

100 Linux Tips and Tricks

by Patrick Lambert

Table of Contents

<u>Introduction</u>	1
Copyright	1
About the author	1
Acknowledgements	1
Audience	1
Organization	2
Web resources	2
<u>Installation</u>	4
What this chapter covers	4
Tip 1: Which distribution is good for you	5
Tip 2: How to find a Linux CD-ROM at low cost	6
Tip 3: Multiple operating systems	7
Tip 4: Installing with no CD-ROM drive or modem	8
Tip 5: Swap and memory	9
Tip 6: More swap with a swap file	10
Tip 7: Kernel size and modules	11
Tip 8: The boot prompt	12
Tip 9: Wrong memory size found	13
Tip 10: Master boot record and LILO	14
Tip 11: LILO can't find a kernel on a big drive	15
Tip 12: X Window configuration options	16
Tip 13: Allowing users to mount drives	17
Tip 14: Allowing users to run root programs	18
Tip 15: Linux and NT booting	19
Tip 16: Annoying boot messages	20
Tip 17: Programs on CD-ROM	21
Tip 18: International console	22
Tip 19: Multiple kernels choices	23
Tip 20: Default file permissions	24
Tip 21: Default boot mode	25
Tip 22: More information from usenet	26
Tip 23: Bytes per inodes	27
Tip 24: LILO and boot problems	28
Tip 25: Making CD-ROM images	29
Tip 26: FTP access restrictions	30
<u>Hardware</u>	32
What this chapter covers	32
Tip 1: Detecting 2 ethernet cards	33
Tip 2: Everything on sound cards	34
Tip 3: Non-PostScript printers	35
Tip 4: Use Windows special keys in Linux	36
Tip 5: Added processors	37
Tip 6: Detecting an ISA device	38
Tip 7: Find hardware information	39
Tip 8: Blinking leds on the keyboard	40

Table of Contents

Tip 9: Reading a foreign file system	41
Tip 10: Can't mount root fs	42
Tip 11: Linux on a 286?	43
Tip 12: Linux without a hard drive	44
Tip 13: Shutdown and power off	45
Tip 14: LPD started but no device found	46
Tip 15: Read files from FAT32 drives	47
Tip 16: TV on Linux	48
Tip 17: Device drivers	49
Tip 18: Mouse problems	50
Tip 19: International keyboards	51
Software	53
What this chapter covers	53
Tip 1: Background image in X Window	54
Tip 2: Customize Netscape Communicator	55
Tip 3: POP3 in Pine	56
Tip 4: Multiple accounts in Pine	57
Tip 5: Running Java programs	58
Tip 6: Virtual hosts in Apache	59
Tip 7: Libc versus Glibc	60
Tip 8: Aliases with Qmail	61
Tip 9: Samba with Windows 98 or NT 4	62
Tip 10: KDE drag and drop icons	63
Tip 11: Find files	64
Tip 12: asm or linux include files not found	65
Tip 13: ICO on Linux	66
Tip 14: Reading foreign documents	67
Tip 15: Scanning with Linux	68
Tip 16: Real audio and video	69
Tip 17: Emulation	70
Tip 18: Shared library not found	71
Tip 19: Hard to erase files	72
Tip 20: Files permissions	73
Tip 21: Changing file permissions	74
Tip 22: An international background	75
Tip 23: Powerful file transfer system	76
Tip 24: Editing in text editors	77
Tip 25: Documentation and manual	78
Networking	80
What this chapter covers	80
Tip 1: Easy PPP dialup	81
Tip 2: Internet for your LAN	82
Tip 3: Domains to search in	83
Tip 4: Display IP rather than hostname	84
Tip 5: Is my modem a winmodem?	85

