carefully when modifying the values. Remember, the new layout will not take effect until you run **mondo-restore** to repartition and reformat the drives:

```
/dev/hda1    /mnt/windows       vfat        4096000
/dev/hda2    swap               swap        256
/dev/hda3    /                  ext2        8192000
```

To change the root partition's size and format, simply alter the relevant fields:

```
/dev/hda1    /mnt/windows       vfat        4096000
/dev/hda2    swap               swap        256
/dev/hda3    /                  reiserfs    16384000
```

Or, you could move from using multiple primaries to using a primary partition (hda1), an extended partition (hda2, created/handled by Mondo) and several logical partitions. Note the new /dev/hdaN entries:

```
/dev/hda1    /mnt/windows       vfat        4096000
/dev/hda5    swap               swap        256
/dev/hda6    /                  reiserfs    8123000
/dev/hda7    /usr               reiserfs    4099000
/dev/hda8    /home              reiserfs    4099000
```

If you have added a second hard drive (e.g., primary slave) then you could move some of your partitions to that drive. See below and note the changes:

```
/dev/hda1    /mnt/windows       vfat        4096000
/dev/hda2    swap               swap        256
/dev/hda3    /                  reiserfs    81422000
/dev/hdb1    /home              reiserfs    9481000
/dev/hdb5    /usr               reiserfs    16384000
/dev/hdb6    /tmp               reiserfs    1589000
```

It is slightly more complicated to move to RAID because you have to create an /etc/raidtab file. You can do this from within Expert Mode. Just type **pico /etc/raidtab** and create a good raidtab file. (That is beyond the scope of this article.) Then, replace the conventional device with a RAID device (/dev/mdN):

```
/dev/hda1    /mnt/windows       vfat        4096000
/dev/hda5    swap               swap        256
/dev/md0     /                  reiserfs    16384000
```

After editing the mountlist, run **mondo-restore**. When asked if you want to partition and format the drives, say yes. You may want to restore the data or run LILO to initialize the boot sector, depending on what you are doing. If you are simply testing a new partition layout, you probably want to say no when answering the other questions. Otherwise, say yes.

Conclusion

Good backup software must be well documented, well tested and easy to use. If the backup software is too fiddly then the user simply will not bother. If it is unreliable, then the user will be lulled into a false sense of security, only to have his or her hopes dashed when the crunch comes. I back up my own system with Mondo two or three times a week. I restore regularly as part of the test regime.

If you want to participate, please download the program and join the mailing list. I would love to hear from you.

Other Uses of Mondo

email: hugo@firstlinux.net

Hugo Rabson is a 26-year-old Briton living and working in Nashville, Tennessee. His primary qualification is in business, but he has been writing code since puberty. He currently works for WebMD. Mondo is his attempt to give something back to the Linux community, and he gladly will accept all comments, suggestions and patches.

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

Webmin: Good for Guru and Newbie Alike

Dirk Elmendorf

Issue #92, December 2001

The modular design and absence of a back-end database make for a powerful and adaptable interface.

To quote their web site, “Webmin is a web-based interface for system administration for Unix.” I found Webmin a few years ago when I was looking to register the domain webmin.com. I had an idea for a cool control panel to help administer my Linux server farm, and I was annoyed that someone had gotten to the domain first, until I started looking at the software they had for downloading. After that I was hooked.

Webmin's goal of providing a web-based interface is not unique. A number of tools are out there, both open source and commercial, with the same goal. What makes Webmin stand out from the crowd is that it can be useful to both experienced and inexperienced Linux administrators. Before I explain how it fulfills such a difficult task, let's look at how you install the software and what you can do with it.

Getting Started

Before we delve into the feature set of Webmin, I should explain a little about its design. Webmin is basically a large collection of CGIs written in Perl. It runs its own web server on a port that you specify when you install it, allowing you to manage the security of Webmin separately from your real web server. The full list of supported platforms is on the site, and it includes Red Hat, Solaris, Debian, *BSD, HP-UX, IRIX, AIX, DEC, SCO and Mac OS X.

What makes Webmin unique is its modular design. Every service or feature is exposed through a set of modules. This means that if the application you would like to control from Webmin is not supported, you can use the Webmin API to build a new module that will allow Webmin to control it. Webmin ships with 50 standard modules and many more are available, so you have plenty of source

code examples to help get you started. Although Webmin itself is distributed under the BSD license, the modules you build can be distributed under any license you choose. This keeps the tool open to submissions from both the Open Source community and commercial vendors.

Installation is a breeze; go to www.webmin.com/webmin and download either the RPM or the tarball. If you use the tarball, you will need to run the setup.sh once you have unpacked it. This program handles the installation for you. One tip if you use the tarball: make sure you unpack it where you want those files to live. After you have installed Webmin, do not delete the Webmin directory you just unpacked or Webmin will not work. If you want to uninstall Webmin later, an uninstall script is included that will remove the initial directory as well as all the other files Webmin puts on your drive.

Now that you have Webmin installed, you need to open a web browser and log in to the IP address/resolvable server name and port (typically 10,000) that you configured Webmin to use when you installed it (see Figure 1). Once you have logged in, you should be greeted by the index page of Webmin, including several category tabs in which the various Webmin modules are grouped (see Figure 2). For example, the “Webmin” tab is where all the configuration settings for general themes, users, modules, etc., are located (the default theme is used in these figures). Other tab groupings include System, Servers, Networking, Hardware and Others.

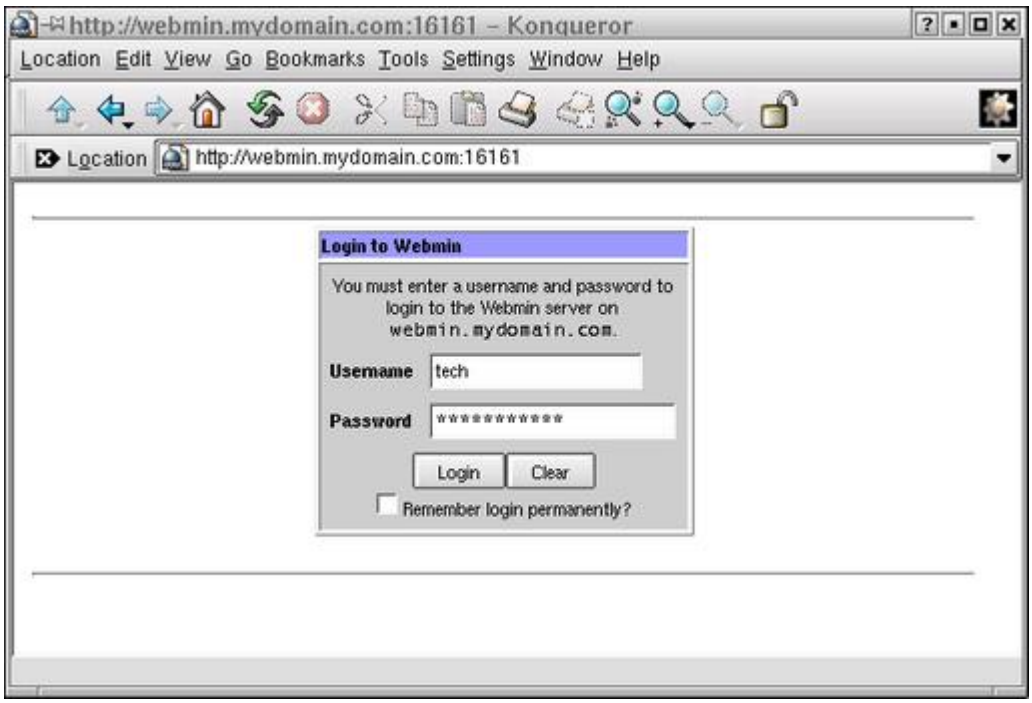


Figure 1. Logging in to Webmin

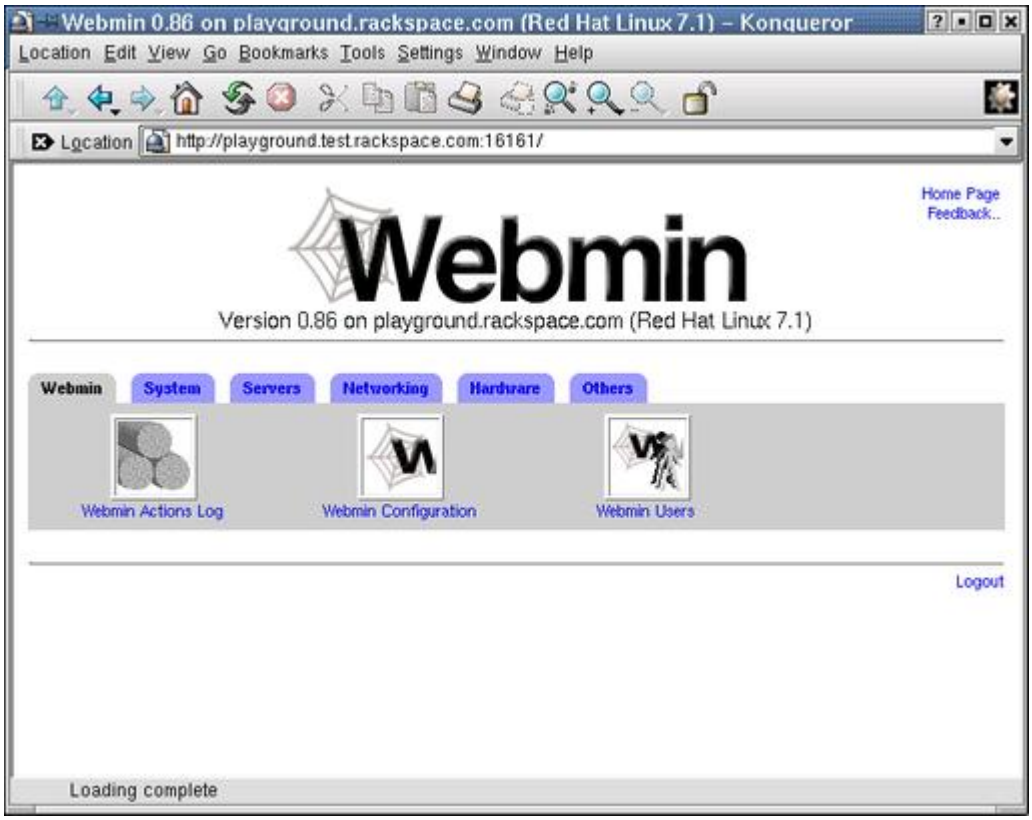


Figure 2. Webmin Tab Groupings

The System tab handles tasks that have to do with managing the computer or server itself. This includes setting disk quotas, setting up NIS, altering PAM, looking at the syslog, adding users, managing cron, setting up NFS and even modifying boot-time services or rebooting the server.

The Servers tab groups all the server service configuration modules together. This is where you will find Apache, BIND, DHCP, sendmail, Squid and a number of other related modules (see Figure 3).

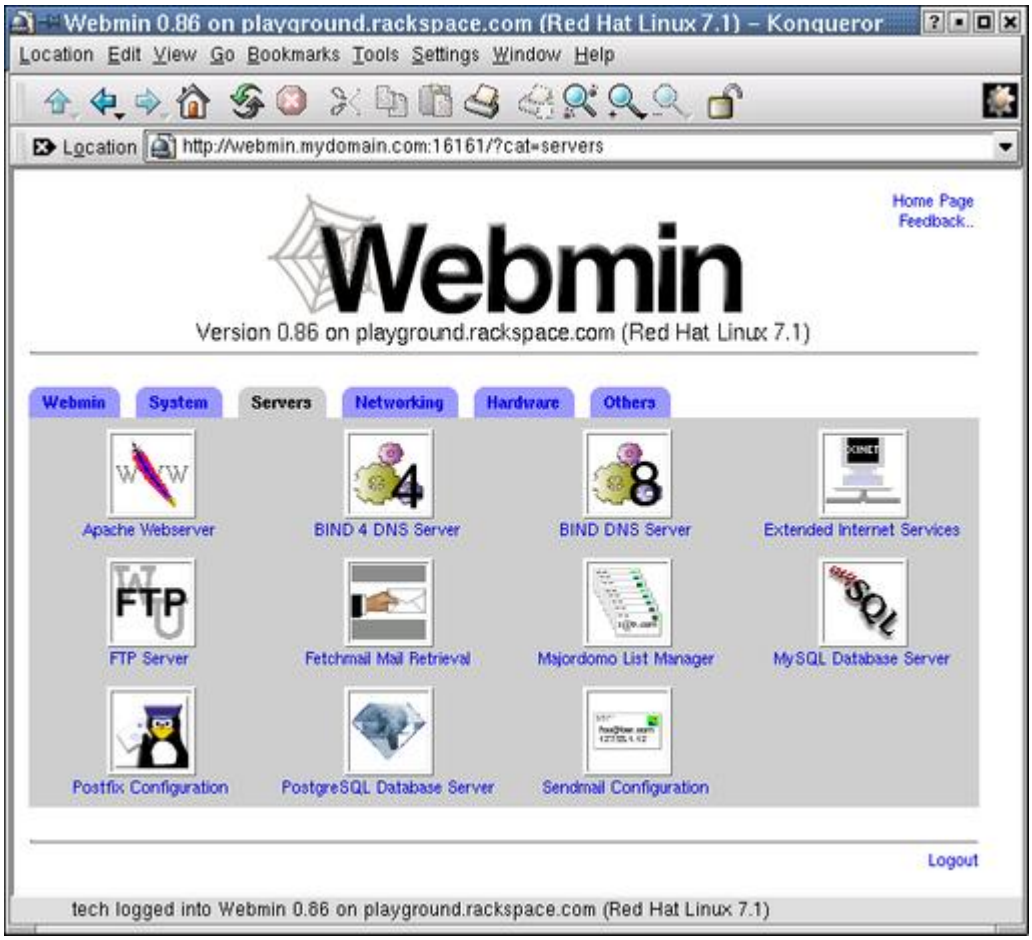


Figure 3. Webmin Modules under the Server Tab

The Networking tab is a new addition, and on Linux it offers a good GUI ipchains configuration module, as well as a subcategory of various network utilities such as ping, traceroute, whois and dig.

The Hardware tab, of course, handles things that are directly hardware-related. This is where you will find information about your disk partitions, system time, network interface configurations, LILO and even software RAID.

The last tab is a potpourri category that has a full Java-based ssh/Telnet client, Java-based file manager, special user-definable module called Custom Commands, system log viewer, Perl modules section, and web-based system and server-status monitoring module. The Custom Commands module lets you build a simple interface to Webmin that allows you to run any command you want. This can be very useful if you need Webmin to do something specific that does not require a full module to accomplish. Figure 4 shows some custom examples that were created quickly and easily.

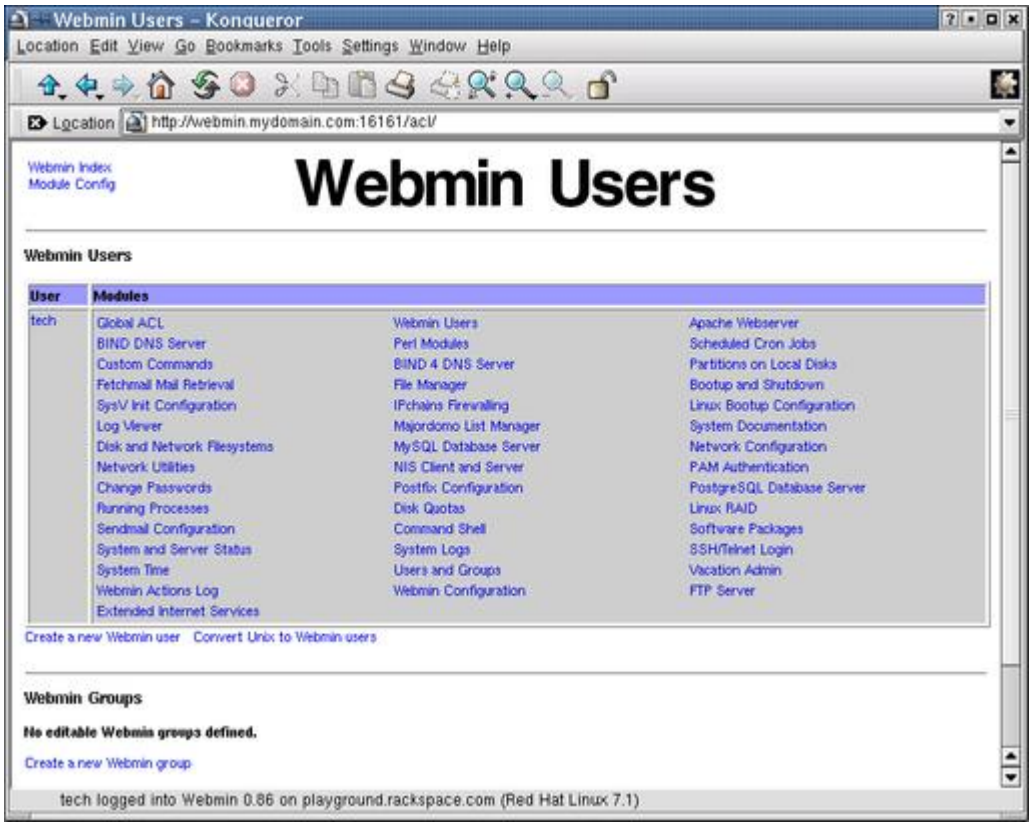


Figure 4. Creating Custom Commands

Step-by-Step Guide

Here is a step-by-step guide to configuring a new virtual-hosted web site in Apache. This guide will show you how to set up the document root, cgi-bin and logging. It looks a lot more complicated than it is. (I discovered that writing the directions for a point-and-click interface takes a lot more effort than just pointing and clicking.)

For this example, the user ID “dirk” already exists, and the virtualhost/domain we will create is test.com. It also assumes that you are using Wu-FTPd and that the domain you are setting up already resolves in DNS. First create the directories (folders) that will contain the HTML files, log files and CGI scripts for your web site. Go to the main Webmin page and click on the Other tab. Next, click on File Manager.

Use File Manager to navigate to the home directory of the web site's administrator. The administrator's home directory is the directory that the user will go to when using FTP to connect to the server. The home directory of this user is /home/username. Double click on the home directory (folder) in the left column. Double click on the user name of the web site administrator inside the home directory.

Now you need to create the necessary directories for the web site. Click on the New Folder button (the button that says “New” with the folder icon) to create

the main directory for your domain (e.g., /home/dirk/test.com). Double click on the directory you created (e.g., test.com) to move inside it. Now, create three folders (htdocs, logs and cgi-bin) in the directory you just created. In our example, we would have created the following directories: /home/dirk/test.com, /home/dirk/test.com/htdocs, /home/dirk/test.com/logs and /home/dirk/test.com/cgi-bin.

Now, go back to the home directory for the user. In the right column, click once on the directory for the web site (e.g., test.com), then click on the Info button at the top of the page. In the Info window, change the user and group to the correct user ID (e.g., dirk). Change the pop-up menu in the "Apply changes to" section to "This directory and all subdirectories". Click Save.

Go back to the main Webmin page by clicking on "Return to index" at the bottom of the page. Click on the Servers tab, then click on Apache Webserver. If this is the first time you have clicked on Apache Webserver in Webmin, it will ask you to allow Webmin to check the configuration of Apache (just click on Configure).

Now scroll to the bottom of the page, and you will see a form for adding new virtual sites. Enter the IP address for the web site in the Address text field. If you are doing name-based hosting, make sure the "Add name virtual server address" box is checked. The Document Root should be /home/*userid*/*domain*/htdocs (e.g., /home/dirk/test.com/htdocs). Enter the name of the web site in the Server Name text field (e.g., www.test.com), and click Create.

Scroll through the Virtual Servers list until you find the web site you created. Click on the Virtual Server for that web site. To set up your cgi-bin folder, click on "CGI Programs" in the "CGI directory aliases" row. In the From text field, enter /cgi-bin/. In the To text field, enter /home/*userid*/*domain*/cgi-bin/. This must end with a "/" (e.g., /home/dirk/test.com/cgi-bin/). Click Save.

Click on Log Files. In Access Log Files, enter /home/*userid*/*domain*/logs/access_log (e.g., /home/dirk/test.com/logs/access_log) in the text field that has File or Program written above it. If you would like to have a log that contains hits and referrer information, then in the Format column of the "Access log files row", switch the check box from Default to the check box next to the text field and enter "combined" in the text field. Click Save, and click on "Networking and Addresses". Enter the e-mail address of the webmaster for this web site in the "Server Admin email address" text field. Then check the box next to the text field, and click Save.

You have now set up the web site, but before it goes live there is one final and important step. In the upper right-hand corner is a link called Apply Changes; click on it to activate your changes.

Security

Webmin offers a number of security features. The first line of defense is a user ID/password authentication system that is completely independent of the user ID list kept in the /etc/passwd file. This means someone can be granted access to Webmin without allowing the user any other OS-level system privileges. Webmin also fully supports SSL. If you have the Perl SSL module installed, all your Webmin sessions can be encrypted to prevent attackers from gaining information about your system from watching them.

Webmin also allows for fine-grained control of the various available modules. You can, for example, grant users control over the entire DNS server without giving them access to Apache configurations or limit them to only the domains they own in the DNS module (see Figure 4). This ability to limit and redistribute control can come in handy if you need to delegate tasks to other administrators. Finally, Webmin also can be configured to log all changes made via the interface, a feature that is extremely useful for troubleshooting.

What Makes Webmin Great?

As you can probably tell, I am a big fan of Webmin. I like that the license lets me get the source code and do what I want. I love the fact that the module system lets me build new things or add modules others have built. I currently am testing the LTSP module for Webmin to help manage some unruly I-Openers. The ability to transfer tasks to less experienced administrators (roommates) and know that they cannot deviate from the area I have provisioned for them, lightens my workload.

If that were all Webmin had to offer, I would be quite impressed. There is one final benefit of the Webmin architecture, however. Webmin works directly with all the files on the system to affect its changes, meaning it does not use a database or other means of storing information that is in any way nonstandard. I can therefore edit the httpd.conf for Apache by hand without worrying that it will cause problems for Webmin. From a support standpoint, this means I can install Webmin on a server and turn it over to someone else. If that person problems I still can use all of my scripts and vi knowledge to troubleshoot the problem.

The command-line friendliness and absence of a master back-end database that pushes the config out to the flat files are things that control panel designers all too often forget. They end up building a system requiring that

everything be done via the control panel or it will break. Webmin allows me to mix-and-match administration styles at my convenience. For example, I tend to make my Apache configuration changes directly. BIND, however, is a different story. BIND is notoriously picky, so I use Webmin as a convenient front end to BIND. It offers me all the esoteric options and greatly lowers the risk that a typo will cause a resolve error. To me, the amazing thing is how well Webmin fits into my administration toolbox. I do not have to use it, but it is always there.

New administrators will come to like Webmin because of its depth of features. The point-and-click interface means that you do not have to keep everything in your head, which can prove to be a daunting task for someone new to administering a Linux server. Webmin's core modules expose almost every feature and function of the services they support. This means you easily can add configuration options that you did not previously know existed.

My only caveat to this is that although Webmin is well organized and feature-complete, it is not for absolute beginners. If you do not know what an A record for DNS is, then Webmin will not help you. Webmin maps the underlying Linux to a web interface, so sometimes when you get this much flexibility and power, you have to sacrifice ease of use. Once you know the fundamentals of the services, Webmin can be a great tool—just don't expect it to summarize the O'Reilly book on BIND for you.

Resources

email: dirk@rackspace.com

Dirk J. Elmendorf is one of the cofounders of Rackspace Managed Hosting and serves as chief technology evangelist. He also serves as a research and development leader, helping to develop and evaluate the new products and services he promotes in his evangelical role.

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

LINUX JOURNAL

LaTeX2HTML: Publish Science to the Web

Michael Yuan

Issue #92, December 2001

Tailoring math-intensive documents to fit the Internet.

Ironically, 20 years after the advent of the World Wide Web, it is still a struggle to publish math-intensive scientific documents on the Web. New standards like MathML have been out, but before the popular web browsers support them, the best way to reach a maximum audience is still to put math formulae as in-line graphics in HTML documents.