Table of Contents

Tip 6: Sharing files from a Windows system	86
Tip 7: Sorry but this host is not in my list	87
Tip 8: Access to various networks	88
Tip 9: Accessing remote file systems	89
Tip 10: Secure Web server	90
Tip 11: Secure alternative to telnet	91
Tip 12: Speed problems on a PPP connection	92
Tip 13: Names and name servers	93
Tip 14: Who owns this port	94
Tip 15: Network printers	95
Development	97
What this chapter covers	97
Tip 1: Graphical messages to the world	98
Tip 2: Code reuse	99
Tip 3: Makefile don't equal C	100
Tip 4: Parsing the command line in BASH	101
Tip 5: Don't grep grep	102
Tip 6: Move a text into upper case letters	103
Tip 7: Using PASCAL on Linux	104
Tip 8: Segmentation fault	105
Tip 9: Who is online?	106
Tip 10: Graphical toolkits	107
Tip 11: IDE and visual interfaces	108
Tip 12: Free software and copyleft	109
Tip 13: Talking to the terminal	110
Tip 14: Internet technologies	111
Tip 15: Library types	112

Introduction

Copyright

This book is copyright by Patrick Lambert. It is provided free of charge in the hope that it will be useful. You may copy, distribute and print this book. You may not modify it without prior written consent from the author.

The tips in this book are given AS-IS. This means that I shall not be responsible for any damage that may occur from their use. You use them at your own risks.

Linux is a trademark of Linus Torvalds
RedHat is a trademark of RedHat Software Inc.
Windows and DOS are trademarks of Microsoft Corp.
Sound Blaster is a trademark of Creative Labs
PostScript is a trademark of Adobe
Other trademarks and copyrights may apply.

About the author

Patrick Lambert is currently a student in Computer Science at the University of Montreal in Quebec, Canada. He is the author of various Web sites for the Linux community, and of various software packages including [GXedit](#)

Although he does everything from systems administration to software programming, he spends most of his time working on Web sites for the Linux community. You can contact Patrick at drow@darkelf.net

Aknowledgements

I would like to thank Tuomas Kuosmanen for the logo and the images on the Web site.

Audience

This book was written for anyone using Linux, from new users to experts who want to explore this wonderful operating system. The tips and tricks in this book were discovered by myself over years of experience using Linux, and learning about it. Some are very basic tips to make your computing life easier, others are advanced tricks that can save you days of work.

I tried to cover all distributions of Linux in this book. I personally use Slackware and RedHat on PC systems. If you find any error in the book, feel free to contact me so a future second edition could correct them.

Organization

This book is divided into 5 chapters. Each chapter covers a specific topic:

- [Chapter 2](#) covers installation of Linux. These are tips and tricks useful when installing Linux itself or any new program. Some tips will cover new means of installing Linux on non-typical hardware, others will explain how to take Linux distributions from an FTP server and make your own CD-ROM with them, or where to find Linux CD-ROMs for as little as \$2.
- [Chapter 3](#) covers hardware related matters. You will learn tips there on how to get your non-PostScript compatible printer to work, or how to get a sound card detected.
- [Chapter 4](#) covers software. You will find tips there about all kinds of Linux software, including where to find and how to install the Java Development Kit port, and everything about the Pine mail and news program.
- [Chapter 5](#) covers networking in all its forms. There you will see how to setup a PPP connection quickly, without editing all of the configuration files yourself, as well as some nice programs that were made to ease dialup procedures. You will also see tricks on how to make your local LAN network without unexpected problems.
- [Chapter 6](#) is the last chapter but covers an important part of Linux: development. Here you will find a lot of tips on how to write powerful scripts to make your system easier to handle, and a full overview of what to do and what you don't want to do in C to avoid problems like memory leaks, and how to allow easy scalability.

Web resources

This book has a sister Web site at <http://tipoftheweek.darkelf.net> where some of the tips from this book can be found, and where you can submit your own tips to the site, to help the Linux community.

Installation

What this chapter covers

Installation is a very important part of any operating system. This is why I cover this topic first. The next most important thing is installation of programs and software to get your system to do useful tasks. This chapter covers both of these aspects.