In the off-line world, TeX/LaTeX packages are known to produce the best formatted scientific documents. LaTeX is, in fact, a standard format for many academic journals. For scientists and researchers, the best of two worlds is to write in LaTeX and then publish on the Web. LaTeX2HTML is the perfect tool for that.

LaTeX2HTML originally was written by Nikos Drakos in 1993 as an open-source project. It was improved and enhanced by many members of the Open Source community. LaTeX2HTML converts LaTeX documents to standard HTML format. Since HTML is a text format, it converts all the nontext elements into in-line graphics. It is a Perl application and runs on many platforms including most flavors of UNIX/Linux and Windows.

In this article, I first give an example of how to convert a simple LaTeX document to HTML and discuss how the program handles in-line graphics and style files. I then explain the LaTeX2HTML-specific LaTeX commands/environments. Finally, I discuss a more creative use of LaTeX2HTML—using it on the server side. For simplicity, the term formula refers to anything that is in LaTeX's math mode. That includes simple in-text math symbols, “displaymath” formulae and numbered equations.

A Simple Example

The usage of the command is very straightforward. If you want to publish your LaTeX document, mydoc.tex, to the Web, you run

```
latex2html -local_icons mydoc.tex
```

LaTeX2HTML creates a new subdirectory (mydoc/) and puts all HTML files and image files in it. You can now copy mydoc/ under the document tree of your web server, and your document is available at http://*yourserver*/*path*/mydoc/index.html.

The title, headings and emphasized text are marked up properly with HTML tags. All the formulae/tables/figures and anything that normal HTML cannot display are converted into in-line graphics.

If mydoc.tex has several sections/subsections, LaTeX2HTML generates an HTML file for each section and a table of contents with links to each section. Each page has a navigation bar with links to next/prev sections and the table of contents. If you want only one big HTML file, use the **-split 0** command-line switch.

LaTeX2HTML needs to know where to find the icon images on the navigation bar. The simplest solution is to use the -local_icons command-line switch, which instructs LaTeX2HTML to copy the navigation icons into the directory mydoc/. You also can use custom navigation icons by replacing mydoc/*motif*.gif files with your own.

In-Line Graphics

I use math formulae as an example to explain how LaTeX2HTML handles in-line graphics. It does the same thing for floating objects such as figures and tables.

LaTeX2HTML scans through mydoc.tex and extracts all math formulae into a file called images.tex. In images.tex, each formula is on a separate page. It then calls **latex images.tex** and **dvips -S 1 -i** to generate a one-page PostScript file for each formula. Ghostscript is used to convert and resize PostScript files into GIF or PNG images. LaTeX2HTML will remember the image names and insert the links back to HTML files generated from mydoc.tex.

There are several command-line switches to control the image generation process. For example, switch ps_images instructs LaTeX2HTML to use links to external PostScript files rather than in-line graphics.

Style Files

LaTeX2HTML handles style files differently for text and in-line graphics. LaTeX2HTML ignores any style file when it generates main text HTML files from mydoc.tex. However, style files often define new commands and environments that cannot be ignored. As a workaround, LaTeX2HTML allows users to rewrite the style file in Perl format. When LaTeX2HTML encounters a style file, it searches for the Perl translation and incorporates it into the main script. Fortunately, many popular style files already have been translated into Perl. To write such translations requires a working knowledge of the LaTeX2HTML internal structure. It is a difficult task, but you have the source code.

We want formulae, figures and tables to look exactly the same as they would print out on paper, with all the custom symbols, fonts and spacing. So, images.tex includes all the style files used by mydoc.tex. Since images.tex is processed by LaTeX, the style files are handled correctly.

Write for the Web

LaTeX2HTML is powerful in converting existing documents into web format. LaTeX2HTML offers a set of custom LaTeX commands and environments that allow you to include your own HTML markups in the text. You can take full advantage of normal HTML: you can add web forms, clickable image maps, external links/graphics or even Java applet/script to your document. For example, if you want to include a link to an outside web page in your generated HTML pages, you can just insert the following to mydoc.tex:

```
\htmladdnormallink{link title}{http://yourlink}
```

or more generically,

```
\begin{rawhtml}
<A HREF="http://yourlink">link title</a>
\end{rawhtml}
```

If you want to print out a paper version of that HTML-enhanced document, you can include style file html.sty and then process it with normal LaTeX. Most of those LaTeX2HTML-specific commands/environments are ignored by LaTeX.

On the Server Side

LaTeX2HTML also can be used at the server side to facilitate web-based mathematical communications. Listing 1 demonstrates a Perl function that takes in a math-mode LaTeX string and returns an HTML markup for an image that displays the formula. There are other ways to do it more efficiently, but this is just an illustration of what LaTeX2HTML can do. It can be used in chat

rooms or bulletin boards to allow the user to input and broadcast complex math formulae.

Listing 1. Perl Function Converting a LaTeX String to an HTML Markup

Resources

email: juntao@astro.as.utexas.edu

Michael Yuan is a PhD candidate in Astrophysics at University of Texas at Austin. He studies remote quasars (20-plus billion light years away) to understand the history and evolution of our universe. When he is not observing quasars, he enjoys developing useful software using earthly languages such as Java and Perl.

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

Editors' Choice Awards

LJ Staff

Issue #92, December 2001

After much deliberation, here are our choices for 2001.

The Linux landscape has been altered considerably since last year's Editors' Choice Awards, which were given at what was somewhere near the peak of the technology/dot-com boom from which the economy is still attempting to recover. Last year we remarked at the difficulty of choosing winners among such a numerous field of competitors, and you'd think that a reduction in competitors necessarily would make the decisions easier. But it ain't necessarily so. Though there's been a decrease of vendors in some categories (most notably hardware), there are still numerous quality offerings (free and otherwise) that, along with the Linux kernel, have improved in quality during the last year.

The *Linux Journal* Editors' Choice Awards are open to both open-source and proprietary software, and among this year's picks in the software categories you'll find representations of both. While all of us may now be smug in being able to choose among so many open-source products for our software needs, let's not forget to do our best to ensure this choice continues by guarding against the dangers of SSSCA-like legislation that would require mandatory Digital Rights Management, making a free or even source-available operating system illegal. Write directly to Senator Hollings and Senator Stevens and to your representatives if you are a US citizen (you can look your representative up on congress.org), support or join the Electronic Frontier Foundation and educate friends and associates.

Contrary to last year we won't encourage the use of alcohol in reconciling you to any of our choices that may not be in accord with your own. But if you find yourself piqued in the extreme with any of our selections, may I refer you to our Readers' Choice Awards?

Server Appliance: Filanet InterJak 200 802.11b

Do you want to set up an 802.11b network with high-performance antennas for inexpensive WAN connections? Do you want an 802.11b base station with VPN functionality to keep users' laptops secure? Do you want to deploy centrally managed mail, Samba and VPN boxes to all your company's home office users? Filanet makes a series of inexpensive, fanless embedded Linux network devices, based around a custom ASIC with an ARM CPU core and hardware 3DES, that will solve a lot of problems for companies and ISPs at only a little more than the price of a dumb DSL box.

Security Tool: Nmap

You know your program has caught on when people start to use its name as a verb. Running Nmap every time you set up a new Linux server, and periodically to see if anything has changed on your network, has become a standard security practice. It's no coincidence that the spread of Nmap has coincided with Linux distributions finally paring down the menu of potentially exploitable services offered by default. For providing an easy-to-use “security idiot light” to Linux system administrators and distributions everywhere, Nmap, we salute you.

Web Server: APPRO 1124

We put this system's dual Athlon MP motherboard in our Ultimate Linux Box but APPRO, working from an original design by VA Linux Systems, put it in a thin, rugged 1U server with four hot-swap SCSI drives plus a thin CD-ROM. Powerful blowers and a custom power supply make this the web server we wish was on the market when there was still such a thing as a web server budget.

Office Workstation: Thinkpad T Series

Any of this year's notebook computers are cheap enough and fast enough to be a primary workstation for almost anyone. But when you have a machine that you need to work and can't fix yourself, you need really good service. The following is a true ThinkPad service story for a 1998 model with a broken display cable: IBM sent a padded shipping box at their expense; we returned it with the computer inside on a Tuesday. It came back that Thursday. Our cost: $0. Got to love that three-year warranty.

Other things we like about ThinkPads include the keyboard, the red nipple pointing device and the thriving user community that posts good compatibility reports to the site. The T series, which Phil, our publisher, carries now, has a nifty white LED light just bright enough to illuminate the keyboard and comes with screen resolutions up to 1400 x 1050.

Some new ThinkPads are available with 802.11b; check www.linux-laptop.net/ibm.html for the latest compatibility reports. Sadly, not all models can be ordered with Linux preinstalled, and IBM, in violation of Microsoft's license, does not offer Windows refunds.

Technical Workstation: Monarch ULB 1200 MP

This is the box spec'ed by our staff and built (and currently offered for sale) by Monarch Computer. It was one of the two Ultimate Linux Boxes described in our annual ULB article in the November 2001 issue. It's got a great looking case and features some excellent hardware including the Tyan Thunder K7 motherboard.

Web Client: Konqueror

Put down that crack pipe, I mean Netscape 4.x browser, Linux fans! Mozilla and Konqueror have both reached the point of stability and featurefulitude that we needed to drop the crusty, Motif-based old Netscape for good. We had to give the Editors' Choice to Konqueror because of its excellent integration with the KDE desktop environment, general speed and ability to easily use that Flash plugin we need to see all those goofy animations people keep sending us. And there was much rejoicing.

3-D Application Tool: Maya 4

As reported in Robin Rowe's recent GFX columns, Linux literally is taking over the motion picture industry for use in special effects and animation. In no other industry is there such a massive migration to Linux. Maya is a big part of this, porting their product to Linux in response to customer demand. It's even earned Linus' approval, and he calls it “the most complex and powerful 3-D graphics application ever to run on Linux”.

Backup Tool: BRU Pro

We thought we had lost BRU to corporate shenanigans, but thankfully longtime BRUmeister Tim Jones, formerly development VP of BRU's original vendor EST, saved the old-school backup workhorse and is offering it under the TOLIS Group brand name. BRU offers easy configuration of your backup plan to match it to the tapes you use and sponsors the linuxtapecert.org web site that lists tested and approved tape drives for Linux.

Miscellaneous Utility Software: Acronis OS Selector

This is a nifty boot and partition manager that has the great advantage of supporting ReiserFS for that added data protection.

Communications Tool: OpenSSH

We have twenty-some OpenSSH processes going on one server at our offices at *Linux Journal*. One workstation has six. We start up tunnels, scp stuff around and basically live in ssh sessions. It's convenient, stable, and a real pleasure to set up and administer. But the real reason for giving this award to OpenSSH is that if not for OpenSSH, we'd all have to live in Seattle.

Development Tool: KDevelop

KDevelop has a thriving user community, offers debugging and class browsing tools and even makes it easy to start up a new project in the standard GNU style. People coming from proprietary IDEs will find that KDevelop can mimic several popular interface styles. Embedded distribution vendor REDSonic chose KDevelop as the integrated development environment for their RedIce Linux.

Database: Oracle

Ported to Linux in 1999, Oracle has become quite a competitor. Last year the award went to PostgreSQL, and while it's still a strong contender and has received a lot of publicity this year, one can't ignore Oracle's sheer performance.

Office Application: AbiWord

This word processor starts up in about three seconds on a decent system and takes about 5MB of memory for a blank document. That is not a misprint—just a good, basic word processor, nothing fancy. Yes, it has printing now, and yes, it will import Microsoft Word documents. Try it—you'll either like it or you won't have wasted much time downloading it.

Desktop Environment: KDE 2

The new KDE desktop environment has a bit of a way to go as far as resource consumption and stability, but each succeeding version promises that it is on its way to a high polish. It has improved architecture and some very active development. Among the nicest features is the integration of the KDE browser, Konqueror, into the desktop as a file manager. Type any word into the address bar and get a Google search of that term. Also, KDevelop is completely integrated as well—see Development Tool.

Mobile Device: Compaq iPAQ

Linux-capable PDAs come in two flavors: capable of running minimal software only and high-powered with speed and space to experiment. The iPAQ is the

best of the latter bunch, with a good industrial design except for the almost-symmetrical stylus. It's attracted quite a development community, so there are plenty of applications and documentation for people starting out with a Linux PDA. And with accessories such as a PCMCIA card sleeve and an upcoming camera/accelerometer, the iPAQ looks to be a good platform for Linux innovation into the future.

Embedded Development Tool: Lineo's Embedix SDK 2.0

Lineo has done a good job of making a product that appeals to a wide range of embedded developers. Embedix has a nice interface, plentiful features and great documentation that work to lessen the embedded Linux learning curve and allow developers to concentrate on their applications.

Real-Time Tool: Preemptible Kernel Patch, Nigel Gamble et. al., MontaVista Software

This patch is only 1,000 lines and uses the existing kernel SMP-locking strategy for respectable real-time gains at a small price. Not just for embedded systems weenies, everyone who wants to listen to an audio stream and untar a big archive at once will appreciate this.

Training and Certification: Linux Professional Institute

The LPI team did an extensive research project before beginning its exams to determine what skills Linux professionals actually use in their jobs. Then they conducted an extensive item-writing and technical review process, performed a Modified Angoff study on Linux experts and did a bunch of super-scientific Hari Seldon stuff to give the test the best possible chance of actually measuring Linux skill. Newly LPI-certified people report that the test is hard but fair.

Game: *Tribes 2*

This multiplayer game was developed by Dynamix and ported to Linux by the folks at Loki. As our game reviewer, Neil Doane says, “Not since *Quake III* have so many developers lost so much time, over so many networks and produced so little. This game rockulates.”

Book: *Just for Fun: The Story of an Accidental Revolution* by Linus Torvalds with David Diamond

It's a testimony to the factiticy of Internet Time that one of the Net's most influential personalities came out with his first autobiography at age 31. We say first because the fun has clearly just started.

Linus Torvalds is no less accidental as an author than he is as a revolutionary. But in a way that's the point. The book seems less published than floated, as the title says, just for fun. It's one side of a conversation about some stuff that might be worth talking about. If not, well, the author doesn't care. Viewed from an angle more native to Linus' tribe, *Just for Fun* is a hack, and an early one, subject to completion and revision over time. From similar angles it's not hard to see as a set of man pages or bug lists.

It's an interesting irony that the operating system best known for its founding character actually is authored by a vast peerage of other characters. It's a subtle thing, but reading this book is a great way to gain insights into what brought all these characters together, and into why it's possible for anything so ordinary as an operating system to be so darn much fun.

Toy: Velcro

In today's economy a lot of people can't afford to go out buying hot geek doo-dads on impulse like some of us had been doing for a while. Fortunately, there are no new expensive must-have toys this year. Time to consolidate, become more efficient and get ready for the next boom. Velcro ties for organizing cables, adhesive-backed Velcro for putting on equipment and pre-Velcroed products are all part of living a more organized, neater life, especially if you use a laptop as your main work machine and need to bundle up all those accessories and cables. No word on whether or not Velcro is good feng shui, but if it helps tidy up messy cables it can't hurt. And it's fun.

Web Site: LinuxDevices

Recently acquired by DeviceForge LLC, LinuxDevices is back in the hands of its founder, Rick Lehrbaum. The site has done a terrific job of providing a wealth of information in the form of news, HOWTOs, product reviews and comparisons, and discussion forums. While primarily focused on embedded Linux, it has a lot to offer the average Linux geek as well.

Product of the Year: SuSE Linux 7.3

In some ways the last year hasn't been a banner one for the German company with the tropical mascot. SuSE took some collateral damage from the dot-com implosion, destaffing offices in the US and cutting back elsewhere in the world.

But while the dot-commies of the world played air guitar, pretending to have real business models, SuSE continued to produce music the old-fashioned way, and for old-fashioned customers. They were a business, and they were in business to do business. The result in 2001 was a series of 7.x distributions,

each built around the 2.4 Linux kernel, that have earned a torrent of praise for their comprehensive utility, their documentation and their performance, among many other virtues, all adding up to a real winner. Plus, it's currently the only distribution to offer an encrypted filesystem as an install-time option.

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

About LinuxBIOS

Eric Biederman

Issue #92, December 2001

Eric shows how performance and increased adoption of LinuxBIOS is attracting attention from embedded and cluster developers.

More than a year ago I was hired by Linux NetworX to work on LinuxBIOS, and I've been on a steep learning curve ever since. After working on LinuxBIOS, I am qualified to say that I have no question that the kernel code is high-level code and that C is a high-level language.

What Is LinuxBIOS?

When a microprocessor powers up, it starts executing instructions located in a ROM chip. These initial instructions are responsible for initializing the hardware (especially enabling RAM) and loading an operating system. The implementations and interfaces to this functionality vary from machine to machine, but its basic responsibility remains the same.

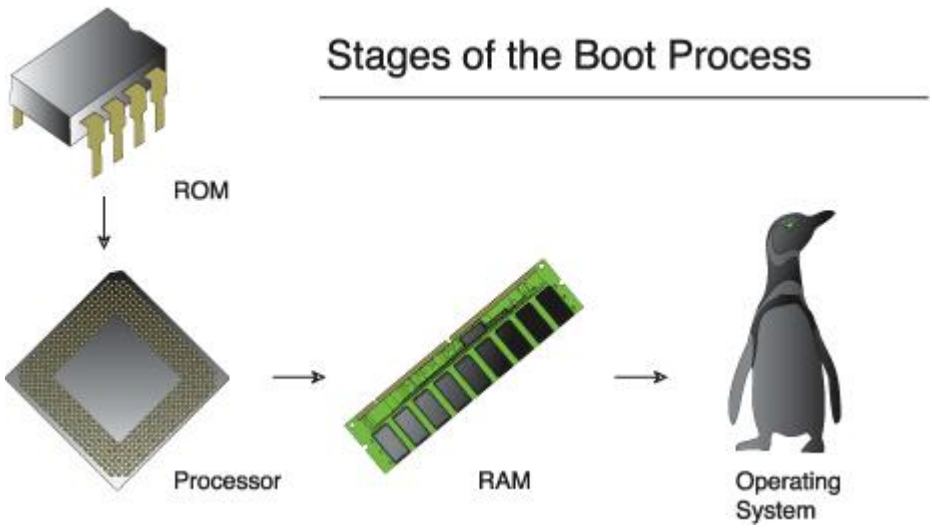


Figure 1. Boot Process Stages

On the Alpha platform the microprocessor reads an entire serial ROM, referred to as the SROM, into the instruction cache and begins executing code. The code in the SROM initializes the processor and memory and loads the SRM from a Flash EEPROM. The SRM then loads the palcode (basically the real kernel on the Alpha), initializes a little more hardware and loads an operating system. Since the firmware is split into two pieces, the SRM can be upgraded or even replaced. In fact, the initial design of the Alpha architecture specified that there would be different firmware (at the SRM level) for each operating system.

The x86 microprocessor begins executing code in 16-bit mode at 16 bytes short of the end of the address space, with the CS register pointing to 64K below the end of the address space. On the 8086 this is at address 0xf000:0xfff0 == 0xffff0, just below 1MB. On the 80286 the address is 0xffffff0, just below 16MB. And on the 80386 and above, this space is at 0xffffffff0, just below 4GB. For the 286 and later Intel processors, the value in CS is not one that you can ever load again. To compensate for this, the hardware maps the ROM chip at both 0xffff0000 and 0xf0000.

Unlike the Alpha, x86 processors fetch instructions one at a time from the ROM chip. As ROM chips are essentially ISA devices, this leads to some interesting consequences, the first being that until some form of caching is enabled, the code runs quite slowly. The second effect is that the chipset must come up mostly enabled, as the usual path to the ROM chip is from the CPU to the Northbridge, to the PCI Bus to the Southbridge, to the ISA Bus to the ROM. When working with a known good board, this second fact makes the debugging of initial devices much easier.

The standard PC BIOS has the responsibilities of initializing the hardware, loading an operating system and providing a variety of services (mostly in the form of minimal device drivers) after an operating system has loaded.

SPARC and PowerPC architectures have specified firmware, also known as OpenBoot, Open Firmware or the defunct IEEE 1275. The standardized Forth firmware sits close to the same location that the SRM does on the Alpha. There are several unique things about Open Firmware: it runs on multiple processor and machine architectures; it uses a Forth-based byte code, so the binaries are processor-independent; and it does most of its system initialization from this Forth-base byte code.

The Itanium/IA64 architecture uses the EFI firmware and is more architecture-dependent than Open Firmware because its drivers are either IA32 or IA64 code. In scope it appears to be even more ambitious; EFI includes an IP stack and some filesystem drivers. As with Open Firmware, the early hardware initialization stage is not specified.

Requirements placed upon the firmware by the Linux kernel are minimal. The Linux kernel directly drives the hardware and does not use the BIOS. Since the Linux kernel does not use the BIOS, most of the hardware initialization is overkill. Linux is not alone in this respect; I don't know of a modern operating system that doesn't follow this trend. Modern operating systems require only basic system initialization services. Extra device drivers and system features that firmware like EFI, Open Firmware or even a PCBIOS provide are not necessary except to help load the operating system. Since these services are not necessary, the LinuxBIOS code does not provide them.

The LinuxBIOS code is sufficient to load a standalone program encoded as an ELF executable from a Flash ROM. A standalone program can be an operating system kernel like Linux, but most standalone programs are hardware diagnostics or boot loaders (e.g., Memtest86, Etherboot and RedBoot). LinuxBIOS is expected to be paired with a standalone boot loader in order to load the operating system.

The original idea of LinuxBIOS was to load the Linux kernel from the ROM and build a boot loader on top of that. The boot loader nbc implements this idea, loading a Linux kernel or a standalone program over the network and booting from Linux using the kexec kernel patch. This solution works fine when 512KB of ROM (or more) is available. Unfortunately, most standard motherboards shipping today have only 256KB of ROM. For the x86 platform it is nearly impossible to get a useable Linux kernel under 360KB.

Various strategies have been developed to address these systems limited by the amount of available ROM. Some of these strategies include Tiara, which appears to be a complete firmware and boot loader for the SiS630 chipset; Etherboot, which has been ported to work under LinuxBIOS; RedBoot, which runs under LinuxBIOS but is not yet useable; and some hacks on LinuxBIOS itself.

Alpha firmware requires a standalone program to be familiar with the motherboard it is running on, which can be problematic. While having this level of familiarity is nice, supporting a new motherboard can be extremely difficult because of the number of pieces of software that must be updated. With LinuxBIOS we do our best to avoid that problem.

We start with the traditional x86 approach: initialize the Super-IO chips to working and expected values (i.e., serial ports at their expected legacy address, IRQ, etc.) and then provide IRQ routing tables and mptables for SMP.

For the long term we need a table tracking the capabilities Plug-and-Play has identified. This software lists what hardware is present and configures which

resources the hardware will use, or at a minimum it lists which resources an individual device uses. The solution I am working on involves creating a table of devices with information about how they are connected to each other on the motherboard. The table will list devices not currently participating in any kind of Plug-and-Play enumeration, as well as give enough information so that IRQ routing can be handled. Additionally the idea seems to fit in well with the struct device tree planned for the 2.5 kernel. ACPI appears to offer an alternative solution but it seems PC-centric, and the interpreted byte codes seem unnecessary and even dangerous.

Is LinuxBIOS Doable?

Since the first IBM PC, the hardware architecture of the boot ROM has evolved considerably, so that today virtually every machine has a BIOS that can be upgraded in the field or recovered from a failed upgrade. The common technique to accomplish this is to have a socketed Flash ROM on the motherboard. The Flash chip allows software to update it while the socket allows replacement of the chip if the update somehow fails. With this type of hardware architecture, developing custom boot firmware is now possible. For production machines, you can update firmware with no special hardware, and during development you can recover if something goes wrong.

A downside to current PC hardware architectures is that normally boot ROMs, at 256KB, are too small. This is enough space for firmware, but it isn't large enough for the Linux kernel.

The Linux kernel can run from LinuxBIOS as well as it does from a standard PCBIOS, when the port is done correctly. To date I successfully have ported LinuxBIOS to three motherboards. On the latest board, the results of booting Linux from LinuxBIOS and the PCBIOS are indistinguishable. So while there are significant technical hurdles in porting LinuxBIOS to new platforms, these can be and have been overcome.