Tip 1: Which distribution is good for you

They are all good. But that's not a real tip. What you should be looking for is which distribution you feel the most comfortable with. RedHat has the reputation of being very easy to install. They provide special tools to make the configuration easier. Debian also has some tools, but will usually require you to go on the command line more often to configure the system. If you want to be on your own, and really learn how to edit configuration files then Slackware is for you. The Web site <http://www.linux.org> lists all the available distributions.

In the end, the best person to decide which distribution you like, is yourself.

Tip 2: How to find a Linux CD-ROM at low cost

The Linux market started from a few distributions available only from FTP servers, to full feature commercial distributions available in stores and online including a printed manual and phone support.

Here are the main choices you have when looking for a Linux distribution:

- You can download any Linux distribution from its FTP server. To take a few examples, RedHat can be downloaded from <ftp://ftp.redhat.com>, Slackware from <ftp://ftp.cdrom.com> and Debian from <ftp://ftp.debian.org>. That method is free, but requires you to have a fast Internet connection. Downloading a full Linux distribution over a 56Kbps modem will take you quite a few hours.
- An other way is to buy a full distribution. RedHat, for example, can be bought online for about \$50. This will include a box, a CD-ROM, a boot diskette, a manual and support from RedHat.
- The last way is to buy only the CD-ROM. There are a few places selling CD-ROMs of various distributions for \$2. One of them is <http://www.cheapbytes.com>. You will only get the CD-ROM, but this is all you need to install Linux if you are comfortable with the fact that you don't get a printed manual or free support. You can find the manual and other documentation on the CD-ROM.

Tip 3: Multiple operating systems

A computer only needs one operating system to work. But what if you just want to try out a new system? Do you need to forget about the old one and erase your hard drive? No, you can have as many operating systems on your computer as you wish.

Linux requires 2 partitions to work. Partitions are sections of the hard drive. When you install Linux, it will provide a program called fdisk or disk druid allowing you to create the needed partitions. The main problems people have is that they don't have empty partitions to use for Linux, and they don't want to erase the current Windows or DOS partition. The trick is to resize your current partition to create empty space. Then you will be able to make the partitions needed by Linux to install properly.

Fdisk doesn't allow you to resize a partition. You will need to use another program to do the job, before using fdisk to create the Linux partitions. A very popular commercial product to do this is Partition Magic from <http://www.powerquest.com>.

Let's see step by step what is needed to resize an existing partition to allow the creation of a new one for Linux:

- Buy Partition Magic, or get any other tool that can safely resize partitions.
- Make sure you have at least 150 megs free on your main partition, the required amount for Linux.
- Resize the partitions so you have at least 150 megs free, outside of any current partition.
- Reboot and launch the Linux installation.
- Run fdisk or any partitioning program that comes with the Linux distribution, and follow the installation instructions to make the required Linux partitions.

Tip 4: Installing with no CD-ROM drive or modem

Most Linux distributions come on a CD-ROM. You can also download them from an FTP site, but that requires an Internet connection. What if you have a system with no CD-ROM drive or Internet connection, like an old 486 laptop? The trick here is to have another desktop system with a CD-ROM drive, and a null-modem serial cable.

I will show you how to do it with Slackware. It is also possible with most other Linux distributions. Insert the Linux CD-ROM in the drive on the desktop and copy the A (base) and N (networking) packages on diskettes. You need at least those in order to use a serial cable to transfer the rest of the packages.

Now you need to enable NFS networking on the desktop, and allow the laptop to connect. You can give a temporary IP address to the laptop, like 192.168.1.11 that you need to add to your /etc/exports file on your desktop.

To link the two systems together, this is what you need to type on the laptop:

```
/usr/sbin/pppd -detach crtscts lock 192.168.1.11:192.168.1.10 /dev/ttyS1 115200
```

And this on the PC:

```
/usr/sbin/pppd -detach crtscts lock 192.168.1.10:192.168.1.11 /dev/ttyS1 115200
```

This is assuming the cable is linked to ttyS1 (COM2) on both systems.

With NFS, you can mount the CD-ROM drive remotely and tell the installation program to use a specific path to install the remaining packages. Mount the CD-ROM with a command like this:

```
mount -tnfs 192.168.1.10:/cdrom /mnt
```

Then run the installation program:

```
setup
```

and enter the new path for the packages files.

Tip 5: Swap and memory

One important setting in any protected mode operating system like Linux is the swap space. In the installation, you will need to create a swap partition. A common question is what size should the partition be?

The proper size depends on 2 things: The size of your hard drive and the size of your RAM memory. The less RAM you have, the more swap you will need. Usually you will want to set your swap space size to be twice the RAM size, with a maximum of 128 megs. This of course requires you to have a hard drive with enough free space to create such a partition.

If you have 16 megs of RAM, making the swap space 32 megs or even 64 megs is very important. You will need it. If you have 128 megs of RAM on the other hand, you won't need much swap because the system will already have 128 megs to fill before using swap space. So a swap partition of 128 megs or even 32 megs could be enough.

If you don't select enough swap, you may add more later.

Tip 6: More swap with a swap file

You installed a new Linux system, but forgot to set enough swap space for your needs. Do you need to repartition and reinstall? No, the swap utilities on Linux allow you to make a real file and use it as swap space.

The trick is to make a file and then tell the swapon program to use it. Here's how to create, for example, a 64 megs swap file on your root partition (of course make sure you have at least 64 megs free):

```
dd if=/dev/zero of=/swapfile bs=1024 count=65536
```

This will make a 64 megs (about 67 millions bytes) file on your hard drive. You now need to initialize it:

```
mkswap /swapfile 65536  
sync
```

And you can then add it to your swap pool:

```
swapon /swapfile
```

With that you have 64 megs of swap added. Don't forget to add the swapon command to your startup files so the command will be repeated at each reboot.

Tip 7: Kernel size and modules

To configure Linux to detect a new hardware part, especially on a new kernel, you may need to recompile the kernel. If you add too many devices in the kernel configuration, you may get an error message telling you that the kernel is too big. The trick is to enable modules.

The kernel itself must be a certain size because it needs to be loaded in a fixed memory size. This is one reason why modules can be very handy. If you enable modules, you will need to make them:

```
make modules
```

and install them:

```
make modules_install
```

Then using the modprobe utility you can load selected modules on bootup. This way the kernel will be smaller and will compile with no error.

Tip 8: The boot prompt

The Linux system uses a program called LILO to boot itself. This is the LInux LOader, and will load a kernel and can pass various parameters. This is what the "boot:" prompt is for.

At the "boot:" prompt, you can enter a lot of parameters. You can send parameters to drivers like the ethernet driver, telling it at which IRQ the ethernet card is located, or you can pass parameters to the kernel, like memory size or what to do in a panic. Reading the LILO manual will tell you all of the nice things LILO can be used for.

Note that for device drivers compiled as modules, you need to pass values when you load these drivers, and not on the "boot:" prompt.

Tip 9: Wrong memory size found

The Linux kernel will detect various settings from your computer configuration. This includes the size of memory you have. In some cases, it will find the wrong size. For example, it could find only 64 megs of memory when in fact you have 128 megs.

The trick here is to specify the amount of RAM memory you have with the "mem=" parameter. Here is what you would type when your system boots if you have 128 megs of memory:

```
LILLO boot: linux mem=128M
```

This will tell LILO to load the linux kernel with 128 megs of memory.

Tip 10: Master boot record and LILO

What is the master boot record (MBR) and why does LILO erase the old boot loader? Every hard drive has a top space called the MBR where the BIOS will try to load an operating system. Every system has its own loader. DOS has DOS-MBR, Windows NT has the NTLDR and Linux has LILO.