Having access to adequate documentation is a nontechnical factor to consider. Convincing hardware vendors to support LinuxBIOS, or to release the documentation for someone else to code it, has met limited success to date. Missing or limited vendor support is not a new issue for free software, and it has been overcome in the past—now is not the point in the game to be discouraged. It is worth remembering that without these kinds of efforts there would be no new hardware on which we could run free software.

What Applications Are Available for LinuxBIOS?

Currently two different interest groups are working on LinuxBIOS: one working on embedded systems and one building large-scale computer clusters. For these applications the legacy x86 firmware is suboptimal.

LinuxBIOS has a lot to recommend itself for embedded applications. As it is released under the GPL, LinuxBIOS is royalty-free. LinuxBIOS generally weighs in under 64KB and doesn't waste ROM space with unnecessary functionality. Because it isn't a legacy design, LinuxBIOS starts up fast, even without code optimization.

In August 2001, General Software announced a 0.8-second boot to LILO on an embedded board after a hardware reset. This is a reasonable amount of time to do the job, but under LinuxBIOS such impressive results are routine. I can load the kernel over the network from a cold power-on in two seconds flat on a dual-processor server board—without optimizing LinuxBIOS.

The small footprint of LinuxBIOS has impressed SiS enough that they have devoted a developer to port LinuxBIOS to their chipsets, aiming at embedded applications. This demonstrates one well-supported platform.

For computer clusters, which is what Linux NetworX specializes in, LinuxBIOS has a lot to recommend itself as well. The serial port is the native console, so you don't need video hardware. Serial connections can be redirected easily into a terminal server for remote console access. The early setup of the serial console also brings benefits. For example, LinuxBIOS can report all errors and hardware failures over the serial console. A normal BIOS, even with serial console extensions, will initialize the serial port too late in the game for some failures to be detected, and it will usually fail if the CMOS is cleared.

LinuxBIOS also supports network booting on most hardware platforms, allowing changes to boot options to be made simply by altering a setting on a DHCP server. Since the code is open source, if the network booting policy is not to your liking it can be changed. The fast booting of LinuxBIOS means that if you are debugging something and have to reboot a node, the hardware doesn't waste the valuable time of the system administrator.

The openness of LinuxBIOS and its focus on Linux allow it to be configured and managed from user space under the Linux kernel. Anything done from user space also can be set up to be done remotely. This is a great advantage in homogeneous clusters, allowing firmware changes to be made and managed globally instead of one node at a time.

With large numbers of machines, the probability of hardware failure is much larger than for a single machine. The reduced hardware requirements of LinuxBIOS—such as unneeded floppy drives, CD-ROMs or hard drives to boot from, and no need for a video card and keyboard to control the system—can lead to a less expensive and more reliable system. Fewer hardware components lead to a reduced risk of hardware failures.

For clusters, LinuxBIOS also brings the potential to plug in to the cluster and, with nothing more than the firmware running, have a machine that acts as a single system, instead of a rack that looks like a collection of nodes.

What Hardware Does LinuxBIOS Support?

In the LinuxBIOS tree there are currently ports to 13 different mainboards. LinuxBIOS has been ported to both x86 and the Alpha hardware architectures. It has run on the AMD Athlon, AMD Duron, Pentium II, Pentium III, Alpha 211264 CPUs, ALI m1631, Digital Tsunami, AMD 760, AMD 760MP, Intel 440BX, Intel 440GX, VIA VT8601, SiS540, SiS550, SiS630 and SiS730 chipsets. And this is just the code that has been finished and placed in the main LinuxBIOS tree. Other ports are still in progress and haven't been committed as of this writing. So while the hardware support is limited, the list is growing. LinuxBIOS currently is not tied to any specific chipset, vendor or processor architecture.

The quality of the hardware support varies. On the chipset front, support for SiS chipsets is very good. Both Intel and AMD have a standard policy of documenting their chipsets so the support at both is pretty good. Via does not publicly document their newer chipsets, making support here a challenge.

On the processor front, Compaq has made the important details public, so supporting Alpha processors is doable. Development for the Alpha has not been a high priority, however, because the Alpha is an expensive processor with a dubious future.

The Pentium II and Pentium III are fairly well documented in Intel manuals, except for the L2 cache initialization of their slot processors (the L2 cache initialization is now supported). The AMD Athlon and Duron are not well supported because AMD hasn't publicly documented everything that needs to be set up for their processors.

Support from the board manufacturer isn't necessary because, most of the time, components on a motherboard can be identified by just looking at it.

Board manufacturers generally are interested in supporting only one firmware for their motherboards. As LinuxBIOS currently does not provide a compatibility layer for booting other operating systems besides Linux (notably

Windows), there hasn't been much interest from board manufacturers in deploying LinuxBIOS in its current form.

Conclusion

LinuxBIOS provides an innovative look at the job of firmware, how it is structured, written and licensed. As machines become increasingly integrated, LinuxBIOS is rising to meet the demand for greater code reuse and flexibility. If the snowballing interest in the technology is any indication, it looks to have a bright future.

Resources

email: ebiederman@lnxi.com

Eric Biederman (ebiederman@linuxnetworx.com) is a software engineer for Linux NetworX, focusing on LinuxBIOS and helping with cluster management tool development. When not deeply involved in LinuxBIOS, Eric reads science fiction, plays with DOSEMU and hikes around the Wasatch Mountains just outside of Salt Lake City.

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

LINUX JOURNAL

Enterprise JavaBeans

Reuven M. Lerner

Issue #92, December 2001

Reuven introduces an infrastructure for creating distributedapplications using JBoss, the Java application server.

As web applications become more serious, developers have become increasingly demanding about their tools. Over the last two months, we looked at two object-to-relational mapping tools (Alzabo and DODS) that make it possible to work with databases using object methods, avoiding the use of SQL inside of your program.

But there are a number of issues that many of these object-relational mapping tools fail to address: how can objects be separated onto different computers? Once separated, how can objects find each other? And if an object's state reflects the state of one or more database rows, how do we handle transactions?

These are messy and difficult questions, and we can expect to wrestle with them for years to come. One of the most comprehensive answers to these questions (and many others) is the J2EE (Java 2 Enterprise Edition) platform and its Enterprise JavaBeans object model. EJB, as it is known, is designed for use in complex, large-scale web sites and reduces the need for programmers to handle infrastructure issues.

This month, we will begin to look at EJB as implemented by the JBoss application server. JBoss is distributed under the GNU Lesser General Public License (LGPL), does not require much memory and is relatively easy to use. As is often the case with Java programming projects, working with EJB requires learning and working with a new set of tools, configuring several XML files in the right way and making sure that your CLASSPATH contains the right values during compilation and at runtime. If you can overcome these logistical hurdles, then JBoss provides an excellent basis for working with a powerful set of server-side technologies.

Java, Enhydra and Other Problems

Before we continue, it's important to stress that neither EJB nor the Java language are truly free software. While you might not have to pay to download Java or the J2EE libraries, Sun owns everything having to do with Java, including the specifications. Sun's Community Source License is more open than many other licenses, but it is far from an open-source license.

This is particularly evident now that Lutris Corporation, who sponsored the open-source Enhydra application server, has pulled the plug on its J2EE-certified Enhydra Enterprise server. Lutris has turned Enhydra Enterprise into a closed-source project, claiming that Sun's license makes it impossible to deliver a fully compliant, open-source J2EE server. There has been a great deal of anger (and defensiveness) on the main Enhydra e-mail list, and many unanswered questions remain. Perhaps Lutris was legally (and financially) obligated to do what they did, but the manner in which they did it is an example of how not to close down an open-source project.

Luckily, the JBoss team has made it clear that JBoss will continue to be an open-source project, and that it will continue to grow and support all of the J2EE standards even if it lacks the official J2EE certification, largely because of the money required to receive such approval from Sun.

What Is EJB?

A good first question to ask is, “Why would I need EJB?” And indeed, there are many applications for which EJB is overkill. However, EJB provides functionality that would be difficult for us to implement on our own, inside of the server or container as it is known:

- EJBs can reside on the same computer as your application, or on a remote computer. Thus, you can create multitiered applications in which each tier sits on a different computer, and your software continues to run unaltered as you move it from computer to computer or change the configuration of one or more tiers.
- The EJB container can handle object-relational mapping issues for you. You define the database tables and the objects that map to them, and the container can handle the rest. Or if you prefer to fine-tune things yourself, you can let your bean manage its own persistence layer.
- Relational databases provide transactions, allowing you to treat two or more operations as if they were a single operation. EJB gives your objects similar transactional capabilities, making it possible for a method to perform multiple actions as an all-or-nothing group.

It is also important to understand what EJB is not; despite the similar names, Enterprise JavaBeans have almost nothing to do with run-of-the-mill JavaBeans. JavaBeans have a standard API that allows us to access them from JSPs using little or no code. EJBs, by contrast, are designed to be used from any Java program, including servlets. Moreover, the standard API for EJBs is more rich, complex and flexible than that of JavaBeans. It's unfortunate that the term JavaBeans has been overloaded by these two popular server-side technologies, but there isn't anything we can do about it now.

One of the most compelling arguments for EJB is that the API is standard across application servers. Thus, you can begin working with an open-source EJB server such as JBoss, and then deploy on a commercial server when the time is ripe. (Although once you learn how much commercial servers cost, you may want to reconsider switching away from JBoss.)

Perhaps the most annoying part of working with Java is the great number of acronyms, project names and version numbers you must remember. This article works with the JDK (Java Development Kit) 1.3 and JBoss 2.4.1a server, which implements the EJB 1.1 standard. Moreover, while it is not particularly difficult to write the EJB classes themselves, the logistics associated with compiling and deploying them can be annoying and difficult for the uninitiated.

Working with EJBs

Basing your application on EJBs means moving as much of the business logic into separate objects as possible. In EJB, these objects come in two different flavors:

- Entity beans are objects that map to a relational database. Each instance of an entity bean typically corresponds to a single row in a database table. Each instance variable corresponds to a single column in the database table. We normally need to define a table in our database to correspond to our entity bean, but the EJB container writes and executes SELECT, INSERT, UPDATE and DELETE queries according to our needs.
- Session beans perform actions, by themselves or by using one or more entity beans. Session beans normally have no state of their own, which makes them more efficient than entity beans. However, there are times when it might help to keep some state around within a session bean. For this reason, EJB offers the stateful session bean, whose state is kept around between invocations.

If we were to create an on-line forum using EJBs, we probably would have to define entity beans (and the corresponding tables) for users, threads and postings. We also would have to define session beans that support adding,

modifying and deleting each of these types of entity beans and for retrieving entire threads and individual postings.

Installing JBoss

JBoss is a Java application server that makes it possible to create and deploy multitier J2EE applications. JBoss doesn't pretend to handle the application side of things; for that, you will need to use Jakarta-Tomcat or another servlet container. But JBoss does provide the back-end features, such as a directory service and messaging service, as well as the EJB container.

Installing JBoss is extremely easy, assuming that you already have installed the JDK. Sun provides a copy of the JDK in RPM format, which you can download from java.sun.com. You also will need to download and install the Ant utility, a Java program meant to replace the venerable UNIX make program. If you are familiar with make and XML, you will find Ant's build.xml format as shown in Listing 1 [available at ftp.linuxjournal.com/pub/lj/listings/issue92/5497.tgz] to be relatively straightforward.

Once you have installed the JDK and Ant, installing JBoss is a cinch. I downloaded the binary code from <j href="http://boss.org" target="_blank">boss.org, opting for the integrated JBoss and Jakarta-Tomcat support. The file comes as a zip archive, meaning that you'll need the Info-Zip utilities (which come with all Linux distributions that I've used) to unpack them.

When unpacked, the JBoss-Jakarta distribution contains two subdirectories, appropriately named jboss and tomcat. Set the JBOSS_DIST environment variable to point to the jboss directory, so that various JBoss-related utilities and features will be able to find the appropriate files.

At this point, you can start the JBoss server with the following two commands:

```
cd $JBOSS_DIST/bin
sh run.sh
```

By default, JBoss logs quite a bit of information to the terminal window.

Writing the Calculator Bean

Our first EJB will be Calculator, a stateless session bean whose multiply() method takes two integers and returns their product. After writing Calculator and its necessary EJB interfaces, we will see how we can use it from within a standalone Java program.

Writing a simple Calculator class with a multiply method normally would not be very difficult. We would create the file Calculator.java and define a method with the following signature:

```
public int multiply (int num1, int num2)
```

EJB allows us to find and invoke our Calculator bean remotely, which means that we must write several classes that make it possible to find Calculator. In the end, our application will manipulate a remote reference to the actual Calculator bean, rather than the object itself. Writing a session bean thus involves writing one Java class and two interfaces.

The Java class is the bean class itself, which performs the actual work. The bean class doesn't know that it has been invoked by an object on another computer; it can learn about its environment by querying its “context” but normally doesn't need to do that very much. The bean class typically is called the EJB's simple name, with the word Bean attached. The bean class for our Calculator EJB is thus CalculatorBean, defined in the file CalculatorBean.java. The bean class must implement either the SessionBean or EntityBean interface, depending on what type of bean it is.

The first interface is the remote interface, which allows the application to locate and get a reference to the Calculator EJB. The remote interface traditionally is given a simple name, such as Calculator and thus is defined in the file Calculator.java. The remote interface should define a method for each public method in the bean class. The remote interface must extend the EJBObject class.

The second interface is the home interface, which allows the EJB container to create, locate, destroy and otherwise manage an Enterprise JavaBean. The home interface is traditionally given the same name as the remote interface, with the word Home attached. The name of our EJB's home interface is thus CalculatorHome, which we define in the file CalculatorHome.java. The home interface must extend the EJBHome class.

One of the nice things about EJB is that your classes can rely on the default EJB behavior much of the time. This might not be the most efficient way to go about things, but it allows us to focus on writing the functional part of our code, allowing the EJB container to handle almost all of the infrastructure.

Writing the Classes

Now that we understand which classes we have to create, we can begin to write some code. You'll quickly notice that there is not that much code to write, and that in the case of our CalculatorBean class, many of our methods are defined

with empty bodies. This is because the SessionBean interface, from which CalculatorBean inherits, forces us to define these methods, even if our bean is simple enough not to use them. Using empty method bodies fulfills our obligations to the interface, while keeping our class simple.

I put all of the Java source files in the il.co.lerner.calculator package, reflecting the fact that they come from my commercial domain and that this is the calculator project. As such, all of the .java source files are in a directory hierarchy il/co/lerner/calculator.

Our bean class, CalculatorBean (see Listing 2), defines a single multiply() method, which takes two integer inputs and returns an integer to its caller. Other than implementing the SessionBean interface, CalculatorBean really doesn't have much to do with EJB; indeed, it is a fairly boring class with a single method. Anything we write to System.out will be printed to the JBoss session log.

Listing 2. CalculatorBean.java, the Bean Class for Our EJB Calculator

Our home interface, CalculatorHome, allows us to create a new instance of CalculatorBean. Other than defining the interface's signature, including the fact that it returns an instance of the remote interface (Calculator), the home interface is extremely short:

```
package il.co.lerner.calculator;
import java.io.Serializable;
import java.rmi.RemoteException;
import javax.ejb.CreateException;
import javax.ejb.EJBHome;
public interface CalculatorHome extends EJBHome
{
   Calculator create() throws RemoteException,
   CreateException;
}
```

Finally, our remote interface, Calculator, lists one method signature for each public method in CalculatorBean:

```
package il.co.lerner.calculator;
import javax.ejb.EJBObject;
import java.rmi.RemoteException;
public interface Calculator extends EJBObject
{
    public int multiply(int num1, int num2) throws
    RemoteException;
}
```

Client programs will be invoking methods via the remote interface, rather than directly on the bean. The signatures for the remote interface and the bean class must match, or you will encounter serious problems later on.

Deploying the Bean

Now that we have defined them, we can deploy our Calculator session bean to our running JBoss server. Deploying our session bean means taking all of its elements and turning them into a single Java archive (jar) file. Our .jar file will contain the compiled classes for Calculator, CalculatorHome and CalculatorBean.

But, it will also contain a “deployment descriptor”, an XML file named ejb-jar.xml that describes the contents of the .jar file to the EJB container. Deployment descriptors are a mandatory part of the EJB standard and do not vary from one application server to another. They tell the EJB container the names of the interfaces and classes that we have chosen, and also allow us to define such items as the type of transactions our bean will support. The deployment descriptor for our Calculator EJB is in Listing 3 and should be placed in the same directory as the .java source files.

Listing 3. ejb-jar.xml, the Deployment Descriptor for Our Calculator Bean

Our .jar file will also contain a short XML file named jboss.xml, which we will place alongside ejb-jar.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss>
  <enterprise-beans>
    <session>
      <ejb-name>Calculator</ejb-name>
      <jndi-name>calculator/Calculator</jndi-name>
    </session>
  </enterprise-beans>
</jboss>
```

The jboss.xml file is specific to JBoss, binding our bean to the Java's Naming and Directory Interface (JNDI). With jboss.xml in place, a client program that asks JNDI for calculator/Calculator will get a reference to it in return.

We could build the .jar file by hand, but it's easier to use Ant to build the .jar file and deploy it into the right place. Listing 1 [ftp.linuxjournal.com/pub/lj/listings/issue92/5497.tgz] contains an Ant build.xml that supports the targets ejb-jar (the default) and deploy. If you place build.xml in $CALCULATOR, then your .java files, ejb-jar.xml and jboss.xml should be in $CALCULATOR/il/co/lerner/calculator. Ant will place the results of compilation in $CALCULATOR/build/calculator, as specified in the build.calculator.dir property in build.xml.

With Ant installed in $ANT, we can compile our .java files, turn them into an EJB-compliant .jar file (with the ejb-jar.xml file in the mandatory META-INF directory) and deploy it to JBoss with the following command:

```
$ANT/bin/ant deploy
```

You should see a number of messages on the screen describing the compilation and deployment process. If the compilation or build fails, check that your environment variables are set correctly, that the Java files don't have any syntax errors and that the directories have appropriate permissions.

If your JBoss server already is running before you deploy the Calculator .jar file, you will notice that it automatically detects and deploys the file without any need for restart. This is one of the great pleasures of JBoss; in order to deploy your EJB .jar file, you simply copy it into the $JBOSS_DIST/deploy directory.

Writing an Application

Now that we have deployed our Calculator EJB, let's write a short Java program that uses it. Listing 4 contains the source code for such a class, UseCalculator.java.

Listing 4. UseCalculator.java, Which Connects to and Uses Our Calculator EJB

While our program is completely independent from our EJB classes and can be compiled and run separately (or even on a separate computer), we use Ant to keep track of the CLASSPATH (which must include the JBoss classes, as well as those from our .jar file), compile our code and then run it. In order to run our application, we simply can say

```
$ANT/bin/ant use-calculator-ejb
```

This runs our program after ensuring that our EJB is compiled, turned into a .jar file and deployed.

Anything that UseCalculator.main() writes to System.out (also known as the stdout filehandle) is printed on the screen when we run Ant. However, anything that our CalculatorBean method writes to stdout is printed to the JBoss logging output. By keeping JBoss open in one terminal window and running Ant in another, we can see them communicate with each other.

UseCalculator's main() method consists of several standard steps for connecting to and using our EJB. We first connect to JNDI, which keeps track of the objects currently deployed to JBoss. This connection is known as a context. Our program looks for jndi.properties, a short Java properties file that tells it where it can go to find a context (this file should be placed in $CALCULATOR/resources/, as specified in build.xml). This file is in Java resources format, where every line contains name=value:

```
java.naming.factory.initial=
java.naming.provider.url=localhost:1099
java.naming.factory.url.pkgs=org.jboss.naming:org.jnp.interfaces
```

Once we have our context, we look up our object using the name that we gave it in jboss.xml, which is inside of our ejb-jar.xml. Without jboss.xml, JBoss will not associate the right name with our EJB, making it impossible to find using JNDI.

JNDI returns an object reference, which we then cast into an instance of CalculatorHome, which is then used to create an instance of Calculator. Notice how we create an instance of Calculator (the remote interface), rather than one of CalculatorBean. The remote interface provides us with a transparent connection to an instance of CalculatorBean on the server, wherever that might be. At no time do we actually know where the real instance of CalculatorBean resides.

Finally, we invoke one of the methods that has been defined in Calculator (the remote interface). Our method invocation is passed along to CalculatorBean (the bean class), where it executes (and prints out some logging information) and returns (where we print the result to stdout).

Conclusion

This month we started to look at Enterprise JavaBeans, an infrastructure for creating distributed applications using Java. While EJB is far more complex than SOAP, XML-RPC or other distributed object systems, it is also designed to handle more complicated tasks. (For example, SOAP doesn't attempt to handle transactions; that's left to the application layer to implement.)

At the same time, working with Java often means spending more time on administrative and logistical issues, rather than on programming. Determining which file must be in which directory can often be frustrating, especially if you are used to working with a more dynamic language such as Perl or Python. Nevertheless, the pain quickly subsides when you see how easily you can create distributed applications with EJB. The fact that JBoss is so easy to download, install and run, and has a very small memory footprint, makes it simple for newcomers to try EJB.

Next month, we will continue working with EJB, looking at the heart of EJB, the entity beans that provide an object interface to our relational databases.

Resources

email: reuven@lerner.co.il

Reuven M. Lerner owns a small consulting firm specializing in web and internet technologies. He lives with his wife Shira and daughter Atara Margalit in

Modi'in, Israel. You can reach him at reuven@lerner.co.il or on the ATF home page, www.lerner.co.il/atf.

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

Lighter Admin Fare with Depth

Marcel Gagné

Issue #92, December 2001

Marcel shows you how to view the strange world of Linux in exciting new ways using fsv, 3dfile and XCruise.

François, *mon ami*, I am impressed. When I asked you to select a wine with depth, I must admit that while I had complete confidence in you, this time I am truly pleased. The 1997 *Volnay-Santenots du Milieu* is a wonderful choice. Please make sure that you bring back enough for our guests. I am expecting them shortly. *Vite*, François.

But they are already here! *Bienvenue, mes amis*. Welcome to *Chez Marcel*, home of fine Linux cuisine. Please sit. François is down in the cellar fetching the wine. You know, *mes amis*, I constantly am amazed when I think of the parallels between good food and wine and cooking with Linux. Take this month's issue for instance...ah, François, you have made good time. Please open the wine and serve our guests. *Merci*.

Just as a good restaurateur becomes one with their wine cellar, so does a good administrator become one with their system. Every directory and every file is as familiar to the system administrator as their own hand. Of course, in terms of visualization, I have had an advantage with the wine cellar. From time to time, I wander its cool halls, letting its bottles, labels and smells fill my senses. However, as Linux system administrators, we have been forced to imagine this world. But today, *mes amis*, all that changes. Today, your Linux system becomes a *place*. With the help of a few open-source recipes, your filesystem is about to enter the third dimension.

The first item on today's menu is a little something innocently called fsv (filesystem viewer), a concoction by Daniel Richard. When you fire up this little program, it scans your entire filesystem (relative to your position in the filesystem's hierarchy) and provides you with a new and unique view of what

you will find there. Suddenly, files and directories appear as cubes of varying heights like buildings in a strange city. Click on a directory and you can zoom in to that directory and explore the "city" beneath it. It offers two different visualization modes: a map view and a tree view. The tree view is interesting in that directories now appear like skyscrapers in a fantastic, cyber city. Figure 1 shows fsv in action.

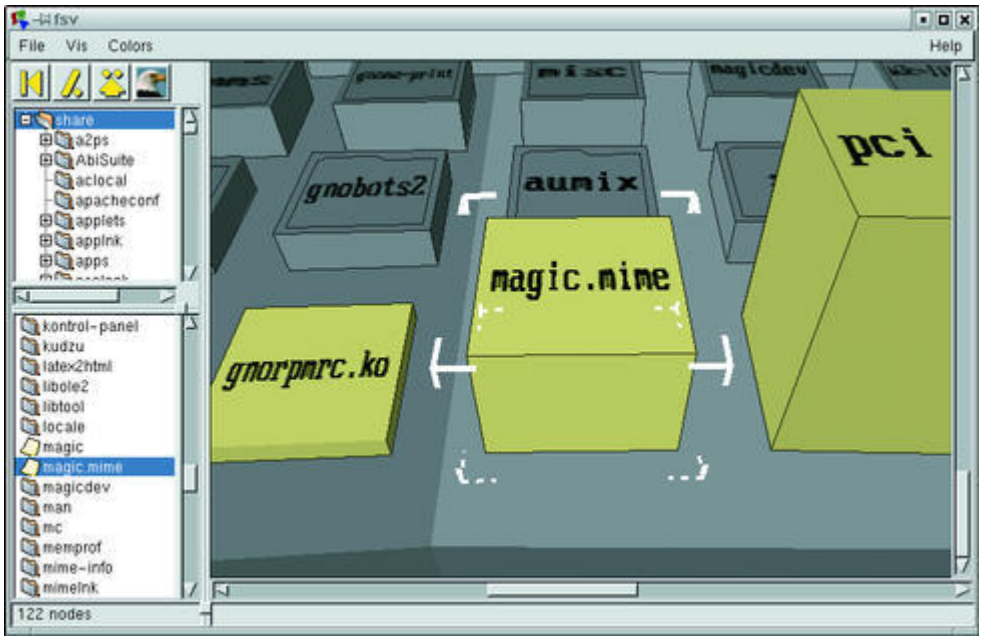


Figure 1. This Is the City—Your Filesystem in fsv

You can get fsv from fsv.sourceforge.net. You will need a couple of things to start, such as OpenGL or Mesa, GTK+ (both of which are probably already installed on your system or are as close as your distribution CD), as well as Janne Löf's GtkGLArea OpenGL widget set for GTK+, available at www.student.oulu.fi/~jlof/gtkglarea. If you are running an RPM-based distribution, you also can find various precompiled packages on www.rpmfind.net. But I digress, *non*?

Extract the source and build as follows:

```
tar -xzvf fsv-0.9.tar.gz
cd fsv-0.9
make
make install
```

In order to run the program, you simply type **fsv**. Depending on your starting directory, this may take a few seconds. You also can specify the starting directory on the command line if you wish. For instance, if you wanted to start at your Linux kernel source, you would type the following:

```
fsv /usr/src/linux
```

Daniel is not the only one with a three-dimensional vision of his Linux system. Another interesting project is aptly named 3dfile. Once again, the 3-D visualization comes to us courtesy of OpenGL or Mesa and Mircea Mitu.

As Mircea Mitu is quick to point out, 3dfile is not quite a file manager, although it does do some pretty cool things (besides rocking representational icons back and forth). For instance, if you click on a red block (these represent executable files), you will launch that application. I fired up a game of *Maelstrom* that way and wasted far too much time. Here's something else to try: right-click on a filename and a small pop-up menu will appear. If you happen to be looking at a text file, you can view the file as text. The other viewing option is hex, a great way to impress your nontechnical friends. Open up a file in the hex viewer, study it carefully and say something like “Ah, there's the problem.” System administrators need to keep the magic alive, *non*?

Have a look at Figure 2, and then we shall toss this little recipe together and get 3dfile working for you.

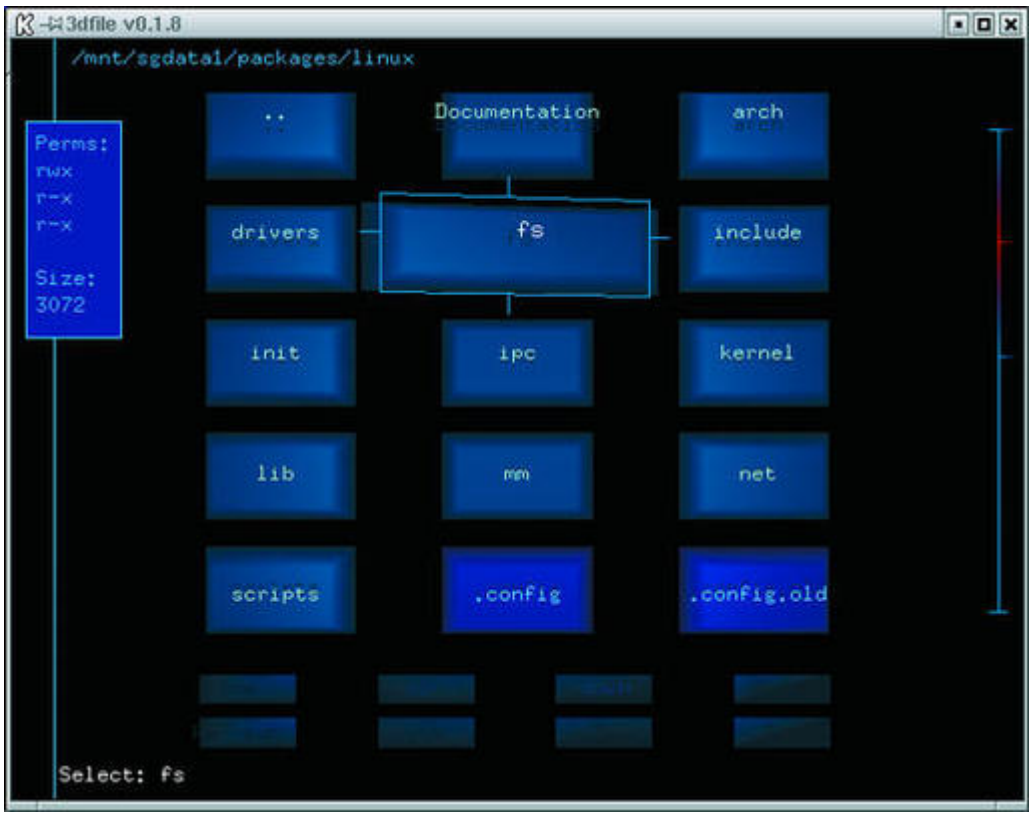


Figure 2. Rocking File Blocks with 3dfile

You'll need to visit web.ss.pub.ro/~mms/3dfile to pick up your copy of the source. The installation is quite straightforward as well, but does have one interesting first step before the nearly traditional ./configure:

```
tar -xzvf 3dfile-0.1.8.tar.gz
cd 3dfile.0.1.8
./autogen.sh
```

```
./configure
make
make install
```

Running the program is as easy as typing **3dfile**. I found working with 3dfile to be quite intuitive. The only thing that may not be immediately apparent is that row of buttons at the bottom. Run your crosshairs over these and you'll find navigation even easier.

The next item on our menu offers another way of looking at the system. Leander Seige's TDFSB is interesting in that it lets you wander through an OpenGL-generated filesystem with a rather enjoyable twist. Directories hover as silver spheres upon a floating grid landscape. What I find really interesting here becomes apparent when you happen upon a graphic or image file. The image is rendered as a three-dimensional image around which you can move. Have a look at Figure 3 for a peek at TDFSB in action.

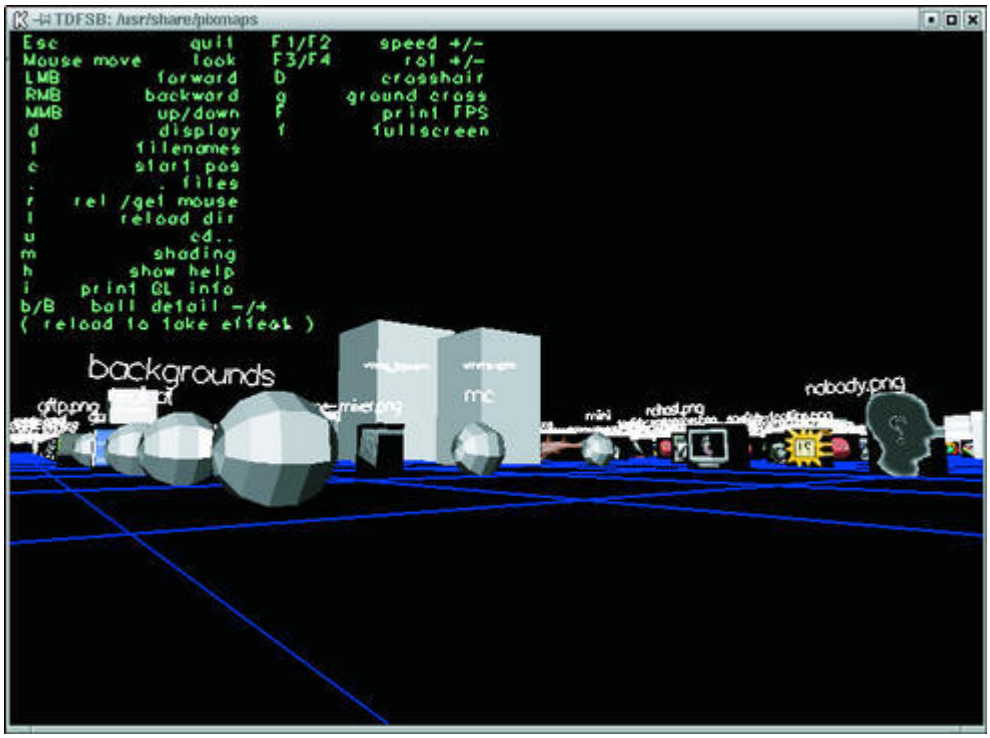


Figure 3. The 3-D Landscape of TDFSB Revealed

Building TDFSB is not complicated, but I warn you that Leander's source does not provide an elegant means of compiling the program. The README does contain just about everything you'll need to compile on your system, and it is very, very short. Please read it. In a moment, I will show you what I did to make it run on my system.

Start by visiting the TDFSB site and picking up the source. TDFSB's home is at www.hgb-leipzig.de/~leander/TDFSB. The only other thing I needed was the gdk-pixbuf image library (part of GNOME):

```
tar -xzvf tdfsb.tgz
cd tdfsb
```

This is where you have to look at your README. From that, I created the following, single-line install script. Please note that while the text wraps, it is one line. It essentially was created by cutting and pasting from that README file. Ah, yes, the system I tested this on was a notebook running Red Hat 7.1:

```
gcc -L/usr/X11R6/lib -I./ -lGL -lGLU -lglut -lXmu
    -lXi -lXext -lX11 -lm -lgdk_pixbuf -march=i686
    -malign-loops=4 -malign-jumps=4 -malign-functions=4
    -fomit-frame-pointer -fno-strength-reduce
    -x c -o tdfsb tdb3.c -I/usr/include/gdk-pixbuf
    -I/usr/include/glib-1.2 -I/usr/include/gtk-1.2
    -I/usr/lib/glib/include
```

Now, run the program by typing **./tdfsb**. To get a little floating, heads-up display help menu, type **h**. Since your mouse is the means by which you navigate this world, you can toggle its release by typing the letter **r**. The system on which I tested did not have 3-D acceleration hardware, but there is little doubt that you could benefit from a nice, fast video card when using TDFSB.

Ah, I see that closing time is approaching. But before you go, allow me to share a final and quite different way of looking at your systems. Those friends of ours who are new to Linux have, from time to time, been heard to remark that Linux is like a strange new world. The tools we have just explored demonstrate some of the many fascinating faces this world can have.

But “world”? How about solar system? How about galaxy? This is without a doubt what Yusuke Shinyama had in mind when creating XCruise. This great little package is a file manager of sorts, one that lets you navigate your disk as though it were a universe all its own. Each filesystem is a galaxy, files are stars (the bigger the file, the bigger the star) and symbolic links are wormholes. Your humble chef and his faithful waiter have been known to spend a great deal of time navigating their respective Linux systems in this way.

Anxious to launch into your own private universe? Have a look at Figure 4, then cruise on over to www.unixuser.org/~euske/pub/.

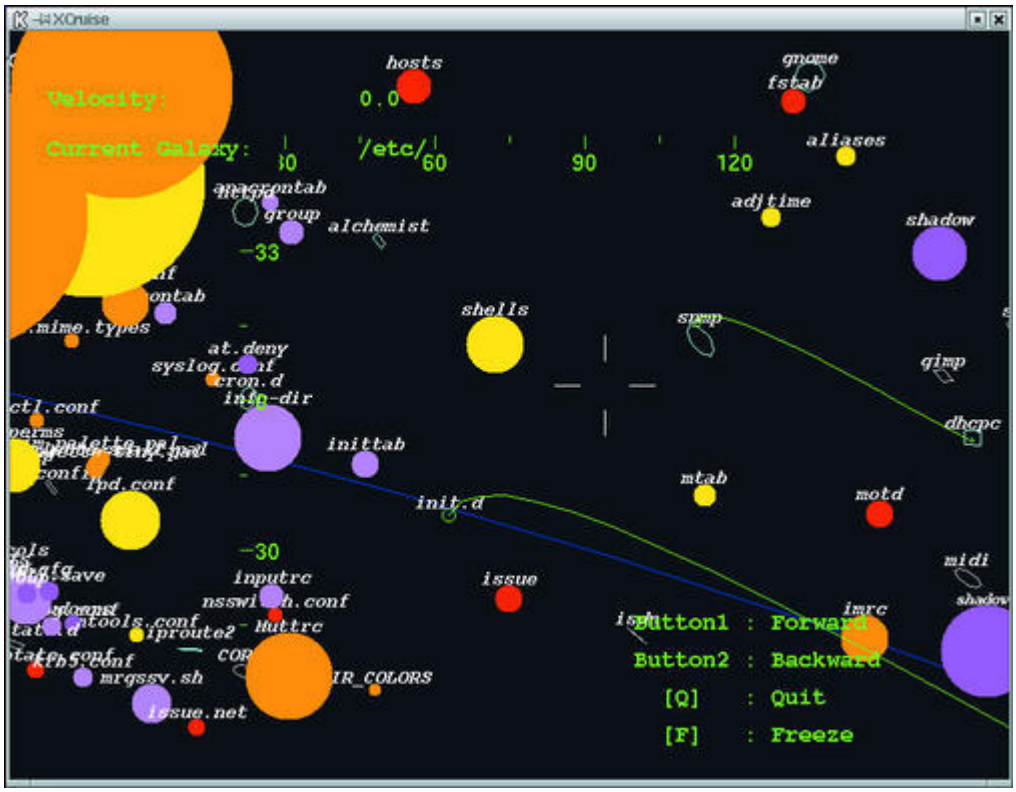


Figure 4. Cruising the /etc Galaxy with XCruise

Building the package is simplicity itself. Just follow this basic recipe and you are almost ready for launch:

```
tar -xzvf xcruise-0.24.tar.gz
cd xcruise-0.24
xmkmf
make
```

The resulting binary is sitting there in the directory and can be copied to wherever you like. To run the program, type **./xcruise &**. You may find that the default incarnation of XCruise is a bit large for your screen. Luckily, you also can use X geometry settings to change the size to something more suitable to your display. For instance, my notebook has an 1024 x 768 display, but I only want to have XCruise fill an 800 x 600 area. To do this, I start the program this way:

```
./xcruise -geometry 800x600+0+0
```

Magnifique! Using my left mouse button, I travel forward at incredible speeds while the middle button (or both buttons on a two-button mouse) lets me move out and back. The scroll keys let me vary my angle and direction. Ah, this Linux universe is rich and exciting, *non*?

Ah, *mes amis*. Once again, it is closing time. Time for François to fill your glasses a final time and for us to return to the real world, a world, that after today, you may never see the same way again. Until next time; *au revoir*, *mes amis*. Your table will be waiting.

A votre santé! *Bon appétit*!

Resources

email: mggagne@salmar.com

Marcel Gagné (mggagne@salmar.com) is president of Salmar Consulting, Inc., a systems integration and network consulting firm and the author of Linux System Administration: A User's Guide, published by Addison-Wesley.

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

LINUX JOURNAL

syslog Configuration

Mick Bauer

Issue #92, December 2001

Making sure your system processes and critical applications log the events and states you're interested in with syslog.

Whatever else you do to secure a Linux system, it must have comprehensive, accurate and carefully watched logs. Logs serve several purposes. First, they help us troubleshoot virtually all kinds of system and application problems. Second, they provide valuable early warning signs of system abuse. And third, when all else fails (whether that means a system crash or a system compromise), logs provide us with crucial forensic data.

This article is about making sure your system processes and critical applications log the events and states you're interested in. The tried-and-true tool for achieving this is syslog. **syslog** accepts log data from the kernel (by way of klogd), from any and all local processes, and even from processes on remote systems. It's flexible as well, allowing you to determine what gets logged and where it gets logged. A preconfigured syslog installation is part of the base operating system in virtually all variants of UNIX and Linux.

This month, therefore, we discuss syslog configuration and use it in-depth, probably in much greater detail than you've previously considered. In my experience the vast majority of Linux users, and even administrators, tend to leave their syslog installations with default settings, tweaking them little if at all. This is seldom a good idea.

I should also mention that if you're really interested in granular, flexible logging, Balazs Scheidler's excellent syslog-ng (syslog, new generation) is well worth checking out. But it's still nowhere near as ubiquitous as syslog, so I won't do more than mention it this time. See the Resources section for more information on syslog-ng.

What about klogd?

Configuring syslog

Whenever syslogd, the syslog dæmon, receives a log message, it acts based on the message's type (or facility) and its priority. **syslog**'s mapping of actions to facilities and priorities is specified in /etc/syslog.conf. Each line in this file specifies one or more facility/priority selectors followed by an action. A selector consists of a facility or facilities and a (single) priority.

In the following syslog.conf line, mail.notice is the selector and /var/log/mail is the action (i.e., “write messages to /var/log/mail”):

```
mail.notice     /var/log/mail
```

Within the selector, “mail” is the facility (message category) and “notice” is the level of priority.

Facilities

Facilities are simply categories. Supported facilities in Linux are auth, authpriv, cron, dæmon, kern, lpr, mail, mark, news, syslog, user, UUCP and local0 through local7. Some of these are self-explanatory, but of special note are:

- auth: used for many security events.
- authpriv: used for access-control-related messages.
- dæmon: used by system processes and other dæmons.
- kern: used for kernel messages.
- mark: messages generated by syslogd itself that contain only a timestamp and the string “--MARK--”. To specify how many minutes should transpire between marks, invoke syslogd with the -m [minutes] flag.
- user: the default facility when none is specified by an application or in a selector.
- local7: boot messages.
- *: wildcard signifying “any facility”.
- none: wildcard signifying “no facility”.

Priorities

Unlike facilities, which have no relationship to each other, priorities are hierarchical. Possible priorities in Linux are (in increasing order of urgency): debug, info, notice, warning, err, crit, alert and emerg. Note that the urgency of a given message is determined by the programmer who wrote it; facility and priority are set by the programs that generate messages, not by syslog.

As with facilities, the wildcards “*” and “none” also may be used. Only one priority or wildcard may be specified per selector. A priority may be preceded by either or both of the modifiers “=” and “!”.

If you specify a single priority in a selector (without modifiers), you're actually specifying that priority plus all higher priorities. Thus the selector mail.notice translates to “all mail-related messages having a priority of notice or higher”, i.e., having a priority of notice, warning, err, crit, alert or emerg.

This behavior can be canceled by prepending an = to the priority. The selector mail.=notice translates to “all mail-related messages having a priority of notice”. Priorities may also be negated: mail.!notice is equivalent to “all mail messages except those with priority of notice or higher”, and mail.!=notice corresponds to “all mail messages except those with the priority notice”.

Actions

In practice, most log messages are written to files. If you list the full path to a filename as a line's action in syslog.conf, messages that match that line will be appended to that file. (If the file doesn't exist, syslog will create it.) In the syslog.conf line above, we instructed syslog to send matched messages to the file /var/log/mail.

You can send messages other places too. An action can be a file, a named pipe, a device file, a remote host or a user's screen. Pipes usually are used for debugging. Device files people use tend to be TTYs, but some people also like to send security information to /dev/lp0, that is, to a local line printer. Logs that have been printed out can't be erased or altered by an intruder, and this is an excellent use for old dot-matrix printers.

Remote logging is one of the most useful features of syslog. If you specify a hostname or IP address preceded by an “@” sign as a line's action, messages that match that line will be sent to that remote host. For example, the line

```
*.emerg          @mothership.mydomain.org
```

will tell syslogd to send all messages with emerg priority to the host named mothership.mydomain.org. Note that the remote host's (in this example, mothership's) syslogd process will need to have been started with the -r flag in order for it to accept your log messages. By default, syslogd does not accept messages from remote systems.

If you run a central logserver, which I highly recommend, you'll want to consider having some sort of access controls on it for incoming messages. At

the very least you should consider TCPwrappers' hosts-access (source-IP-based) controls or maybe even local firewall rules (ipchains or iptables).

Stealth Logservers

More Sophisticated Selectors

You can list multiple facilities separated by commas in a single syslog.conf selector. To extend our original syslog.conf line to include both mail and UUCP messages (still with priority notice or higher), you could use this line:

```
mail,uucp.notice     /var/log/mail
```

The same is not true of priorities. Remember that only one priority or priority wildcard may be specified in a single selector.

You may, however, specify multiple selectors separated by semicolons. When a line contains multiple selectors, they're evaluated from left to right; you should list general selectors first, followed by more specific selectors. Think of selectors as filters: as a message is passed through the line from left to right, it passes first through coarse filters and then through more granular ones.

Continuing our one-line example, suppose we still want important mail and UUCP messages to be logged to /var/log/mail, but we'd like to exclude UUCP messages with priority alert. Our line then looks like this:

```
mail,uucp.notice;uucp.!=alert     /var/log/mail
```

Actually, syslogd's behavior isn't as predictable as this may imply. Listing selectors that contradict each other or that go from specific to general, rather than the other way around, can yield unexpected results. Therefore, it's more accurate to say that for best results, list general selectors to the left and their exceptions (and/or more specific selectors) to the right.

Wherever possible, keep things simple. You can use the logger command to test your syslog.conf rules (see the “Testing System Logging with logger” section toward the end of this article).

Note that in the second selector (uucp.!=alert) we used the prefix “!=” before the priority to signify “not equal to”. If we wanted to exclude UUCP messages with priority alert and higher (i.e, alert and emerg), we could omit the “=”:

```
mail,uucp.notice;uucp.!alert      /var/log/mail
```

You might wonder what will happen to a UUCP message of priority info; this matches the second selector, so it should be logged to /var/log/mail, right?

Based on the above examples, it won't. Since the line's first selector matches only mail and UUCP messages of priority notice and higher, such a message wouldn't be evaluated against the second selector.

There's nothing to stop you from having a different line for dealing with info-level UUCP messages, however. You even can have more than one line deal with these if you want. Unlike a firewall rule-base, each log message is tested against all lines in /etc/syslog.conf and acted on as many times as it matches.

Suppose we want emergency messages broadcast to all logged-in users as well as being written to their respective application logs. We could use something like the sample shown in Listing 1 to achieve this. Note the “-” sign in front of the write-to-file actions. This tells syslogd not to synchronize the specified log file after writing a message that matches that line.

Listing 1. A Sample syslog.conf File

Skipping synchronization increases the chances of introducing inconsistencies such as missing or incomplete log messages into those files, but it decreases disk utilization and thus improves performance. Use the minus sign, therefore, in lines that you expect to result in frequent file writes.

In Listing 1 we see some useful redundancy. Kernel warnings plus all messages of error-and-higher priority, except authpriv messages, are printed to the X-console window. All messages having priority of emergency and higher are printed there too and are also written to the screens of all logged-in users.

Furthermore, all mail messages and kernel messages are written to their respective log files. All messages of all priorities (except mail messages of any priority) are written to /var/log/messages.

The previous examples were adapted from the default syslog.conf that SuSE 7.1 put on one of my systems. But why isn't such a default syslog.conf file fine the way it is? Why change it at all?

Maybe you needn't, but probably you should. In most cases default syslog.conf files either assign to important messages at least one action that won't bring those messages to your attention effectively (e.g., by sending messages to a TTY console on a system you only access via SSH), or they handle at least one type of message with too much or too little redundancy to meet your needs.

Table 1 summarizes syslog.conf syntax, facility values, severity values and action types. Note that the three main columns of this table are independent; there's no correlation between facilities, severities and actions, i.e., a message may be sent to any facility with any severity and have any allowed action

performed on it. Note also that the numeric facility and severity codes are provided strictly for reference; you should not use these in syslog.conf, but you may come across them in source code or in a packet-dump of network traffic.

Table 1. Summary of syslog.conf Usage and Values

Running syslogd

Just as the default syslog.conf may not meet your needs, the default startup mode of syslogd may need tweaking. Table 2 and subsequent paragraphs touch on some syslogd startup flags that are particularly relevant to security, but for a complete list you should refer to the man page sysklogd(8).

Table 2. syslogd Startup Flags

In addition, note that when you're changing and testing syslog's configuration and startup options, it usually makes sense to start and stop syslogd and klogd in tandem (see the Sidebar “What about klogd?” if you don't know what klogd is). Since it also makes sense to start and stop these the same way your system does, I recommend that you use your system's syslogd/klogd startup script.

On most Linux systems this startup script is either in /etc/init.d/syslog or /etc/init.d/sysklog (sysklog is shorthand for “syslog and klogd”).

The first syslogd flag we'll discuss is the only one used by default by Red Hat 7.x in its /etc/init.d/syslog script: -m 0, which disables mark messages. These messages contain only a timestamp and the string “--MARK--”, which some people find useful for navigating lengthy log files. Others find them distracting and redundant, given that each message has its own timestamp anyhow.

To turn mark messages on, specify a positive nonzero value after -m that tells syslogd how many minutes should pass before it sends itself a mark message. Remember that mark has its own facility called, predictably, mark, and you must specify at least one selector that matches mark messages (such as mark.*, which matches all messages sent to the mark facility, or *.*, which matches all messages in all facilities).

For example, to make syslogd generate mark messages every 30 minutes and record them in /var/log/messages, you first would add a line to /etc/syslog.conf similar to