When you install LILO, you can install it in the MBR or in a boot record for the Linux partition. If you want to keep your current boot loader, you can select the Linux partition, and make sure it is the active partition in fdisk. This way you will be able to boot to LILO, and then boot the old loader from the MBR.

If you plan on only using Linux on your system, you can tell LILO to boot right into Linux and not display a "boot:" prompt, and you can install it in the MBR.

Tip 11: LILO can't find a kernel on a big drive

On some big hard drives, LILO can have problems loading your kernel. The problem is because the hard drive has more than 1024 cylinders.

The trick is to make sure your kernel is in the first 1024 cylinders so LILO can find it. The way to do this is to make a small /boot partition at the beginning of the drive, and make sure the kernel is in the /boot directory. You can set the partitions in fdisk, and select the right path for the kernel in /etc/lilo.conf so LILO knows where it is. When you compile your kernel, simply move the new kernel in that directory so LILO can load it.

Tip 12: X Window configuration options

Each Linux distribution has its own X Window configuration program. XFree86 also has a text-based configuration program which is complex to use. But what if both the distribution program and `xf86config`, the text-based configuration for XFree86, do not seem to do what you need? XFree86 also comes with a graphical configuration tool.

The name of the graphical program is `XF86Setup`. This will launch a graphical window and allow you to configure the X Window Server. So if you don't like the console configuration programs, you can use this one:



Tip 13: Allowing users to mount drives

By default, Linux will not allow users to mount drives. Only root can do it, and making the mount binary suid root is not a good idea. With a special command in the `/etc/fstab` file, you can change that.

This is a typical line for the fd0 (A:) drive in `/etc/fstab`:

```
/dev/fd0          /mnt              auto              noauto,user 1    1
```

The keywords here are `noauto` and `user`. `Noauto` tells mount not to try to mount a diskette on boot, and `user` allows any user to mount the drive into `/mnt`. The `auto` keyword is also interesting. It tells mount to try to find out which file system is on the diskette. You could also use `msdos` or `ext2`.

Tip 14: Allowing users to run root programs

When a user starts a command, it runs with the permissions of that user. What if you want to allow them to run some commands with root permissions? You can, and that's called `suid`.

You can set a command to be `suid` root with the `chmod` command. This will make it run as root even if a user starts it. Here is how to set `mybin` `suid` root:

```
chmod +s mybin
```

Note that you must be very careful with this option. If the command has any security hole, or allows the user to access other files or programs, the user could take over the root account and the whole system.

Tip 15: Linux and NT booting

Some people choose to have both Windows NT and Linux on the same system. Windows NT has its own boot loader called NTLDR and Linux has LILO. Which should go on the MBR?

The safest way is to install Windows NT first, and give it the MBR. Then, when you install Linux, tell LILO to install on the Linux partition. Also set the Linux partition as the active partition. When the system boots, LILO will be loaded, and if you want to boot Windows NT, then LILO can load the MBR with NTLDR in it. There is a mini HOWTO text covering this subject available at <http://metalab.unc.edu/pub/Linux/docs/HOWTO>.

Tip 16: Annoying boot messages

When recompiling your kernel, you might end up seeing strange messages on bootup like:

```
modprobe: cannot find net-pf-5
modprobe: cannot find char-major-14
```

These are messages from the modules loader telling you that he can't find specific modules. This usually happens when you compile modules, but modprobe tries to load modules that were not compiled and it can't find them. The way to remove those messages is to set the modules to off. In the file `/etc/conf.modules` you may want to add:

```
alias net-pf-5 off
alias char-major-14 off
```

This will stop modprobe from trying to load them. Of course you could also try to resolve the problem by compiling the modules and make sure modprobe knows where they are.