```
mark.*              -/var/log/messages
```

You would then need to start syslogd as shown here:

```
mylinuxbox:/etc/init.d# ./syslogd -m 30
```

Another useful syslogd flag is -a [*socket*]. This allows you to specify one or more sockets in addition to /dev/log for syslogd to accept messages from.

If you've ever secured a nameserver running BIND, you may have used -a to allow a chroot-ed named process to bounce its messages from a dev/log device file in the chroot jail to the non-chroot-ed syslogd process. In such a case, since named can't access /dev/log, it has its own, for example, /var/named/dev/log. You therefore need a line in /etc/init.d/syslog like this:

```
daemon syslogd -m 0 -a /var/named/dev/log
```

Note that the dæmon function at the beginning of this line is unique to Red Hat's init script functions; the important part here is

```
syslogd -m 0 -a /var/named/dev/log
```

More than one -a flag may be specified, like this:

```
syslogd -a /var/named/dev/log -a /var/otherchroot/dev/log -a /additional/dev/log
```

Continuing down the list of flags in Table 2, suppose you need to test a new syslog configuration file named syslog.conf.test but prefer not to overwrite /etc/syslog.conf, which is where syslogd looks for its configuration file by default. Use the -f flag to tell syslogd to use your new configuration file:

```
mylinuxbox:/etc/init.d# ./syslogd -f ./syslog.conf.test
```

We've already covered use of the -r flag, which tells syslogd to accept log messages from remote hosts, but we haven't talked about the security ramifications. On the one hand, security clearly is enhanced when you use a centralized logserver or do (almost) anything else that makes it easier for you to manage and monitor your logs.

On the other hand, you must take different threat models into account. Are your logs sensitive? If log messages traverse any untrusted network, and if the inner workings of the servers that send those messages are best kept secret, then in fact the risks may outweigh the benefit (at least, the specific benefit of syslogd's unauthenticated clear-text remote logging mechanism).

If this is the case for you, you definitely should consider using syslog-ng. **syslog-ng** can send remote messages via the TCP protocol and therefore can be used in conjunction with stunnel, ssh and other tools that greatly can enhance its security. Since syslog uses only the connectionless UDP protocol for remote logging and, therefore, can't tunnel its messages though stunnel or ssh, syslog is inherently less securable than syslog-ng.

If your log messages aren't sensitive (at least, the ones you send to a remote logger), then there's still the problem of denial-of-service and message-forgery attacks. If you invoke syslogd with the -r flag, it will accept all remote messages without performing any checks whatsoever on the validity of the messages themselves or on their senders. Again, this risk is most effectively mitigated by using syslog-ng.

One tool you can use with syslog to mitigate partially the risk of invalid remote messages is TCPwrappers. Specifically, TCPwrappers' hosts-access authentication mechanism provides a simple means of defining which hosts may connect, via which protocols, to your logserver. Hosts-access authentication is tricked easily by source-IP spoofing (especially since syslog transactions are strictly one-way), but it's better than nothing and is probably sufficient to prevent mischievous but lazy attackers from interfering with syslog.

If you're willing to bet that it is, obtain and install TCPwrappers (all modern Linux distributions have a binary package of it; some even install it by default), and refer to its hosts_access(5) man page for details. Note that despite its name, TCPwrappers' hosts access can be used to control UDP-based applications.

Testing System Logging with logger

Before we leave the topic of system-logger configuration and use, we should cover a tool that can be used to test your new configurations regardless of which log dæmon you use. **logger** is a command-line application that sends messages to the system logger. Besides its relevance as a diagnostic tool, logger especially is useful for adding logging functionality to shell scripts.

The usage we're interested in here is diagnostics. (Although, come to think of it, you really should use this tool in any important scripts you routinely run, especially ones that run unattended via cron or at.) The easiest way to explain how to use logger in this regard is with an example.

Suppose you've reconfigured syslog to send all dæmon messages with priority “warn” to /var/log/warnings. To test the new syslog.conf file, you'd first restart syslogd and klogd, then you'd enter a command like this:

```
mylinuxbox:~# logger -p daemon.warn
"This is only a test."
```

As you can see, logger's syntax is simple. The -p parameter allows you to specify a facility/priority selector. Everything after this selector, and any other parameters or flags, is taken to be the message.

Because I'm a fast typist, I often use while-do loops in interactive bash sessions to run impromptu scripts (actually complex command lines). The following sequence of bash commands works either interactively or in a script:

```
mylinuxbox:~# for i in
> do
> logger -p daemon.$i "Test daemon message, level $i"
> done
```

This sends test messages to the dæmon facility for each of all eight priorities. Listing 2, presented in the form of an actual script, generates messages for all facilities at each priority level.

Listing 2. Generating Messages for All Facilities at Each Priority Level

Conclusion

Hopefully that's enough to get you started in building, testing and using custom syslog configurations. May your logs be detailed, plentiful, closely watched and uninteresting!

Resources

email: mick@visi.com

Mick Bauer (mick@visi.com) is a network security consultant in the Twin Cities area. He's been a Linux devotee since 1995 and an OpenBSD zealot since 1997, and enjoys getting these cutting-edge OSes to run on obsolete junk.

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

LINUX JOURNAL

Mainstream Linux

Robin Rowe

Issue #92, December 2001

Robin examines the ever-increasing presence of Linux in theserver, workstation and desktop markets.

Linus Torvalds says in his entertaining autobiography *Just for Fun* that when he started being asked if he thought Linux would someday take over the desktop and make a dent in Microsoft, he invariably replied that it would. That prediction hasn't come true yet, but there's growing reason to think it may. For insight into the growing Linux client OS business we'll talk with IBM, HP, Compaq, Dell, Gateway and SGI.

Linux first succeeded as a server platform running Apache, so much so that IDC reports that last year 27% of server operating systems sold were Linux, compared to 41% for Windows. But, last year Windows scored 91% of the desktop OS market, with Linux just at 1.4%. Apple continued its slide to 3.6%. Only Windows and Linux are showing growth as desktop operating systems: Windows up by 11%, Linux up by 25%. However, IDC is collecting new numbers to take into account the many free copies of Linux being installed and to count pirated copies of Windows, not just copies sold of each OS. IDC analyst Al Gillen notes, “The most common hardware for Linux systems is a recycled Windows PC.”

As a free operating system it may seem odd that today Linux is a greater commercial force in high-end workstations than in desktop PCs. Linux is migrating down from servers, not up from entry-level PCs. Reasons for this include Linux compatibility with other UNIX operating systems that dominate high-end platforms and the barrier of growing a new support infrastructure rather than using the established one for Windows. Note that you won't see any OEM dual-boot Windows/Linux systems because that is precluded by Microsoft's secret licensing terms with vendors, a point that the government failed to make in the antitrust case.

All the top PC manufacturers now offer Linux. Not long ago you needed to specify a secret Linux web page, but now you usually can drill down into Linux offerings from their home page. To jump straight to Linux see www.ibm.com/linux, www.hp.com/linux, www.compaq.com/linux or www.dell.com/linux. Let's take a look at what the leading PC vendors are offering with Linux.

IBM's PC line includes IntelliStation workstations, NetVista desktops and ThinkPad laptops. The desktop NetVista N2200l comes with Turbolinux. ThinkPad Series A and Series T are available with OpenLinux eDesktop 2.4. The ThinkPad T22 is the first Linux computer ever to ship a licensed software DVD player, the InterVideo LinDVD. Linux open-source DVD players are avoided by manufacturers due to the infamous DeCSS lawsuit. Most DVD movies are copy protected, and the US DMCA law prohibits software capable of removing copy protection.

“Workstation users are switching to Linux because it is fast, simple and reliable”, says IntelliStation workstation director of marketing Doug Oathout. Although Linux is displacing Windows on servers, he hasn't seen it making much impact on Windows workstations; instead Linux is displacing other flavors of UNIX. Many flavors of Linux are supported. “We test Caldera, SuSE, Turbolinux and Red Hat on all models of IntelliStation”, says Oathout. “A lot of that testing involves graphics cards. ATI, NVIDIA and Matrox have Linux drivers, but 3Dlabs currently does not. We're working on that.” A significant part of IBM's one-billion-dollar commitment to Linux goes into testing and driver development.

“The IBM workstation market is strong for electronic design, and geoscience is coming on strong”, says Oathout. Having products available for Linux from EDA (electronic design automation) industry leaders such as Cadence and Mentor are encouraging users to switch from HP-UX (HP's UNIX) and Solaris to Linux. In the financial sector, IBM is seeing a mixed environment of Linux and Windows. Linux desktop applications for trading are available, but Reuters and Bloomberg access requires Windows. Popular Linux movie-making tools include Maya, LightWave and Softimage. The primary Linux CAD applications are ANSYS, Nastran and Patran, all for finite element analysis. “Oil and gas applications are up and coming”, says Oathout. “They will be converting over to Linux next year.”

CATIA is a Windows-based CAD system sold by IBM and developed in France by Dassault Systèmes. “CATIA V5 runs successfully on Linux”, says media relations spokesman Anthony Marechal. “From a marketing standpoint, we have made no decision so far but remain open to support CATIA running on Linux if the market pressure increases.”

Hewlett-Packard announced in September 2001 that it would acquire Compaq for $25 billion sometime in 2002. After the merger, the new company, with a combined revenue of $87 billion (HP $47 billion, Compaq $40 billion), will be close in market size to the $90 billion of IBM. Together HP and Compaq account for 75% of retail PC sales in US stores.

“We've been shipping Linux 3-D high-end workstations for about a year and a half”, says director of marketing Mike Balma. “Linux for digital content creation, Hollywood, is really showing growth thanks to support of industry standard applications like Maya, Houdini and Shake. In fact, Maya is only certified for Linux on HP workstations.” HP worked closely with DreamWorks in the Linux transition there (see this column in the August 2001 issue of *Linux Journal*). The HP workstation market is expanding in EDA, software development in telecommunications and large-screen financial trader applications. HP workstation buyers seem to be mainly former Solaris customers. As at IBM, Linux doesn't seem to be impacting the Windows workstation market yet.

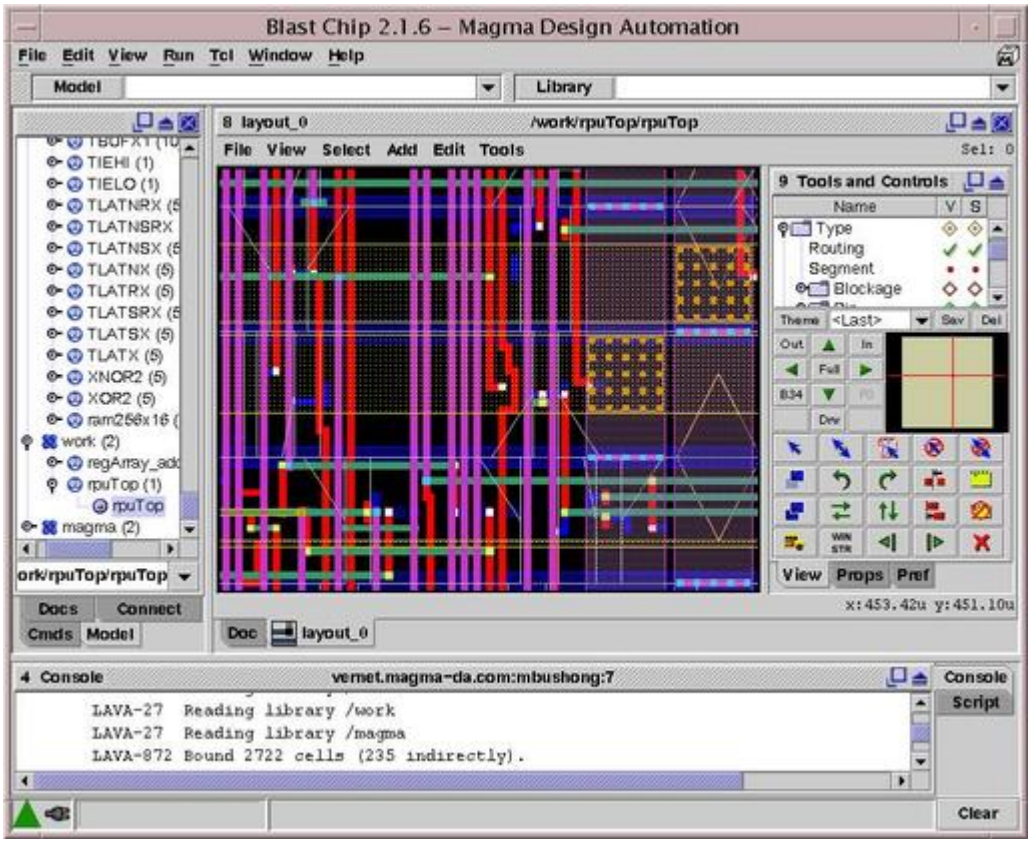
All HP workstations are certified by Linuxcare for Turbolinux, SuSE, Red Hat, Caldera, Mandrake and Debian. Linuxcare CEO Arthur Tyde explains, “We do a thorough QA on each distro and an in-depth analysis of kernel performance against system devices such as IDE.” Instructions for overcoming installation obstacles on a per-model basis for HP, Compaq, IBM and Dell are on-line at www.linuxcare.com/labs/certs.

While Linux is available standard on HP workstations everywhere, Linux desktops and laptops are a special order except in Asia and Europe. “Eastern Europe is more open to Linux as a desktop”, says Balma. “They don't have the installed base of Windows and are more sensitive to price.” Because of Winmodem issues, HP Pavilion home desktops and laptops don't fully support Linux, but that is being worked on.

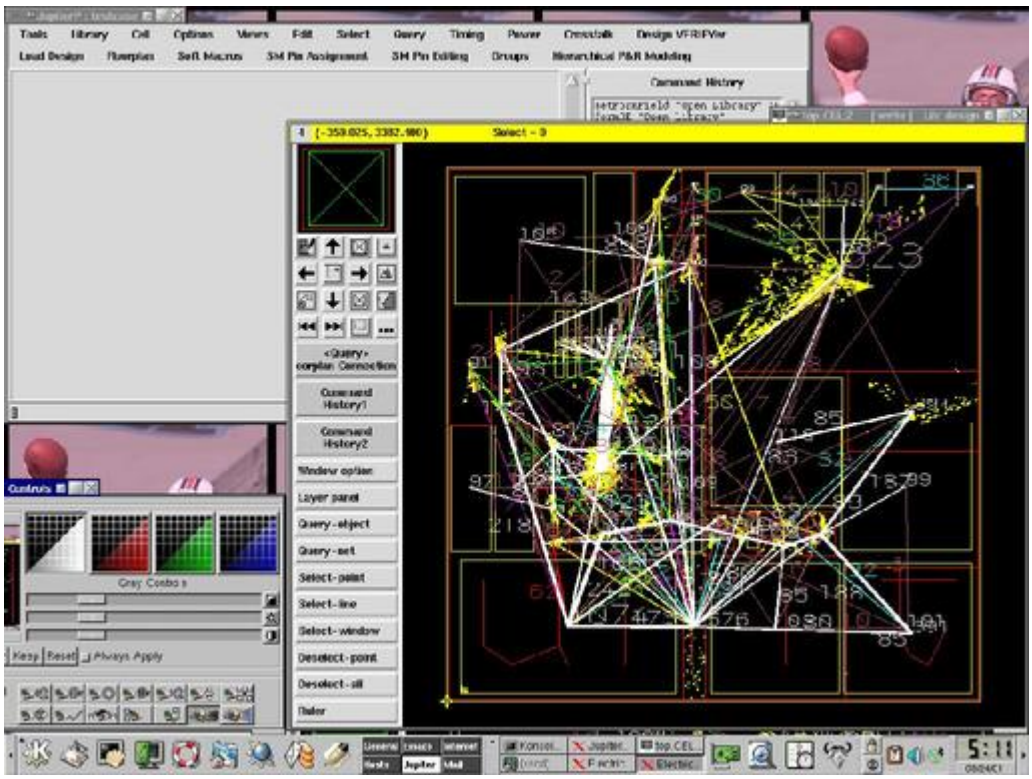
Former Debian Project Leader Bruce Perens joined HP as Linux strategist in December 2000. HP is making many of their drivers, particularly those for printers, open source. Open-source GNOME has been made the standard desktop for HP-UX. GNOME was independently adopted by HP and Sun at about the same time. HP denies they are offering their own Linux distro. “Some mistook our security-enhanced version of Red Hat for a distro, but it is really just a matter of bundling security with an existing OS”, says Balma.

Compaq Linux Program Manager Judy Chavis says, “I've been involved with Linux at Compaq for three years.” Compaq certifies with Red Hat, SuSE, Caldera and Turbolinux. “We don't see a lot of Linux desktop demand, but for workstations there is a lot of demand in EDA”, says Chavis. “Workstations are just starting to ramp up in the last six months, not much activity before that.”

EDA, oil and gas, and digital content creation are the basis of Compaq's Linux workstation market. Linux is certified and shipped pre-installed on workstations but is not offered on Compaq desktops and laptops.



Magma Blast Chip Screenshot



Avanti Jupiter View of a Floorplan Flyline Analysis

EDA chip design software maker Magma Design Automation made their Blast Chip and Blast Fusion products available for Linux in May 2001. “Over the past year we have seen an increasing demand to deliver our products for the Linux OS”, says VP of Marketing and Business Development Bob Smith. Avanti, another leading EDA vendor, is preparing to offer an end-to-end chip design system for Linux in Q4. “In response to customer input, many Avanti products have already been ported to Linux over the past couple of years”, says Chief Operating Officer Dr. Paul Lo.

Dell Computer, currently number four in the market with $33 billion in revenue, will become number three after the HP/Compaq merger. With 28% of the world market going to the Dell Precision line, Dell is the number one workstation provider. Before adding Linux in late 1999, Dell workstations were Windows-only, although Novell NetWare and SCO UnixWare were available on servers. “We are seeing significant interest from the motion picture market”, says spokesman David Graves. Dell recently dropped US support for Linux consumer PCs, an offering that was on their web site but never widely promoted. “We're a customer-demand-driven company”, says Graves. “For corporate customers wanting custom factory integration of 50 or more units, we'll still burn their Linux disk image on OptiPlex desktops and Latitude laptops.” Dell has also stopped offering Linux to consumers in Australia.

Gateway, a PC maker with nine billion dollars in revenue, announced in August 2001 it is pulling back from global markets outside the US and may lay off 25% of its workforce. In an effort to improve customer service, Gateway announced in February 2001 that it was reducing the 23 million potential combinations of computers it sells to hundreds of configurations. “We don't offer Linux as a standard configuration”, says spokeswoman Lisa Emard, “but do provide it to larger customers as needed through our custom integration group.” Gateway cites insufficient demand to support Linux at retail.

SGI was the uncontested market leader for servers and workstations combining high performance with advanced graphics. SGI, with $1.5 billion in revenue, is now down to 11% of the market for such systems, mainly due to a market shift to Windows workstations. Linux is part of the SGI strategy to recover from an unrewarding diversification into Windows workstations and Cray supercomputers. Although SGI bought and later sold Cray, supercomputing remains a core business with the release of the Origin3000 in 2000, Origin2000 in 1996 and Challenge in 1993.

“Linux is very exciting to us”, says SGI Advanced Graphics Product Line Manager Simon Hayhurst. “Our goal is to get O2 and Octane capabilities onto lower-priced Intel machines. The PC has a lot of legacy issues. We're making high-performance IRIX components like XFS, OpenGL and Open Inventor available as

open source in Linux." As standard definition television is becoming viable on PCs, SGI is looking to HDTV as an emerging market where extra performance is needed. "We capture some piece of the high end from business generated by Dell", says Hayhurst. "Low end always drives the high end." SGI doesn't offer desktops or laptops.

MicroTron2000 offers the lowest-priced desktops we could find looking in the local (San Diego, California) computer rag, *ComputorEdge*. While bottom-of-the-line PC prices at the majors ranged from a low of $659 US at Dell to $799 US at Gateway, a complete MicroTron2000 PC starts at $418 US for an AMD Duron 750 ($239 US including 256MB P133, 10GB HD, CD-ROM, modem, 100BASE-TX) and 17" ViewSonic E70 monitor ($179 US). One thing not included in that price is any Microsoft software. That's $95 US extra for Win98, $139 US for Win2K and no Microsoft Office at all. Manager Charles Tran says, "We would like to include Linux as a free operating system but haven't been able to find a Linux expert to join our staff. We are looking to support Red Hat."

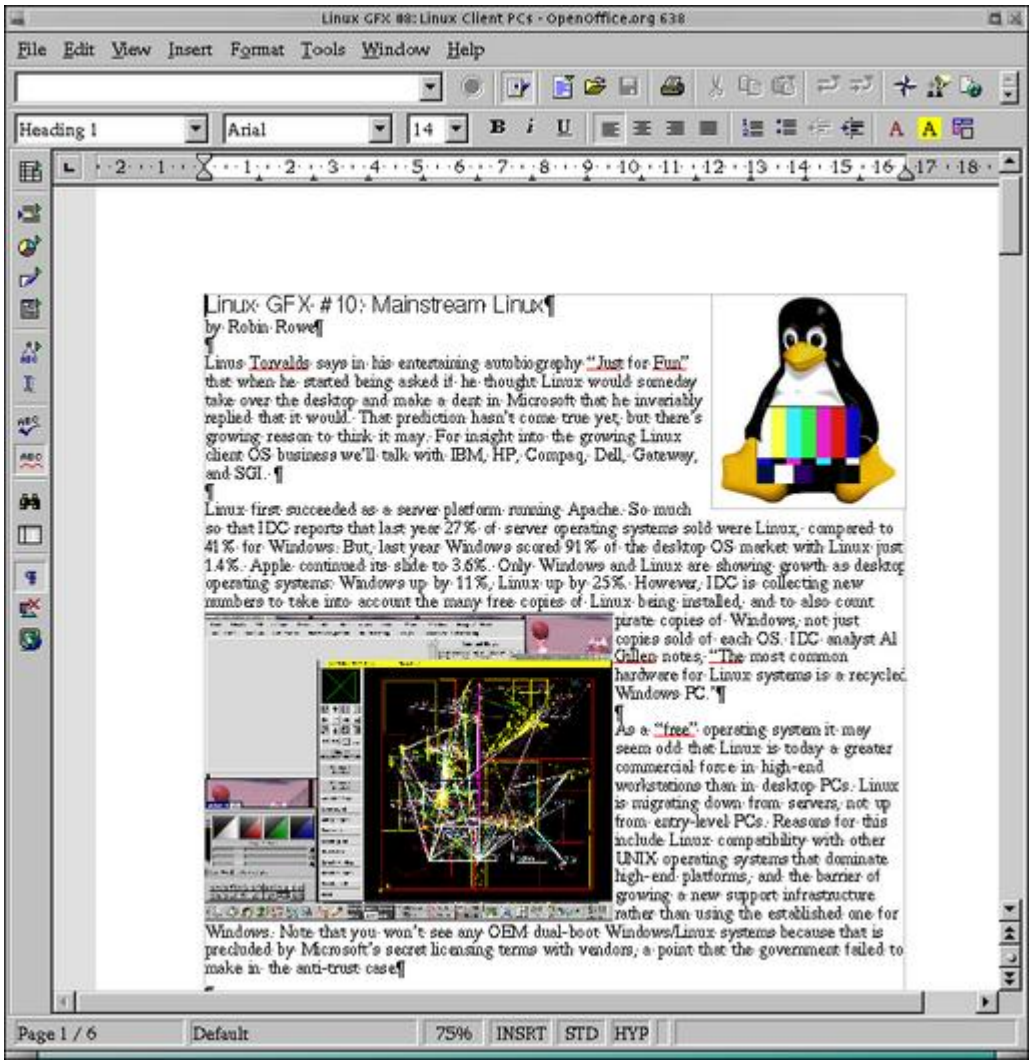
As PC prices continue to drop, Microsoft software becomes the single most expensive component. There's an unmet opportunity in the low-end market to create a useful PC without Windows.

The World Bank estimates that of the planet's six billion people, more than 1.2 billion live on less than one dollar a day (mostly in South Asia and Sub-Saharan Africa). Two billion more people are only marginally better off. Proportionately, few people in the world have a PC. The US has 285 million people but constitutes 102 million of the 236 million people active on the Internet. Total PC shipments worldwide this year are expected to be 130 million units. With an annual world population growth of 75 million and a PC obsolescent cycle of just a few years, progress is very slow toward everyone having a PC. Windows is only taking the low-hanging fruit. A 37% price break, saving $241 US per PC, does make a difference in most of the world.

Mexico City is undergoing a two-year transition to open-source software, already in use at the department of motor vehicles. GNOME evangelist Miguel de Icaza, a former system administrator at a Mexico City university, made a personal appeal to President Fox to consider open source for Mexico's broad eMexico computer initiative. Government initiatives to employ open source are underway in Brazil, France, Germany, South Korea and China.

For the desktop, Microsoft Office has become the dam holding back a Linux flood. A significant alternative to Microsoft Office is StarOffice. Sun Microsystems spent $74 million acquiring StarOffice as a product to give away for free, then further confounded business analysts by announcing last year that it would make version 6.0 GPL, open sourcing nine million lines of code.

Due for release in October 2001, OpenOffice 6.0 should be available by the time you read this. Anticipated enhancements include making the annoying integrated desktop feature optional, improved compatibility with Microsoft Office file formats and support for Chinese, Japanese and Korean.



Screenshot of Beta OpenOffice 6.0

In June 2001, Sun announced a US military commitment to StarOffice by DISA (Defense Information Systems Agency), the agency responsible for IT systems for the Pentagon and 600 other military organizations worldwide. StarOffice isn't displacing Microsoft, rather 10,000 UNIX copies of desktop suite Applix. Microsoft wasn't a contender for lack of Solaris support. StarOffice runs on Linux, Windows and Solaris. DISA anticipates deploying as many as 25,000 copies. Being free, StarOffice didn't net any revenue to Sun directly on the deal, but DISA has extensive support contracts with Sun.

An issue with government users is that Linux hasn't been certified as secure for military/government use (nor for that matter is Windows XP—only Windows NT 3.51). However, the NSA (National Security Agency) has developed a version of

Linux called SELinux to push the state of the art in OS security. NSA, DARPA and other US agencies are funding open-source projects for millions of dollars.

With the release of Windows XP, Microsoft says it has finally put the Windows DOS legacy to rest. XP is based on 2K/NT, whose design is closest to VMS. Windows 2000 was the first reliable Windows OS but shares the annoying trait with Windows NT of pausing for seconds from time to time to do system housekeeping. XP introduces some new annoyances (or features), including dropping support of Netscape plugins, not including Java by default (a consequence of losing a lawsuit with Sun), special antipiracy measures (requiring registration locking to a particular machine signature), the .NET portal (or gatekeeper) and the usual upgrade thrash of incompatibilities with existing third-party applications.

Linux is established in the server market. It is storming the workstation market. It has a beachhead in the government market. The next barrier Linux must pass is the corporate desktop. No *Fortune 500* company today has a significant Linux desktop deployment. But the increasing cost of Microsoft XP and broadening support for Linux are making many companies pause to consider. Ford Europe, with 33,000 desktops, has stated it is looking hard at Linux.

Most Windows desktop users merely want a Windows 98 that doesn't crash, that is not a moving target. In about one year Linux has come from nowhere to be the OS of choice for the motion picture industry. All the studios are converting to Linux, starting with DreamWorks, which has already released the first hit movie created primarily using Linux, *Shrek*. Now Linux is surging into other workstation markets like EDA. Advances in niche markets are strengthening Linux, improving its device driver and graphics support. Each improvement is removing an obstacle to the desktop. With broad industry support, Linux is unstoppable.

“Software is like sex: it's better when it's free”, according to Linus.

Resources

email: Robin.Rowe@MovieEditor.com

Robin Rowe (robin.rowe@movieeditor.com) is a partner in MovieEditor.com, a technology company that creates internet and broadcast video applications. He has written for *Dr. Dobb's Journal*, the *C++ Report*, the *C/C++ Users Journal* and *Data Based Advisor*.

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

No Longer Easy for Sys Admins

David A. Bandel

Issue #92, December 2001

With Linux moving to the desktop, life gets more complicated for system administrators.

Until now, system administrators running Linux have had it fairly easy. But this won't last long. Why? Well, until now Linux has been basically a server OS. The GUI, and therefore window manager, was just a pretty option. But as Linux moves to the desktop, the Linux system administrator's life becomes much more complicated. Now, it's not just a matter of knowing how to configure sendmail, DNS and Apache, but being able to fix GUIs users have managed to break somehow. That means knowing, inside and out, KDE, GNOME, XFCE, TWM, FVWM, MWM, Blackbox, etc. Sure, your present company may dictate KDE and enforce the choice. But another company will push GNOME or any of a dozen other WMs. Then there's the rogue “power user” who knows enough to be dangerous and has installed a private copy of his or her favorite WM. I've looked at the KDE configuration files and I'm awed. My /opt/kde/share/config has more files than /etc. Guess I'll be figuring them out for a while. Then it's on to GNOME. Yes, I know there are GUI config tools for all this. And that's all well and fine when you're standing in front of the system. But when the broken system is on the other end of a 56k dial-up connection, vi is still my tool of choice.

vcheck: www.tu-ilmenau.de/~gomar/stuff/vcheck

If you must maintain a system with the latest and greatest that's out, then vcheck can be a lifesaver. I don't know of any distribution that will keep libraries and other files up to date except for security reasons. I find that my libraries must be the most current for new software to build. This utility makes it easy. The default configuration will check on your kernel, but adding entries is easy. A quick cron job and you always have the latest software ready for installation. Requires: Perl and Perl modules: LWP::UserAgent, Getopt::Long, File::Copy, HTTP::Request and File::Basename.

GtkDiskFree: gtkdiskfree.sourceforge.net

While I generally prefer the command line to check disk usage, this particular utility is very nice. What I like about it is that I can put a copy on all my systems, then use ssh's ability to run an application remotely and display it locally (if you have X11 forwarding configured). I can put half a dozen on the screen and keep an eye on all the systems at once. Requires: libgtk, libfdk, libgmodule, libdl, libglib, libXext, libX11, libm, glibc.

ettercap: ettercap.sourceforge.net

ettercap goes beyond the simple (though extremely useful) tcpdump and shows you more about what's happening on your LAN. It can show all traffic between two hosts or can show only web traffic, etc. It also can analyze encrypted traffic and works even on a switched LAN. It can, however, be a little hard on a LAN on startup as it uses an ARP storm to gain LAN information in a switched environment. Of course, there are always command-line options that can be used to override this behavior. Requires: libdl, libform, libncurses, libm, libssl, libcrypto, glibc.

Perltidy: perltidy.sourceforge.net

I may be wrong (and often am), but I believe most administrators have at least a little knowledge of Perl. It's just that Perl is so darn useful and simple it's relatively easy to write in. And combined with a lot of powerful modules, so much can be done by those of us with little real programming knowledge. The Perltidy application can help you by making your Perl code a little easier to read. I personally don't always indent. And between writing sessions (write/test/rewrite) I find I occasionally do things differently. Perltidy attempts to clean up all that and more. It's not perfect, but it does make your code more readable. Requires: Perl and the Perl module IO::File.

gocr: jocr.sourceforge.net

The gocr program is an attempt to bring optical character recognition (OCR) capabilities to Linux. This program is designed to work with PNG files (faxes) to turn them into ASCII text. I tried a number of different documents and found that, while gocr does work, it is fairly sensitive to the font size and type used. If someone sends you a very fancy fax with italics and fonts in differing sizes, gocr probably won't help much. But a very plain fax using a courier font comes through fairly well. As usual, your mileage may vary. Requires: libpnm, libpgm, libppm, glibc.

ipband: ipband.sourceforge.net

This utility will show you bandwidth utilization on your network. Yes, you can use MRTG, but if you don't want to run SNMP and a web server, this will do you well. Most administrators need something to help justify increases in bandwidth or capacity for your LAN, or just to identify the bandwidth hogs. Requires: libpcap, glibc.

GtkLP: www.stud.uni-hannover.de/~sirtobi/gtklp

If you're running CUPS as your printer software, you might want to look at GtkLP for use in the X Window System. While CUPS comes with a web interface, it's not the quickest way around the CUPS system. GtkLP will get you around much more quickly. The utility is easy to use and allows you to do a number of things much more easily than with CUPS' web interface. Worth a look. Requires: libcups, libgtk, libgdk, libgmodule, libglib, libdl, libXext, libX11, libm, libssl, libcrypto, glibc.

Until next month.

email: david@pananix.com

David A. Bandel (dbandel@pananix.com) is a Linux/UNIX consultant currently living in the Republic of Panama. He is coauthor of Que Special Edition: Using Caldera OpenLinux.

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

A Walk on the Embedded Side of LinuxWorld

Rick Lehrbaum

Issue #92, December 2001

Rick gives his impressions of all things embedded andembeddable at August's mega-Linux event.

Despite the somewhat subdued mood of this year's big West Coast LinuxWorld, there was near-universal agreement that one thing seemed to have grown stronger over the past six to twelve months: embedded Linux. Although the event does not particularly focus on the embedded market, a host of new products, technologies and strategies catering to the needs of embedded systems and smart device developers and manufacturers was unveiled and showcased. Everywhere you turned, you ran across companies embedding Linux inside PDAs, entertainment devices (especially TV set-top boxes), automotive telematics systems, thin clients, etc.

In the past, embedded Linux products and technologies accounted for roughly 10% of what was showcased at LinuxWorld. At this show, the embedded Linux fraction seems to have increased to around 15-20%, which is not surprising, given the strong growth in developer interest in embedded Linux reported in recent months by market analysts VDC, Evans Data Corporation, *Embedded Systems Programming* magazine, LinuxDevices.com and others.

For those unable to make it to the show, or whose attention was elsewhere, here's my traditional walk on the embedded side of LinuxWorld in which I offer a summary of all things embedded and embeddable (in alphabetic order). My apologies in advance, to anyone or anything I missed.

Applied Data Systems

Visitors to ADS' booth were treated to a dazzling array of LCD-adorned StrongARM-based, single-board computers—truly a hardware geek's paradise. ADS has created their own embedded Linux implementations that are offered as an option with all of their SBCs, but in addition ADS was showcasing three

Java-enabled, graphically oriented application environments: Insignia's Jeode VM running on Lineo's Embedix, IBM's VisualAge Micro Edition running on MontaVista's Hard Hat Linux and Blackdown's JDK running on ADS' own flavor of embedded Linux. ADS also announced this week that they have been selected by Intel as a third-party platform provider for Intel's StrongARM and XScale processors and will be developing a modified version of their Graphics Master SBC for use as a reference platform by Intel customers. (See www.applieddata.net.)

Advantage Business Computer Systems

Tucked away at the Linux Terminal Server Project (LTSP) booth, David Anders was showing off his company's tiny Linux system—definitely a cool little Linux box. Anders said a new version, which is coming out soon, adds a CompactFlash slot that facilitates system expansion and the use of removable media. Nice touch. (See www.abcsinc.com.)

A Tiny Linux System from Advantage Business Computer Systems

Century Embedded Technologies

Century occupied their usual position at a pedestal within Red Hat's large pavilion. Century's big announcement at the show was PIXIL, a respin and enhancement of its previous key products (Microwindows and ViewML) into a new suite of graphical environments, tools and applications targeting Linux-based PDAs, webpads and thin-client devices. Demonstrations included PIXIL running on a Compaq iPAQ PDA, as well as a PIXIL-based set-top box based on National Semiconductor's SP1SC10 set-top box development platform. Greg Haerr (Century's CEO) made the claim that, whereas the TV Linux Alliance Project is just at the point of beginning to define standards, Century/National's newly announced Linux4.TV is available for immediate use by customers and already provides support for both digital and analog TV, personal video recorder (PVR) functions and APIs for both kernel and middleware software interfacing. (See embedded.censoft.com.)

CodeWeavers

CodeWeavers demonstrated the recently announced CrossOver technology that allows Linux systems to use Windows browsers and application helpers. According to Jeremy White (president), an embedded version of CrossOver is being developed that will occupy on the order of 1MB of RAM and 4.5MB of storage. Unfortunately, the Windows plugins are sometimes quite greedy in terms of the system resources they require in order to run under CrossOver. CodeWeavers, a driving force behind WINE, can provide services for getting embedded applications to run on Linux under WINE or can port the apps directly to Linux APIs. (See www.codeweavers.com.)

Earthlink's R&D Group

This group showcased the prototype of an open-standards Automotive Vehicle Location (AVL) telematics platform that was created to serve as a platform to demonstrate automotive software applications, study how Linux and internet technologies can be applied to the automotive environment and test the feasibility and effectiveness of systems such as remote diagnostics, m-commerce and location-sensitive applications. Earthlink is sponsoring a contest for Linux, XML, Java, wireless and web application developers. Come up with a clever idea and get one of these interesting systems to play with. (See www.research.earthlink.net.)

Earthlink R&D's Telematics Research Platform Runs Linux

Embedded Linux Consortium

The ELC announced that several new members have joined recently, including Future Sound Technologies, Intel Corporation, American Megatrends, Inc., Aleph One, Ltd. and Vibren Technologies. Based on the roster at the ELC web site, the consortium currently has 68 corporate members (36 executive and 32 affiliate level). One bit of hot news is that the ELC's Board of Directors recently approved a change to the ELC's bylaws to allow the ELC to own intellectual property, thereby clearing the way for ELC to engage in developing and licensing embedded Linux standards. (See www.embedded-linux.org.)

Embedded Linux Journal

On the opening day of LinuxWorld, *ELJ* announced the winners of their first embedded Linux design contest. Each winner gets an all expenses-paid trip to Costa Rica! (See embedded.linuxjournal.com.)

Empower Technologies

Empower demonstrated their new LinuxDA "Linux upgrade" for the Palm III and V. They also announced that LinuxDA has been licensed by two Palm-compatible Taiwanese consumer device manufacturers: Elitegoup Computer Systems and APlux Communications. (See linuxda.com.)

Hewlett-Packard Company

Hewlett-Packard issued a major embedded Linux strategy announcement in connection with LinuxWorld. Per the announcement, embedded Linux is now the OS of choice for all HP-developed devices. HP's embedded Linux system is called Chai-LX. HP also demonstrated the new Digital Entertainment Center audio system, which contains an x86 processor and uses Chai-LX as its embedded OS. (See www.hp.com/linux.)

Linux in the Living Room? HP's New Digital Entertainment Center

IBM

IBM alphaWorks showed off its experimental TechMobile, a modified Ford Explorer 2002 Limited Edition that has been outfitted with Linux-powered computers running a variety of web-enabled applications. In the demo, a simple user interface on a Bluetooth-enabled PDA controls the car's headlights, door locks and engine starter. Other IBM technologies showcased in various locations on the show floor with applicability to embedded Linux-based systems included the DB2 database, VisualAge Micro Edition (shown by MontaVista) and Embedded ViaVoice (also at MontaVista's booth). (See www.alphaworks.ibm.com.)

IBM's Car of the Future Has “Linux Inside”

Lineo

In contrast with last summer, when Lineo occupied multiple locations on the show floor due to having acquired several other Embedded Linux companies, Lineo was not an exhibitor this time, but instead held a press conference to make several strategic announcements. These included: the selection of Lineo's Embedix Digital Media core by Motorola for use in the DCT5000 set-top box and the planned release of a “GPL Compliance Toolset” to help developers and their companies identify which licenses govern their software and what the implications of those licenses are. (See www.lineo.com.)

Lisa Systems

If you ask Compaq when (or where) you can get an iPAQ with Linux pre-installed, they'll probably refer you to Lisa Systems. Lisa demonstrated their iPAQ/Linux combo, which makes use of Trolltech's Qt/Embedded as its GUI framework, at a pedestal in Compaq's large LinuxWorld pavilion. (See www.lisa.de.)

LynuxWorks

Like Lineo, LynuxWorks decided not to have their own booth at LinuxWorld. They did, however, have a small presence at a pedestal within Intel's pavilion, where they demonstrated their port of BlueCat Linux to Intel's Internet Exchange Architecture (IXA). (See www.lynuxworks.com.)

MontaVista

Like a show within a show, MontaVista's large pavilion-style booth was bristling with product and technology demonstrations. Beneath a giant hard hat-adorned Tux suspended from the convention hall ceiling, were:

- A demo of Hard Hat Linux (HHL) 2.0's cross-development tools targeting an Alchemy AU1000 system-on-chip based set-top box reference platform. The use of the new KDevelop IDE was also being shown, as were MontaVista's Target Configuration Tool and newly GPLed Library Optimizer Tool.

- A demo that dramatized the improvement in interrupt latency that can be obtained using MontaVista's “fully preemptable kernel” enhancement, in which streaming audio was being played in a system that was under a heavy load (repeated disk saves). Interrupt latencies were captured and graphed for visual comparison.
- A demo of MontaVista's newly announced High Availability Framework, in which a CompactPCI-based Linux system was shown to maintain a video display reliably from a network-based streaming video source while one or more of three Ethernet cables were disconnected.
- A set-top box demo based on HHL and the recently announced Hard Hat Graphics running on IBM's “Redwood” PowerPC 405-based reference platform.
- HHL running on an iPAQ PDA, using two approaches—Hard Hat Graphics and Trolltech's Qt/Embedded—including demonstrations of both tools' GUI-builder development methods.
- Demos of two IBM products: VisualAge Micro Edition, a Java-like VM; and ViaVoice, speech recognition software that supports both desktop and embedded systems. ViaVoice-based speaker-independent command and control supposedly can fit within 200KB of RAM.
- A delightful demonstration of embedded Linux-based machine control, in which Intrinsyc's tiny StrongARM-based CerfBoard SBC running HHL was shown controlling the walking of a Lego MindStorm Robot that looked like something out of a *Star Wars* movie. (See www.mvista.com.)

Look What a Tiny StrongARM SBC Plus Embedded Linux Can Do

PalmPalm

PalmPalm showed off their Tynux Box StrongARM-based PDA reference platform and Tynux Linux OS. They also had a Korean-manufactured cell phone/PDA on display. (See www.palmpalm.com.)

Red Hat

Joe deBlaquiere (senior engineer) demonstrated a prerelease version of a new Red Hat Embedded Linux Developer Kit. The tool is meant to make it easy for developers to embed Red Hat Linux. According to deBlaquiere, the tool makes

use of standard Red Hat SRPMS, so it “leverages the stability of Red Hat Linux transitioned to the embedded space”. A number of predefined baseline configurations are available, including minimum bootable system, minimum networked system, etc., to which developers can add whatever they require by means of an easy-to-use, GUI-based configurator. One unique aspect to Red Hat's target OS builder approach, according to deBlaquiere, is that unlike Lineo's LIPO and MontaVista's LOT, Red Hat's library reduction/optimization process makes use of the four defined EL/IX profiles, with the result that embedded systems will be characterizable as being compliant with a specific well-defined API set. Additionally, Red Hat's RedBoot debugger/boot loader has been integrated into the new Embedded Linux kit as a standard component. deBlaquiere said the first beta release of the new kit is expected within a few weeks. Speaking of which, both RedBoot and Red Hat's other OS, eCOS, were also being demonstrated in Red Hat's pavilion. (See www.redhat.com/embedded.)

REDSonic

REDSonic demonstrated four new products, including: Secure SOHO, a gateway/firewall software solution; Windows-Based Terminal, software that provides thin-client, Linux-based access to Windows applications and services; LinuxBIOS, a ROM-based Linux kernel that replaces legacy BIOS code for secure, fast boot; and a REDSonic PowerPC starter kit for the MPC 823. REDSonic also showcased an interesting customer application: SignSite, manufactured by Clarity Visual Systems. It is a bright, full-color “digital billboard” that is used for advertising and information display in public places. The device contains an embedded Ampro Encore 500 Pentium-based SBC running REDSonic's REDICE-Linux OS. (See www.redsonic.com.)

RidgeRun

RidgeRun demonstrated DSPLinux running on TI's digital still camera reference platform based on the TI TMS320DSC21 dual-core (RISC + DSP) system-on-chip processor. That chip, according to Rudy Prince (RidgeRun CEO), is used in several high-end digital cameras from HP and Kodak. Also demonstrated was the DSPLinux development environment's very slick “appliance simulator”. Can we expect to find DSPLinux inside digital cameras sometime soon? “Yes”, replied Prince. Embedded Linux will play a growing role in devices that feature Ethernet, 802.11 or Bluetooth connectivity and other high-end functions, according to Prince. (See www.ridgerun.com.)

The SH/Linux Project

While passing by the Japan Linux Association's booth in the .org area, I noticed some demos of Linux running on two Japan-manufactured game machines:

Sony's PlayStation 2 and Sega's Dreamcast—the former, based on Sony's recently released PS2 SDK, and the latter, based on efforts of the Linux-SH Project.

Tuxia

Tuxia demonstrated their TASTE Embedded Linux running on a new set-top box reference platform and also on some very nice looking thin-client systems. They also gave the first LinuxWorld demonstration of their TASTE OS running on an iPAQ PDA. The iPAQ demo included TUXIA's Nanozilla small footprint browser with 802.11 wireless web connectivity. The company issued two key news announcements in conjunction with the show: an agreement with system-on-chip maker Rise to support Rise's SOCs with TASTE Embedded Linux and the selection of TASTE as the Embedded Linux OS for a new consumer media gateway reference platform being developed by Intel. (See www.tuxia.com.)

Transvirtual Technologies

TVT was showing TV set-top box and PDA applications for its new Java/XML/ Linux based XOE architecture. (See www.transvirtual.com.)

Trolltech

Trolltech demonstrated a prerelease of version 2 of the Qt Palmtop environment, running on the ubiquitous iPAQ PDA. According to Aron Kozak (Trolltech's PR manager), the environment also works on Sharp's Zaurus PDA. New functions being demonstrated included synchronization, a multiplatform desktop PIM suite, improved add/remove software features, an enhanced GUI design and several new applications including an e-mail client. According to Trolltech's announcement, the full release of version 2 will have features designed to support not just PDAs, but also next-generation “smart phones”. A downloadable beta release is expected in October 2001. Incidentally, Trolltech won LinuxWorld's coveted “Best Embedded Linux Solutions” award for Qt/ Embedded at the show. (See www.trolltech.com.)

$99 Linux PDA?

Empower Technologies has announced a $99 PDA that the company claims is a Palm IIIxe clone from the hardware perspective, but runs Empower's Linux DA operating system instead of Palm OS. Here's a summary of specs, according to information on the company's web site:

- Palm IIIxe-compatible
- Supplied with Linux DA O/S for PowerPlay III
- Screen: 240 x 320 pixels, monochrome touchscreen

- Processor: Motorola Dragonball, 16MHz clock rate
- 8MB RAM
- 2MB Flash (upgradeable)
- Serial port for connection to PC
- Expansion slot(s): none
- Battery: nonrechargeable AAA cells
- Color available: Galaxy Grey
- 15 days technical support by e-mail
- Six-month Linux DA O/S software upgrade
- One-year warranty
- Includes: carrying case, two AAA batteries, PC Sync Cable, CD with PC Sync
- Program, Quick-Start Manual, Handwriting Guide, Warranty Card, Registration Card

  (See www.empowertechnologies.com.)

> **Rick Lehrbaum** (rick@linuxdevices.com) created the LinuxDevices.com “embedded Linux portal”. Rick has worked in the field of embedded systems since 1979. He cofounded Ampro Computers, founded the PC/104 Consortium and was instrumental in creating and launching the Embedded Linux Consortium.

email: rick@linuxdevices.com

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

License FUD

Lawrence Rosen

Issue #92, December 2001

The threat of Microsoft's shared-source licenses.

A few proprietary software vendors argue that there is something almost un-American about free software and the GPL. Microsoft has been one of the loudest voices in that chorus of fear. That company's web site describes it this way:

> The GNU General Public License (GPL)...poses a significant threat to the IP base of companies seeking to build a business around GPL-covered software. Even businesses who may believe they are “mere users” of GPL software are threatened since they combine what they believe to be separate applications with GPL code. This licensing model has the effect of foreclosing a business' choice of what IP to share with the community and on what terms (www.microsoft.com/business/licensing/ssfaq.asp).

I've already argued elsewhere (see www.rosenlaw.com/html/GPL.PDF) that this supposed GPL threat is hogwash. But, there's a deeper irony to this: Microsoft's own shared-source license is an even greater threat to the software development community. That license is a Trojan horse license that can destroy your open-source and proprietary software unless you are very careful.

The simplest version of Microsoft's shared-source licenses is the one they use to distribute their Windows CE 3.0 source code. That license states, in its second paragraph, “You can use this software for any noncommercial purpose, including distributing derivatives.” The license then makes it clear that running your business operations “would not be considered noncommercial”.

Commercial users—and I consider most open-source software developers to be in that category—must look further in the license to determine the

restrictions on use that apply to them. The third paragraph of the license conveys the bad news:

> For commercial purposes, you can reference this software solely to assist in developing and testing your own software and hardware for the Windows CE platform. You may not distribute this software in source or object form for commercial purposes under any circumstances.

Note that Microsoft has not given you permission to copy their code or to incorporate it into your own software in a derivative work. Your use of their software for commercial purposes is limited to reference purposes only.

Obviously, you can agree not to make a copy of any portion of the Microsoft software or to use it to create derivative works. But what happens if later you, independently and without consciously remembering what you saw in Microsoft's code, create something substantially the same as their code? Can you still be liable for infringement?

That's where the Trojan horse comes into play. The courts have made it clear that, under copyright law, proof of substantial similarity between your work and another work, along with proof of access to the other work, may be enough to prove infringement, even when you don't realize that you're making a copy.

How easy is it to forget something important that you read? A copyright infringement case from the 1970s will illustrate the problem. In 1976, George Harrison's music company was sued for copyright infringement. A music publisher claimed that Harrison plagiarized his successful song, “My Sweet Lord”, from an earlier successful song, “He's So Fine”. In order to prove copyright infringement, the publisher of “He's So Fine” had to prove not only that there was “striking similarity” between the two songs, but that Harrison had copied the original song when composing his. Harrison admitted that he was familiar with the original song, but that while he was working on “My Sweet Lord” he wasn't conscious of the fact that he was using the “He's So Fine” melody. The court concluded:

> In seeking musical materials to clothe his thoughts...there came to the surface of [Harrison's] mind a particular combination that pleased him....Did Harrison deliberately use the music of “He's So Fine”? I do not believe he did so deliberately. Nevertheless, it is clear that “My Sweet Lord” is the very same song as “He's So Fine” with different words, and Harrison had access to “He's So Fine”. This is, under the law, infringement of copyright, and is no less so even though subconsciously accomplished.

> —Bright Tunes Music Corp. v. Harrisongs Music, Ltd., 420 F.Supp. 177 (S.D.N.Y. 1976).

Subsequently, Harrison's music company was found liable for some $1.6 million in damages.

Anyone familiar with the art of computer programming will recognize that, as with music, there are rather standard ways to express certain thoughts. Having once seen Microsoft's code, will an expert programmer be able to erase that example from his mind? Even if he or she consciously attempts to forget and does not intend to copy, will his or her subconscious memories be expressed in his later code with sufficient similarity that a court will find copyright infringement has occurred?

I encourage open-source programmers to avoid that risk. Don't look at source code licensed under Microsoft's shared-source licenses unless you're one of the rare breed of humans who can control his or her subconscious.

Legal advice must be provided in the course of an attorney-client relationship specifically with reference to all the facts of a particular situation and the law of your jurisdiction. Even though an attorney wrote this article, the information in this article must not be relied upon as a substitute for obtaining specific legal advice from a licensed attorney.

email: lrosen@rosenlaw.com

Lawrence Rosen is an attorney in private practice in Redwood City, California (www.rosenlaw.com). He is also executive director and general counsel for Open Source Initiative, which manages and promotes the Open Source Definition (www.opensource.org).

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

The Triumph of Stuff that Matters

Doc Searls

Issue #92, December 2001

Doc contemplates the events of September 11th and the betternature of the species.

Thirty years ago the World Trade Center located at the foot of Manhattan was nearing completion. One of its boxy towers, five years in the making, had recently become the tallest building in the world. The other would tie that record when it was finished two years later.

I was 19 and a freshman in college when they broke ground on the World Trade Center in August 1966. I was 25 and launching my third career—this one in radio—when it was finished in April 1973. I was also married and a father. That whole time I took a keen interest in the project.

I grew up in New Jersey, across the river from Manhattan. My father grew up there too, in a house atop the Palisades, in Fort Lee. As a young man he was a high-steel construction worker, helping build the George Washington Bridge. We both followed the progress of the World Trade Center and talked about it often, since we shared the same erector-set mentality and love of heights.

For a while I also shared a yard with a British chap named Roy who worked as a foreman on the World Trade Center project. We'd often sit outside drinking Bass Ale while Roy clued me in to the building's highly unusual construction.

With most high-rise buildings, Roy explained, the exterior is cosmetic. It's just fascia. But with the World Trade Center, the exterior was structural. The floors inside were like shelves in a metal box, each held up by tabs and bolts.

I thought about that as I watched the North Tower burn like a giant chimney on live television. By then all we knew was that an American Airlines 767 had flown squarely into the building, filling the upper floors with thousands of tons of wreckage and burning jet fuel.

Like millions of others watching the same coverage, I was filled with dread. Somewhere in there is somebody I know. I also wondered if the structure would hold. The surviving concrete could take the heat, but what about the steel frame? Then came news that another jet had plowed into the Pentagon, and fear became terror. Moments later a third jet flew into the South Tower, disappearing intact, like a knife slicing into a cake. A second after it vanished an orange fireball blasted holes out two other sides of the building. Horrified, I asked myself an awful question: how long before the shell melts and the floors start dropping?

The answer came soon enough. About 15 minutes after absorbing a United 767, the South Tower—the second one hit—sagged into a cloud of gray dust. A few minutes later the North Tower followed. In less than an hour, the world's most important skyscrapers were reduced to a mound of debris and a grave for thousands.

“Who did we know who might be in there?” I wondered. Then I remembered David Alper.

The last time I had seen Dave—whose fortuitous nickname is Save—was back in January at LinuxWorld Expo. (That's him in the picture, getting a slap on the back from yours truly.)

Dave is as New Yawk as they come. Big, generous and full of life, it was hard to imagine the world without him. But I knew he worked in the World Trade Center. Was he at work? Was his office on a high floor? Did he get out?

For days calls to Dave and his wife Susan reached only busy signals or voicemail boxes. But finally, we got a call. Sure enough, Save had earned his nickname, big time.

Turns out he was in his office on the 84th floor of the South Tower when the North Tower was hit. Nobody knew for sure what had happened, but clearly it wasn't good. Ignoring the public address voice assuring people that the South Tower was safe, Dave joined the crowds who started down the stairs. It took an

hour to reach the street. While he was still in the stairwell, the second plane tore through Dave's office and pretty much everything else on the floors running from 82 through 85. The building shook but stood. For another fifteen minutes it performed a final service to thousands pouring down the stairs from floors below the impact. Dave was two blocks away when the building fell. From his floor alone more than 70 people were killed.

Nearly a week would pass before my other New York friends, including several who worked in the shadow of the towers, wrote or called to announce their survival. The only casualty I knew personally was Daniel Lewin, the cofounder and CTO of Akamai, which uses Linux and Apache servers to make the Web more efficient for its customers (one of which is Microsoft). Danny was born in Denver but raised in Israel. He was a passenger on one of the planes that crashed into the WTC. If he had known the hijackers' true intentions, I am sure he would have been a hero.

Still, Akamai's technology carried much of the redistributed load when the loss of the World Trade Center echoed through the Internet. Across its native internet habitat, the Linux community has taken to heart the business half of the Slashdot slogan: Stuff that matters. Quietly, with no publicity, enormous rebuilding work is already being done.

As I write this, the event is still only a few days past. It's almost impossible to think, much less write, about anything else. Nothing matters more.

Airplanes have barely resumed flying, but I took one yesterday. It was nearly empty, so I spent the whole flight talking with the flight attendants. One of them described September 11 not as an “Attack on America” (the consensus name given by the major media), but as a “heart attack”. Everything stopped and the country shut down cold. How long, she wondered, could vital economic organs withstand the absence of business travel and freight transportation?

Right now the only thing that's clear is there's a new kind of war going on.

“Compared to war, all other forms of human endeavor shrink to insignificance”, George Patton famously said. No wonder, given the enormous loss of life and property, that our government immediately declared the attack an “act of war”--even though the enemy was an -ism rather than a country.

That enemy seeks to raze in minutes what civilization takes years to raise. Only a few months before terrorists destroyed the World Trade Center, the terror-supporting Taliban government in Afghanistan used enormous ancient Buddhist statues for target practice. How can they possibly succeed at anything

that doesn't involve suicide? Especially against the better nature of the species? Of which, both Linux and UNIX are excellent examples.

We've been at work on Linux for more than ten years. Before that we worked on UNIX for decades. This work has had a profoundly civilizing effect on all forms of human endeavor that rely on computing and communications. Without UNIX, the Internet would have been an impossible dream. Now it's an unstoppable reality. Linux makes it more unstoppable every day.

And the more we have to rebuild, the more apparent that will become.

Doc Searls is senior editor of *Linux Journal* and coauthor of *The Cluetrain Manifesto*.

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

The HP SureStore Ultrium 230 Tape Drive

Tanner Andrews

Issue #92, December 2001

It can write tape faster than your disk subsystem can supply data.

The HP Ultrium 230 low-voltage differential (LVD) tape drive is fast. It can write tape faster than your disk subsystem can supply data. If you want to stream, you'll need several systems feeding it at once, which made it a good choice for our application of backing up several systems each night. The 230 is about twice as fast as the 215 model, but also costs significantly more.

We bought one of the first HP Ultrium generation 1 tape drives, together with the Qualstar TLS-8211 tape library robot. Both are connected to an Adaptec PCI SCSI controller. It has been in service since December 2000, and we have done both file and system restores for the systems it protects.

The robot and tape drive are available in standalone and rackmount configurations. We ultimately chose the rackmount, and I advise others to do the same: something this big sitting on the floor will draw attention from the cleaning people. In the rack, the device occupies 36.75" (21U) of space, and it is a two-man job to get it installed. It is best to mount the cabinet with the robot, and then install the tape drive and magazine. The magazine has an 11-tape capacity. The tape drive slides easily into place once the cabinet is mounted.

Our selection of the device was driven by two considerations. First, the device had to be able to hold all of our data. The single-largest server potentially holds over 200GB of data, uncompressed, and the other servers roughly match that. The second requirement was that the data had to be written during the off-hours. We cannot have backups running during the day. We prefer full backups; restoration is much simpler in this case. Since periodic full backups must be made anyway, we do it consistently.

This drove us to a trial of the LVD tape subsystem, which promised to write 50GB/hour and hold 100GB (raw) on a tape. Our informal tests show that the

drive can really stream data at these speeds. Unfortunately, these same tests show that none of our systems can supply data at this rate! The tape drive can accept data far, far faster than a RAID-5 controller can supply them.

Our solution was to prepare a backup aggregation program. It collects backup streams from several computers and writes them all to tape together with headers identifying their origins. Several computers, combining their efforts, come closer to keeping the tape drive streaming. A companion program deaggregates the streams, providing the desired stream for restoration. Each system supplies a tar stream, and on restore the system receives the same.

Our installation combined the tape drive with a tape-loading robot. We use the mtx program to operate the robot, under the control of a fairly simple shell script. The shell script determines the next tape number, invokes the robot to load the tape, and starts the backup program. When the tape program is done, another script unloads the tape. For convenience, we throw a sheet on the printer showing the date and tape number used.

The robot and tape drive can be moody, and like small children will misbehave in order to receive attention. Loading and unloading the tape require care. If not done correctly, the subsystem will achieve a state in which one must manually extract the tape. Sometimes a power cycle is even required.

The tape drive tracks usage to decide when it needs cleaning. When the tape drive wants to be cleaned, it really wants to be cleaned: be sure you have a cleaning tape in the shop.

Preparing to use a tape consists of two stages: directing the robot to pick a tape and put it in the drive, and then encouraging the tape drive to do the actual load. The mtx command handles the former. In order to use the tape, however, you must then wait for it to become ready. We use a shell loop, waiting until the mt rewind command returns successfully when the drive has loaded the tape.

After the tape is written, we want to unload it and put it back in the magazine. Normally, when the tape-writing program finishes and closes the device, it will wait some time for the tape to rewind. The tape will be left in a loaded state, and in this drive the robot cannot remove it and return it to the magazine.

You must use **mt rewoffl** or something similar to ensure that the tape is rewound and unloaded. Our experience is that, due to the length of the tape, the first issuance of the command may return prematurely. To avoid problems, wait several seconds and re-issue the command. The rule “This time, for sure!” applies.

Only when the tape is really unloaded can the robot pluck it from the drive and return it to the magazine. We have found that occasionally, if the robot tries to pluck the tape prior to unloading, the tape drive will get grumpy. This can sometimes be cured by manually pressing the unload button and waiting; it is often necessary to cycle power and then press the unload button.

So as long as you don't do anything to upset its routine, the robot will take good care of your tapes. The drive is big and fast. If you have a regular need to back up a lot of data, this combination may be right for you.

The tape drive and library are available from Zzyzx. They were kind enough to provide us with an evaluation unit that we kept for about a month, in order to determine suitability. We found that we would require the optional bar code reader, and we preferred a rackmount model over the freestanding unit. Zzyzx took back the evaluation unit and supplied the rackmount unit, with a bar code reader. While we have not needed to do so, the unit can be painlessly upgraded with a second tape drive, and/or more tape cartridges.

Product Information/The Good/The Bad

email: trandrews@learningsoft.net

Tanner Andrews is a longtime civic activist in the DeLand/West Volusia area. He regularly points, laughs and jeers (www.payer.org) at bad government. When he's not fighting the good fight, Andrews works at Learning-Soft in Miami, Florida as a network guru and programmer. For nearly 20 years, Andrews has been counselling users on the importance of regular, reliable backups.

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

Letters

Various

Issue #92, December 2001

Readers sound off.

Letters

Formerly Too Informal

I was very impressed with the September 2001 issue of *LJ*. The new look is great. It even seems the writing is more professional in these last two issues than it had been in the past. I initially thought this issue was the best I'd read! But the October 2001 issue is right up there, if not better! I had considered dropping my subscription to *LJ* because the articles read too informally. I don't know if that makes sense. But these last two issues have been excellent! Keep up the good work.

—Matt C.

Shorter UPFRONT?

I must say that overall I like the new format of *Linux Journal*. I am disappointed, however, in the reduced content of the UPFRONT section. I really enjoyed all the quotes and odd statistics in the old format of the UPFRONT section. I hope that the lack of these items in the September 2001 issue is only a temporary condition.

—John Metzner

Not to worry, John—the amount of UPFRONT content varies somewhat month to month.

—Editor

Disallowed Dirty Phrase

I could not resist trying the small driver from the article “Loadable Kernel Module Exploits” (*LJ*, September 2001). But there were some problems: 1) there are too many headers in Listing 1, 2) char * temp_buf is unused, 3) wrapped_write() should stick whether to “n” or “count” as one of its parameters and 4) wrapped_write I would put as wrapped_write(int fd, void *buf, size_t n).

I liked the idea about replacing all “Linux sucks” phrases to “Linux rules”, so I tried to implement it. I extended the wrapped_write function to do such a replacement. The function strstr() gave me problems while compiling, and I did not find an implementation of it on my Mandrake 8.0 laptop. “extern strstr” was everywhere. So I used a simple loop with strncmp(), and it proved to be more useful because it can replace several occurrences of a phrase. To do so and please the compiler, I removed “const” from the wrapped_write() parameters to write back to “void *buf”. Also, I had to change the linux/version.h file because I had compiled the kernel myself, and insmod was swearing at the difference. But in the end when I had succeeded, I really enjoyed playing with such a kernel module: whenever I tried to type that dirty phrase it was immediately replaced. I tried some editors, sent naughty e-mails to myself and so on...great! Thanks a lot to the author; now I agree with Linus Torvalds: the real purpose of Linux is to have fun.

—Vladislav Belogroudov

Agrees with Jones

I would like to express my strong agreement with Daniel D. Jones, whose letter you published in the October 2001 issue of *Linux Journal*. The French chef gag employed in Marcel Gagné's Cooking with Linux column is irritating and tiresome. While the content of the column is of some value, I doubt I'm alone in finding the style annoying and unhelpful.

—John Doherty

Satisfied Diner

I look forward to my monthly visit to *Chez Marcel*. His topics are always interesting, his explanations have that extreme French lucidity, and I live in hope that François will pour me some of that Chambertin 1998. Perhaps Mr. Jones has a touch of the *mal de mer, non*?

—Peter P. Chase

El Cheapo Offends

“El Cheapo Linux” Mr. Rosen? Come on. How about “CheapLinux”? Living next to a major city like San Francisco in the state of California, I guess one just starts speaking Spanish without realizing it. A Spanish name does not imply low quality. But are you sure you haven't stepped on some legal toes using the name El Cheapo Linux? Google finds 2,440 pages with El Cheapo Linux.

—Steven M. Ruiz

Rosen replies: Mr. Ruiz, I'm sorry you took offense at my Spanish-sounding name for Linux. I suppose I could have used the made-up mark “Cheap Linux” to make the point that a trademark eventually becomes associated with a product and conveys many marketing messages based on the meaning of the words in common parlance as well as the mental associations created by advertising dollars. I didn't have to go to Spanish for that point. I used a foreign-sounding mark for another reason, although I failed to explain the reason in my article. (There is only so much one can say in about 500 words!) Many people assume that translating a trademark into a foreign language (or foreign-sounding language) will help them avoid trademark infringement claims. The law doesn't work that way. I doubt the US Trademark Office would allow registration of Manzana computers, or Linux Sombrero Rojo, or that Apple Computer or Red Hat would stand idly by while those marks were applied to competing goods. I suppose I could have chosen more precise Spanish (Linux Barato), French (Linux Vulgaire) or German (Linux Billig) for my example. Perhaps then someone else besides you would have taken umbrage, or I could have escaped without criticism at all. I did a quick Google search on “El Cheapo Linux”. Indeed there are many hits. But my quick scan found none that would qualify as a valid trademark. (That's a relief!) It appears that, even were a person to seek trademark registration for that mark, it would be disapproved as being merely descriptive, or perhaps now, generic. For what it is worth, I meant no disrespect. I represent a number of Spanish-speaking clients, I do pro-bono work for nonprofit Hispanic groups in the Bay Area, and I even lived in Guatemala when I was a child. So I'm not trying to sound like I'm disparaging your language (or any language) or the people who speak it.

Love the MUD

In the October 2001 issue Marcel turned me on to that delectable project called ALICE. In his article, he gave notice of an age old net entertainment called MUDs. I want to thank Marcel in cooking up such fine recipes in the past, and for adding a bit of MUD seasoning to his cuisine.

MUD games and Linux fit together hand in hand. Ask a few of the local geeks in the office, such as Alan Cox or Illiad, what a MUD is. Graphical games are

greatly enjoyed, but you just can't beat sitting down and role playing or hacking up your best bud through a text-based interface. Huzzah Marcel and *LJ*! Perhaps Monsieur François has a MUD pie recipe, no?

—Calvin “Kyndig” Ellis

Errata

The October 2001 issue of *Linux Journal* stated (on page 44):

> Since then, universities like Berkeley and companies like VA Linux have developed efficient software packages for cluster monitoring and have made them open source. We use a node-cloning package called SystemImager from VA Linux (www.valinux.com) to do software upgrades.

I would like to make the clarification that SystemImager was actually created and developed by yours truly, Brian Finley. VA Linux did release a version of my software, as did SGI, in a similar way that Red Hat and SuSE release their versions of Linux. Current and future official releases of SystemImager are available at systemimager.org.

—Brian Finley brian@systemimager.org

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

UpFront

Doc Searls

Issue #92, December 2001

Stop the Presses, *LJ* Index and more.

Free Bumper Sticker

If you're a Linux usin' geek of nature, you'll certainly want to advertise it. Get a free “geek by nature, linux by choice” bumper sticker simply by practicing your snail mail skills and sending a self-addressed, stamped envelope to:

Linux Journal, “Bumper Sticker Promo”PO Box 55549Seattle, WA 98155-0549

Within just two weeks you'll receive a “geek by nature, linux by choice” bumper sticker.

LJ Index—December 2001

1. Percentage of TimeWarner-related celebrities featured on AOL's starter page in a 30-day period: 78
2. Number of children who jumped up and down on one day in the UK hoping to cause an earthquake: 1,000,000
3. Tons of energy released by a million children jumping up and down: 75,000
4. Number of zSeries mainframe computers sold by IBM as of September 24, 2001: 1,000
5. Percentage of those mainframes on which Linux was installed: 10
6. Typical cost of a zSeries mainframe in millions of dollars: 1
7. Sum in millions of dollars pledged to Stanford University by Jim Clark: 150
8. Sum in millions of dollars suspended from the pledge by Jim Clark “pending the outcome of ongoing political deliberations” over stem-cell research restrictions by Congress: 60

9. According to a Cisco poster, number of cold beverages consumed per day by Cisco employees: 5
10. According to the same poster, annual savings in millions of dollars to Cisco from consumption of one less beverage per day per employee: 2
11. According to a web site responding to the poster, sum in millions of dollars Cisco spends daily venting carbon dioxide out of its buildings: 7
12. According to the same web site, sum in dollars saved monthly if every Cisco employee breathed four percent less: 140,000
13. Wireless manufacturing spending in millions of dollars by 2002: 884
14. Millions of Wi-Fi (802.11b) products installed by the end of 2001: 10
15. Percentage of companies that plan to allocate less than $250,000 to support wireless access: 60
16. Percentage of companies that expect to spend less than one million dollars by 2004: 61
17. Distance in feet traveled by the scramjet in 30 milliseconds: 5,325
18. Speed in miles per hour reached in the same flight: 260
19. Length in feet of the cannon from which the scramjet was fired: 130
20. Peak G-force acceleration of the scramjet in flight: 10,000
21. DARPA expenditures in dollars for the scramjet project: 800,000

Sources:

1: Drudge Report

2-3: Yahoo News

4-6: Bloomberg News

7-8: Jim Clark, in an open letter published in *The New York Times*

11-12: *The New York Times*

13-14: William Gurley, CNET, quoting Frost & Sullivan and Cahners In-Stat, respectively

15-16: Dow Jones Newswire

17-21: Slashdot

They Said It

Gee, everyone said that there'd be consolidation in the Linux space, but this is a bit bigger than I expected!

—Dave Sifry, on the HP-Compaq merger

Open Source and Complexity theory hold the strategic keys to managing risk in the age of terrorism.

—Eric Norlin

Tragedy purges the mind of trivia.

—George Gilder

Computers pose no threat to humans beyond Microsoft's blue screen of death and fatal-error messages.

—George Gilder

The problem [with the Internet] is that it was devised by a bunch of hippie anarchists who didn't have a strong profit motive. But this is a business, not a government-sponsored network.

—Tom Nolle

The Internet did not replace TV, newspapers, magazines, Sears, the US Postal Service, Barnes & Noble or grocery stores in people's daily lives. It augmented them.

—Jason Kottke

Proprietary software developers are all doing something wrong, but this doesn't mean they are all incompetent.

—Richard M. Stallman

Networking is simply the cultivating of mutually beneficial, give and take, win-win relationships. It works best, however, when emphasizing the “give” part.

—Bob Burg

Advertising is, and always will be, inherently ludicrous, and is generally deserving of satire.

—John Chambers

The Current Ports of Linux

“Linux is not portable (uses 386 task switching etc.), and it probably never will support any thing other than AT-hard disk, as that's all I have.” --Linus Torvalds, August 25, 1991 The Current Ports of Linux web site reminds us how far Linux has come by providing information and links on on architectures to which Linux is ported thus far. Visit www.cyut.edu.tw/~ckhung/resource/linux_ports.html.

New Web Site for Maragda

Jordi Bataller has a new web site for Maragda (see his article “Maragda: Running Linux from CD” in the March 2001 issue of *Linux Journal*): www.iti.upv.es/~maragda.

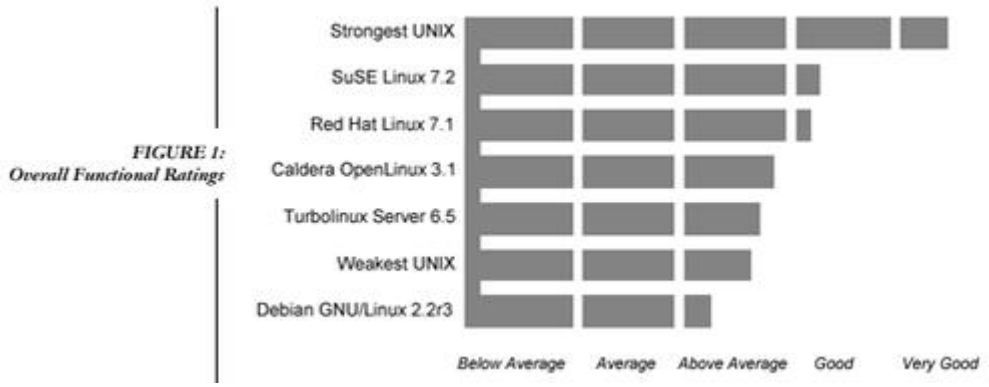
According to Jordi, the new site is faster, more reliable and includes a new version of Maragda, this time based on Debian 2.2r2 (potato).

Stop the Presses: Linux Scores as a Serious UNIX Business Platform

On September 25, 2001, the research firm D.H. Brown released a study that went a long way toward legitimizing Linux as a serious business platform and not just a handy way to serve up web pages and perform other low-level business grunt work. The report begins:

> For the first time since D.H. Brown Associates, Inc. began studying the functional capabilities of Linux-based operating systems, the strongest Linux distributions surpass the weakest UNIX systems in overall functionality. Using the version 2.4 Linux kernel has improved the features of the operating system. In addition, all of the vendors studied—SuSE, Red Hat, Caldera, Turbolinux and Debian GNU—have increased the breadth and depth of their bundled network infrastructure software.

The report analyzed scalability, RAS (reliability, availability, serviceability), system management, internet and web-application services, and directory and security services. “With these upgrades in place, the leading distributions of Linux are able to serve as general-purpose operating systems for a wide range of departmental and workgroup applications.” The graphic below shows how the various distributions stacked up.



Overall Functional Ratings

In a 1999 study, D.H. Brown articulated the prevailing business wisdom about Linux at the time: “that Linux was primarily suited for low-end web, file and print service, along with use in appliance servers and clustering for high-performance technical computing.” Now, the company says, “the leading Linux distributions are quite capable of serving as general-purpose operating systems for a broad range of departmental and workgroup applications.”

Where Linux fell short of the top UNIX systems, the report said, was in areas “such as high-end shared memory multiprocessing (SMP) scalability and advanced reliability features [which] require a degree of close coordination between hardware and software designers that has not yet emerged in the Linux community”. The report adds, however, that “the gap between Linux and these UNIX systems has certainly closed at a rapid pace.”

For more information, visit dhbrown.com/dhbrown/Linux.cfm.

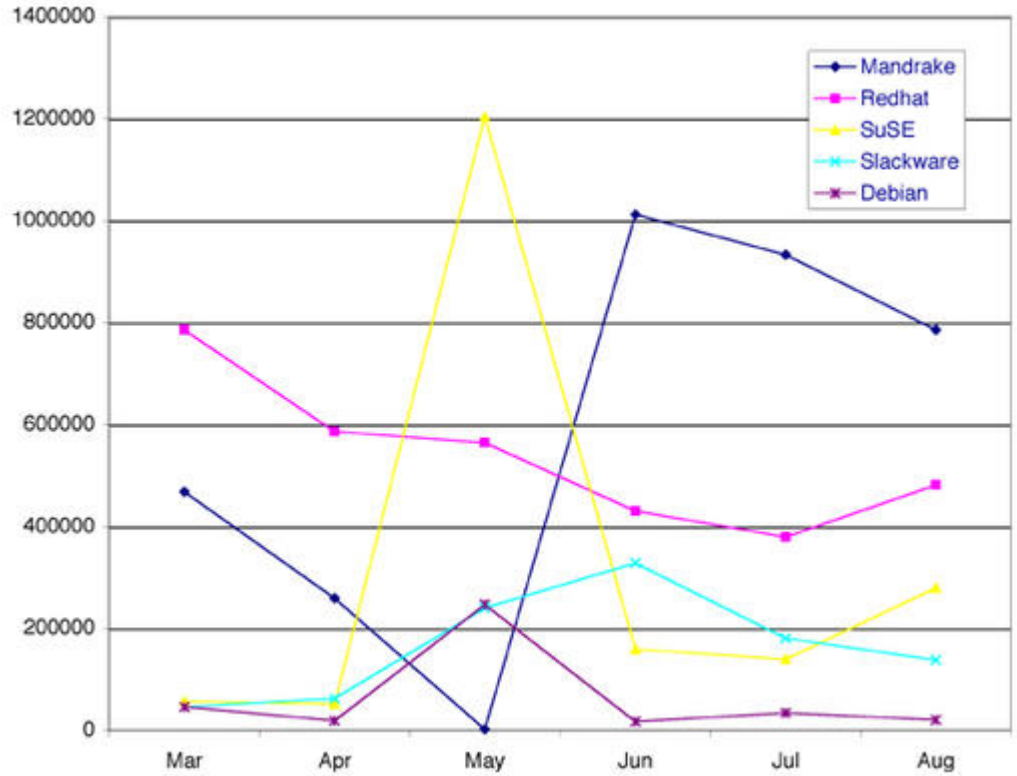
—Doc Searls

Mandrake Leads in Tucows Downloads

Tucows was the first of the big-time download sites to pay close attention to Linux, and the practice continues. In the past we've run occasional snapshots of one month's download percentages. This time we're looking at the six-month download trend running through August 2001 (see Figure 1).

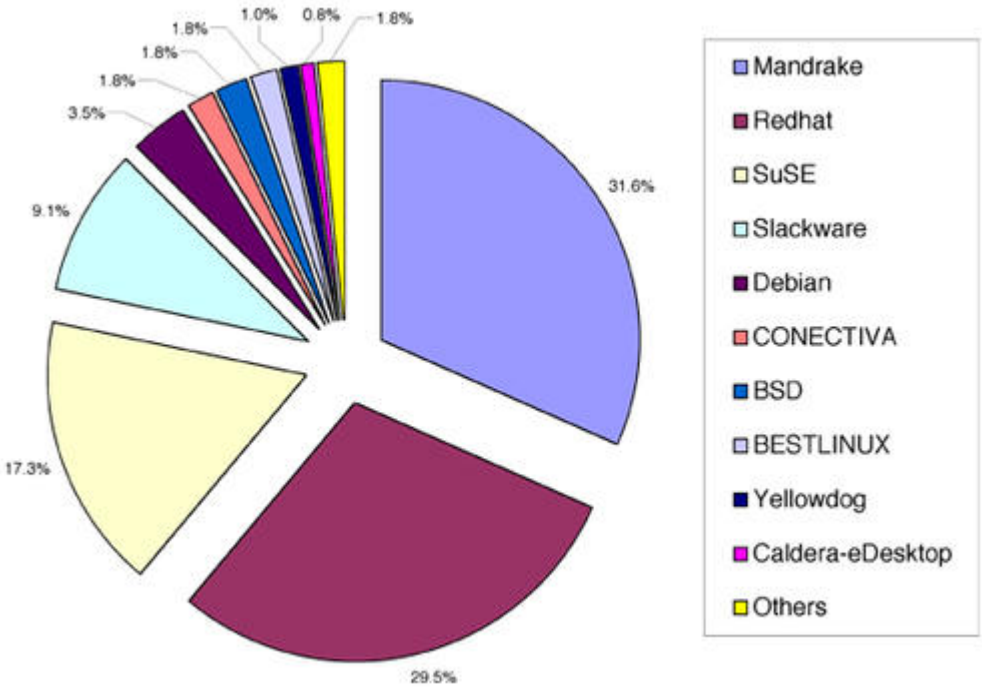
Mandrake has been a steady leader, with Red Hat a strong second (though with a somewhat downward trend over the period).

As we see from the huge spike in SuSE downloads in May 2001, a six-month total tends to even out any one month's extreme results. Still, it's easy to ponder that SuSE spike: it was over 1.2 million downloads.



Linux Distribution Download Trends, March-August 2001

The pie chart (Figure 2) shows shares of download totals for the whole six-month period, showing the top ten distributions. While there is a clear break below the Big Five shown in the trend chart, the Brazilian CONECTIVA distro is an interesting standout contender.



Linux Distribution Download Shares, March-August 2001

—Doc Searls

email: doc@searls.com

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

EFF Wants You

Richard Vernon

Issue #92, December 2001

There are worse ways of opposing the Security Systems Standards and Certification Act.

With the very real possibility of some version of the SSSCA being introduced to US legislators, threatening to make software that is either open source or source-available illegal, you may be asking yourself what a single individual can do that will have much effect, short of turning full-time activist.

Technical Editor Don Marti, recently made a comparison between the now burgeoning fight for the protection of digital rights and the 1960s fight for environmental protection that culminated in the formation of the EPA by President Nixon, pointing out that in our current fight we have opponents who are highly organized and that if we ourselves are not, we cannot hope to win. Those who wanted to protect the environment started as simply a bunch of individuals and small groups and became a powerful lobbying body and eventually even got a government agency assigned to protect their interests. Those individuals had at least two factors that worked to their victory—fierce opposition and popular support. The former we have, the latter we lack (at least it's nowhere near the level it needs to be to win the fight).

Rachel Carson's *Silent Spring* was, to a great extent, responsible for the degree of popular support that the early environmentalists enjoyed. Lawrence Lessig's *Code* could play the role of our *Silent Spring*, but obviously not nearly enough people are reading it.

One reader wrote in (jokingly) advocating a “Boston CD Party” where free software supporters would dump CDs, DVDs and eBook readers into the Boston harbor to call public attention to the plight of free software. He correctly points out that most of the general public doesn't use nor understand free software, so when presented with exaggerated claims by copyright-holding entertainment companies and the resultant economic “damage”, many will be

inclined to believe them and see defenders of free software as would-be thieves. The illogical fear of “hackers” instilled into the public consciousness by the popular media doesn't help either.

Fortunately the SSSCA has had its predecessors, and these previous (though less menacing) threats have led to the establishment of an organization for the protection of rights digitally expressed and for the education of the public on the dangers posed by the SSSCA and similar legislation—the Electronic Frontier Foundation. This organization has the potential to be a very powerful lobbying force, but it is up to those whose interests it strives to protect to support it by becoming members and/or supporting it with donations. If you want to learn more about the EFF, subscribe to their e-mail newsletter, EFFector (www.eff.org. Joining the EFF is an easy and effective way to participate in the defense of open-source software.

Richard Vernon is editor in chief of *Linux Journal*.

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

Best of Technical Support

Various

Issue #92, December 2001

Our experts answer your technical questions.

Best of Technical Support

128-Bit Precision with GCC

I am using the command xlC to compile C++ programs on a UNIX platform. When I need to increase the precision of my mathematical calculations from 64b-128b, the following command is helpful:

```
xlC128 -qldbl128 <filename> [-lm]
```

where -lm is used to include mathematical libraries, if required. Now, I want to perform the same calculation with the same precision on Linux. Please guide me as to how I should proceed. If any mathematical libraries are required, please also indicate the source from where I can obtain them.

—Pramod, l_pramod@hotmail.com

Get GMP (Gnu Math Precision), a free library for arbitrary precision arithmetic, operating on signed integers, rational numbers and floating-point numbers. You can find it at www.swox.com/gmp.

—Paul Christensen, pchristensen@penguincomputing.com

Upgrade Causes Weekly Crash

After I upgraded one of our systems to Red Hat 7.1 from 7.0, the system crashes frequently, something like once a week. The crashes always occur soon after 4:00 **A.M.** (after cron.daily is executed). Kernel output is

```
Unable to handle kernel NULL pointer dereference
at virtual address 00000000.
```

I upgraded kernel to 2.4.3-12, but there's not much improvement.

—Atsuko Crum, acrum@hood.edu

I had similar problems with a different distribution but a 2.4.x kernel. It turned out that a BIOS upgrade for the motherboard took care of most of my problems. Although I still have similar occurrences, they are much less frequent.

—David Brown, david@caldera.com

While hardware problems can happen at any time, if this started after you upgraded, you should try downgrading the kernel to the version you had on Red Hat 7.0, or if you don't mind rebuilding a kernel, you can try getting the latest 2.4 available (2.4.10 at the time of this writing). Many bugs were fixed in the first versions of the 2.4 kernel. If you want to rule out hardware issues, you can try an excellent stress tester, Cerberus, which you can get from SourceForge at sourceforge.net/projects/va-ctcs.

—Marc Merlin, marc_bts@valinux.com

Insufficient Space for Install

Slackware kept prompting that I had no space left for the installation to proceed. I could not believe it because I had 10GB reserved. I partitioned my hard drive like this:

5GB WinNT 4.0512MB for root512MB for swap4GB for usr4GB for home

—Cheppy, banggae@fisika.ui.ac.id

After formatting your partitions and setting up their mountpoints, switch to a second virtual console (Alt-F2) and verify, using df or mount, that they actually are mounted. If not, you will be installing Slackware on your 512MB root partition. That size may be too small if you are installing X or other large applications.

—Chad Robinson, crobinson@rfgonline.com

I am betting here that you are mislabeling the / (root) with a /root label. Try labeling your partitions like this: 5GB WinNT 4.0512MB /512MB swap4GB / usr4GB /home That should give you adequate space for the installation.

—Paul Christensen, pchristensen@penguincomputing.com

PAM on Slackware?

As I install more applications on my Slackware server for my users, I find that many require authentication, but there are a few that don't read data from the passwd file. The result is that users need to change their password in multiple locations.

There seems to be a movement toward PAM and LDAP so that if I could switch to support them, users could change their password to all of the services, including Samba, e-mail, pppd and phpgroupware from one location (such as a web-based password change application). However, Slackware doesn't support PAM, and I can't find a HOWTO to install PAM.

—Brian Johnson, bjohnson@jecinc.on.ca

It is possible to install PAM on a non-PAM-aware distribution, but it involves, among other things, replacing all your applications that do authentication with PAM-aware ones (and if your distribution doesn't provide them, you have to get the source, find PAM patches if they're not integrated and recompile everything with the right options for your system). It is a lot of work, and unless you're looking for a special challenge, I recommend you switch to a modern distribution like Debian Linux or Red Hat Linux (which both support PAM natively). Those are only two examples; many other distributions support PAM too.

—Marc Merlin, marc_bts@valinux.com

SCSI Emulation for Just One Drive

I have an HP IDE CD-Writer, so I am using SCSI emulation to use it with cdrecord. Under the 2.2.18 kernel, I could tell the ide-scsi module only to make use of my burner and leave my ATAPI CD-ROM alone. I did this via an append option in lilo.conf:

```
append="hdc=ide-scsi"
```

This worked well, as /dev/hdc is my burner and /dev/hdd is my ATAPI CD-ROM. Unfortunately, under the 2.4 kernel, this no longer works. The ide-scsi module grabs both devices, making /dev/hdd unavailable, preventing cdparanoia from working with it and forcing me to mount it with /dev/scd1. How do I get the same effect under 2.4 and force the ide-scsi module to emulate only /dev/hdc?

—Michael Soulier, michael.soulier@home.com

What you want is for hdc to do SCSI emulation and hdd to continue to be used as an IDE device. Usually IDE-CD support is disabled and SCSI emulation is enabled so that both drives are seen as SCSI devices. See www.wizball.co.uk/linux/cd_rewriter.php and www.teknospy.com/pages/howtos/cdburn.php for tutorials.

—Paul Christensen, pchristensen@penguincomputing.com

What Does This RPM Depend On?

I maintain 39 Red Hat servers and am always adding more (web hosts). I always try to install the bare minimum of packages during an install, figuring that if I don't include something the customer decides they want later, I can always figure out how to add it. Occasionally, a package requested will have such a fun list of dependencies requested that what should be a simple **rpm -ivh package.rpm** becomes a 30 minute dependency adding game. Is there a simple switch I can enter to have RPM tell me what one package needs before I try to install it?

—Bruce D. Meyer, gh1@rocsoft.net

You describe one of the reasons why many Linux server administrators use Debian Linux instead of a Red Hat derivative. With Debian, it is just a matter of typing **apt-get install foo**. Last I checked, RPM doesn't have a magic switch to do what you want. One way to do this on Red Hat is to use gnorpm or rpmfind. The problem is that in either case, with RPM-based solutions, RPMs are often incompatible across distributions and even within different versions of the same distribution (like RH 6.2 and RH 7.1). While rpmfind/gnorpm will probably be your best bet, if you get the option to install Debian Linux next time, it would solve all these problems.

—Marc Merlin, marc_bts@valinux.com

One little gem is Red Carpet. Documentation and downloads can be found at the www.Ximian.com web site.

—Paul Christensen, pchristensen@penguincomputing.com

Welcome, Guest Printer Users

I have a Linux box that acts as my DHCP server, firewall and gateway to the Internet. Many of my clients are visiting Windows users. Is it possible to set up the Linux box to be a print server such that a Windows client can print via the Linux print server without the client having to install printer drivers?

—Danny Patel, dharmesh@yahoo.com

I would set up a Samba Share for both the printer and the Windows drivers for that printer.

—Christopher Wingert, cwingert@qualcomm.com

You might want to look into using CUPS (Common UNIX Printing System). It supports IPP, LPD, SMB (Windows) and AppSocket (JetDirect) protocols.

—David Brown, david@caldera.com

cron Won't Run a Shell Script

I can't execute a shell script with crontab. I get this error:

```
cannot execute binary file
```

—Leonardo Fermoselle, lfermose@hotmail.com

Does your script start with #!/bin/sh as the very first characters of the file? It should. If this isn't the problem, was the script written on a DOS machine? Open the file with the joe text editor and if you see funny M characters at the end of every line, that is your problem. You can use a tool found at freshmeat.net/projects/fixdos to fix that.

—Ben Ford, ben@kalifornia.com

Run the command **file /usr/backup/myscript.sh**, or load this file in vi and make sure it does not have any junk characters in the file. Make sure that you can run the script from command line.

—Usman Ansari, uansari@yahoo.com

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.

New Products

Heather Mead

Issue #92, December 2001

VXA AutoRak, LinuxCad 3.0, Niveus 205 and more.

VXA AutoRak

The VXA AutoRak, available from Ecrix Corporation, is a rackmountable tape storage autoloader that stores up to 660GB of compressed data at a transfer rate of 21.6GB/hr. Designed to utilize up to ten data cartridges, the AutoRak stores data in an automated 2U form factor and can be used in standard 19-inch racks. The AutoRak has a control panel that allows users to configure and monitor their data backup and restore operations. An entry/exit port that can be locked for security measures and intelligent bar code readers are also available.

Contact: Ecrix Corporation, 5525 Central Avenue, Boulder, Colorado 80301, 303-402-9262, info@ecrix.com, www.ecrix.com.

LinuxCAD 3.0

Version 3.0 of LinuxCAD is now available and includes extensive support for 3-D graphics creation. All commonly used commands of Acad for 2-D and 3-D drafting, editing and displaying are implemented in LinuxCAD exactly as they are in Acad. LinuxCAD is integrated in the X environment, allowing users to edit the same file on multiple windows and multiple displays, copy parts of the drawing and work on ten or more drawings simultaneously on the same computer. LinuxCAD supports DXF, DWG, DXS, SLD and SHX graphics formats and is available for Intel-based systems, Solaris workstations and LinuxPPC.

Contact: Software Forge, Inc., 913-663-1724, sales@softwareforge.com, www.linuxcad.com

Plesk Control Panel

Rackspace Managed Hosting announces that the Plesk Server Administrator (PSA) 2.0, a multiplatform, web-based software tool that allows users to perform a variety of server administration tasks, is now available for Rackspace's Linux and UNIX servers. Users can create e-mail accounts and manage domains with Plesk's point-and-click interfaces. The PSA also allows service companies to share server management tasks with clients, employing three levels of web administration: Admin, Reseller Client and Domain Owner.

Contact: Rackspace Managed Hosting, Inc., 112 East Pecan, Suite 600, San Antonio, Texas 78205, 1-800-961-288 (toll-free), www.rackspace.com

Niveus 205

The Niveus 205 is an Intel-based workstation from Penguin Computing designed for 3-D graphic and application development. Included in the Niveus are dual Pentium III processors for up to 1.26GHz, a 133MHz front-side bus, ATA-100 hard drives and up to 1.5GB of PC133 RAM. Five PCI slots, one AGP 4x slot and three 5.25" bays are also provided, as are a 52x CD-ROM drive and a 3.5" floppy drive. Niveus workstations come with Red Hat installed. A variety of peripherals are available, such as Klipsh ProMedia speakers, LCD screens and high-end graphics cards, including the GeForce 3.

Contact: Penguin Computing, Inc., 965 Mission Street, Suite 600, San Francisco, California 94103, 888-736-4846 (toll-free), info@penguincomputing.com, www.penguincomputing.com

e/pop Server

WiredRed Software Corp. has made available the e/pop Linux Server, providing instant messaging (IM) and real-time communications software for small and large businesses. Released as part of the e/pop Standard Server Edition, e/pop Linux Server enables scalable and secure business communications with centralized administration, off-line message storage, and network and internet routing. Remote communication is achieved using a dial-in, VPN or internet connection. Security features include RSA 512-bit encryption with AES, DES, Triple DES or RC4. e/pop also allows text-based chat conferencing, VoIP conferencing and application sharing.

Contact: WiredRed Software Corporation, 4669 Murphy Canyon Road, Suite 108, San Diego, California 92123, 858-715-0970, www.wiredred.com

JSS 3.1

JSS 3.1 is Network Security Services for Java, a collection of packages that access a native implementation of cryptographic algorithms, allowing access to crypto accelerators and smart cards. With JSS, developers can communicate securely using SSL or TLS, process certificates, perform crypto operations, and parse and code arbitrary ASN.1 structures. SSL v2 and v3, TLS, PKCS #s 5, 7, 11 and 12, and x.509 v3 certificates are among the supported standards. JSS 3.1 is released under the GNU GPL; source code and binary distributions are available for download free of charge from the web site.

Contact: JSS Project Page, www.mozilla.org/projects/security/pki/jss

IEMS6

International Messaging Associates (IMA) announces the release of Internet Exchange Messaging Server (IEMS) version 6.0. The core of IEMS6 is the messaging application framework, an application development environment where administrators can build simple messaging-enabled applications and link them to e-mail, GSM, SMS and the Internet. Users can access applications in corporate and SOHO environments, on home networks, by SMS-enabled cell phones or any web-enabled device. IEMS6 offers calendaring and scheduling functions that support Linux, Solaris, HP-UX and Outlook back ends. Enhanced SSL support for SMTP and SMTP Auth Support, an attachment removal filter and a message storage encryption module also are included in IEMS6.

Contact: International Messaging Associates, Ltd., 27/F China Resources Building, 26 Harbour Road, Wan Chai, Hong Kong, sales@ima.com, www.ima.com

Archive Index Issue Table of Contents

Advanced search

Copyright © 1994 - 2019 *Linux Journal*. All rights reserved.