Tip 17: Programs on CD-ROM

<http://metalab.unc.edu>, <ftp://ftp.cdrom.com> and more are sites with a lot of programs available freely for Linux. But you may not want to download gigabytes of data over a slow Internet link.

Several places offer a bunch of free programs on CD-ROM. <http://www.cheapbytes.com> and <http://www.linuxmall.com> are 2 places that can sell multiple CD-ROMs with all those programs for a very low price:



Tip 18: International console

Most Linux distributions are configured to use a US english keyboard. If you need to write on a french or any other kind of keyboard, you will want to change the locale so special keys like accents appear in the console.

The way to do this is to change the system locale with a program called loadkeys. For example, to enable a canadian-french locale, you need to add this line in your startup files:

```
loadkeys cf
```

Here cf means the canadian-french keyboard. Other locales are us, fr and more.

Tip 19: Multiple kernels choices

When you compile a new kernel, you will often change your configuration. This means you may forget to include an important driver, like the IDE driver, or otherwise make your system unbootable. The solution is to always keep your old kernel.

When you compile your kernel, the compilation procedure will often copy your old kernel into `vmlinuz.old`. If it does not, you can do it manually. What you should do is add an entry to `/etc/lilo.conf` allowing you to boot your old kernel. You should view the lilo man page for the complete syntax. You could also add entries for different kernels, for example if you want to have an older stable version of the kernel and the newest development version on your system.

Note that some distributions name their kernel with the version they represent. For example, your current kernel may be `/boot/vmlinuz-2.0.36-0.7`

Tip 20: Default file permissions

When you create a file, the system gives it default permissions. On most systems the permissions are 755 (read, write and execute for the owner, and read and execute for others).

This default is setup with the umask command. To use the command, you need to find the right octal number to give it. The permissions in the umask are turned off from 666. This means that a umask of 022 will give you the default of 755. To change your default permissions from 755 to 700, you would use this command:

```
umask 077
```

Tip 21: Default boot mode

When a Linux system boots, it loads the kernel, all its drivers, and the networking servers, then the system will display a text login prompt. There, users can enter their user names and their passwords. But it doesn't have to boot this way.

There are 3 modes defined in most Linux distributions that can be used for booting. They are defined in `/etc/inittab` and have specific numbers. The first mode, also called runlevel 1, is single user mode. That mode will only boot the system for 1 user, with no networking. Runlevel 3 is the default mode. It will load the networking servers and display a text login prompt. Runlevel 5 is the graphical mode. If you have X Window installed and configured, you can use it to display a graphical login prompt.

The way to change this is to edit `/etc/inittab` and change the `initdefault` line:

```
id:3:initdefault:
```

Changing a 3 to a 5 will make the system display a `xdm` graphical screen on bootup.

Tip 22: More information from usenet

There are newsgroups about everything. Newsgroups on the latest TV show, on gardening, and more. There also are newsgroups on Linux. In fact, the best help can be obtained from newsgroups. But which ones?

Here is a list of a few newsgroups dedicated to Linux, and what they are used for:

- `comp.os.linux.advocacy`: This newsgroup is used for advocacy. People stating their opinions about Linux or Linux applications, and about Linux competitors. Some post facts, some will flame other people.
- `comp.os.linux.setup`: This is a general purpose setup help group. Users will post questions and get answers there.
- `linux.*`: There now is a linux section on usenet. Currently there are more than 150 groups in `linux.*` and they are all about Linux!

Make sure you read the FAQ and rules of every newsgroup you want to post to.

Tip 23: Bytes per inodes

When you format a partition using Linux's primary file system, ext2, you have the choice of how many bytes per inode you want. From the man page:

```
-i bytes-per-inode
    Specify the bytes/inode ratio. mke2fs creates an
    inode for every bytes-per-inode bytes of space on
    the disk. This value defaults to 4096 bytes.
    bytes-per-inode must be at least 1024.
```

This means that by using a smaller size, you will save disk space but may slow down the system. It is a space/speed trade off.

This is similar to one of FAT16/FAT32' major differences.

Tip 24: LILO and boot problems

When a computer starts, the number of beeps the BIOS outputs tells you the state of the computer. On some computers, one beep means all is ok, but 2 beeps mean there is an error. LILO uses the same kind of codes. The number of letters you see from the word LILO on the screen says what is wrong. The whole word means everything is fine, only LI means only the first part of LILO could be loaded. A full description of this is available from the Bootdisk HOWTO.

When LILO can't load, it's a major problem. This often means that the boot code was corrupted. The only way to boot is from a floppy disk. In RedHat, you can use the rescue disk, in Slackware, you can use the boot disk with the "mount" image.

When LILO is fine, it's often easier to figure a boot problem. If the kernel panics when it tries to boot, it is usually due to a configuration error. You can tell LILO to mount another kernel you may have, like a "safe" or "old" image you kept for these cases. If the problem is in initialization scripts, you can tell LILO to boot directly into a shell with the following boot command line:

```
LILO boot: linux init=/bin/sh
```

Where "linux" would be your kernel image.

Tip 25: Making CD-ROM images

With other operating systems, such as Microsoft Windows or IBM OS/2, you are not allowed in the license to make your own CD-ROM with the OS on it and then distribute it.

Linux, being Open Source and free, can be copied. You can download a distribution or buy it from an online store and burn your own copy, and then install it on many computers, or give it to your friends. Usually, you will find instructions on how to do that on the FTP server for your favorite distribution. You will need the main directory on the CD-ROM. The sources are not needed since they are available from the FTP site.

Some distributions also come with ISO images of their CD-ROM. This is a single file that can be put onto a CD-ROM, and will create a full file system with files on it.

One thing you have to be careful is not to copy commercial programs. The basic CD-ROM where the Linux distribution is located is composed of free software. But some distributions may come with other commercial programs, and you should read the license first.

Tip 26: FTP access restrictions

When you first install Linux, it comes with a lot of Internet services running, including mail, telnet, finger and FTP. You really should disable all those that you don't need from `/etc/inetd.conf` and your startup scripts.

FTP may be very useful, but must be configured correctly. It can allow people to log into their accounts, it can allow anonymous users to login to a public software directory, and it can display nice messages to them.

The files that you will probably want to modify are `/etc/ftpusers` and `/etc/ftpaccess`.

The file `/etc/ftpusers` is very simple. It lists the people that will not be allowed to use FTP to your system. The root account, and other system accounts should be in that file.

The file `/etc/ftpaccess` is a bit more complex and controls the behaviour of the FTP server. It tells it what to use as README file to display on a directory listing, what kind of logs to create and what messages to display.

Note that if you create an anonymous FTP area, you will need to read the FTP man page and do exactly what it tells you to avoid possible security risks.

Hardware

What this chapter covers

Hardware support has once been a very big problem with Linux. Generic hardware always was well supported, but most of the hardware today is unfortunately non-generic. 100% Sound Blaster compatible cards often are not detected by the Sound Blaster driver, and non-PostScript printers don't accept PostScript input unless a program previously converted it. These are the kind of problems we solve here.

Tip 1: Detecting 2 ethernet cards

To configure an ethernet card in Linux, you need to enable it in the kernel. Then the kernel will detect your ethernet card if it is at a common IO port. But it will stop there, and will never check if you have 2 ethernet cards.

The trick is to tell the ethernet driver that there are 2 cards in the system. The following line will tell the kernel that there is an ethernet card at IRQ 10 and IO 0x300, and another one at IRQ 9 and IO 0x340:

```
ether=10,0x300,eth0 ether=9,0x340,eth1
```

You can add that line on bootup at the "boot:" prompt, or in the /etc/lilo.conf file. Don't forget to run:

```
lilo
```

That will reload the lilo.conf file and enable changes.

Tip 2: Everything on sound cards

A sound card can be easy or hard to detect. It depends on who made it. Many Sound Blaster Compatible cards in fact are not compatible with the Linux Sound Blaster driver. Other cards will be compatible with a driver you would never have thought of.

There are multiple drivers for Linux. Currently the kernel comes with its own set of sound drivers, plus the OSS/Free drivers. These will support most generic cards. In the installation program, or when compiling your kernel, you can pick the sound card that matches yours, or the one that matches the chipset on your sound card (for example, the Sound Blaster PCI64 card uses the AudioPCI chipset).

If your sound card is not supported by the kernel, you will need to get another driver. 2 popular ones are ALSA available from <http://alsa.jcu.cz> and OSS/Linux available from <http://www.opensound.com>

OSS/Linux is a commercial product that supports a lot of cards not available in other drivers because of card specification restrictions. You will need to see the list of supported cards in each driver and pick the driver you need.

Tip 3: Non-PostScript printers

Unfortunately, most printers are non-PostScript compatible. This means that your LPR program won't like it.

You will probably notice that when you first use 'lpr' to print, the output looks weird on your printer. This is because these models do not support PostScript. You will need a converting program for it.

Note that newer versions of RedHat already have those programs or similar filters so it may not apply to all Linux systems.

First, you need to go read the Printing HOWTO to find out how to use lpr and related printing programs.

Then, you'll need to get 2 programs from <http://metalab.unc.edu>:

- `bjf`
- `aps`

These are the filters to convert text and PostScript to your printer's format.

First, install bjf which will be used to print text. Installation is very simple. type:

```
make
cp bjf /bin/bjf
```

Then, make a simple shell script to print text files and call it print.sh:

```
#!/bin/sh
/bin/bjf <$1> /dev/lp0
```

Where /dev/lp0 is your printer.

Now, install aps by running the SETUP script in its package. It's really easy to setup, but you do need to have the GhostScript program installed prior to installation. You are now ready to print PostScript files from, for example, Netscape or XV.

Tip 4: Use Windows special keys in Linux

Why are all the new keyboards sold with Win95 keys on them? How about making them do real keyboard functions while in X Window? Here is how.

First you need to find out which key mapping you are using. Usually it will be US, it might also be en_US, ca or else. Locate the file, usually in /usr/X11/lib/X11/xkb and edit it with your favorite editor. For me the file is called /usr/X11/lib/X11/xkb/symbols/ca.

The file lists all the key codes and what they do. The key codes for the Win95 special keys are LWIN, RWIN and MENU. All you need to do is add them to the list, with the functions for them. I decided to map the left WIN key to "@" and the right WIN key and MENU keys to "{" and "}". Here are the lines I added:

```
key <RWIN> { [ braceleft ] };
key <LWIN> { [ at ] };
key <MENU> { [ braceright ] };
```

By browsing the file you can find all the other symbols and what they do. You can also add multiple functions to a key, by using ALT and SHIFT.

The changes will take effect when you restart X Window. With the XKB extension (you do need to have it enabled in /etc/XF86Config btw) it's easy to change the mapping of any key.

Tip 5: Added processors

Dual processors are becoming more and more popular in computers. Of course, you won't be able to see much performance increase in Linux unless you tell Linux about the second CPU. Here is how to do it.

Go in the kernel, and enable SMP. SMP means Symetric Multi-Processing and tells the kernel that more than one processor can be used.

After a reboot, Linux should tell you that it has detected 2 processors and what their status are.

Tip 6: Detecting an ISA device

ISA devices like modems should be detected automatically by the kernel. One case where they would not be detected is if the device is Plug and Play. The kernel won't be able to detect ISA PnP devices unless they have been prepared by a program called isapnptools.

The trick is to use this program in initiation scripts. It will detect the ISA PnP devices and make them available to kernel modules and applications.

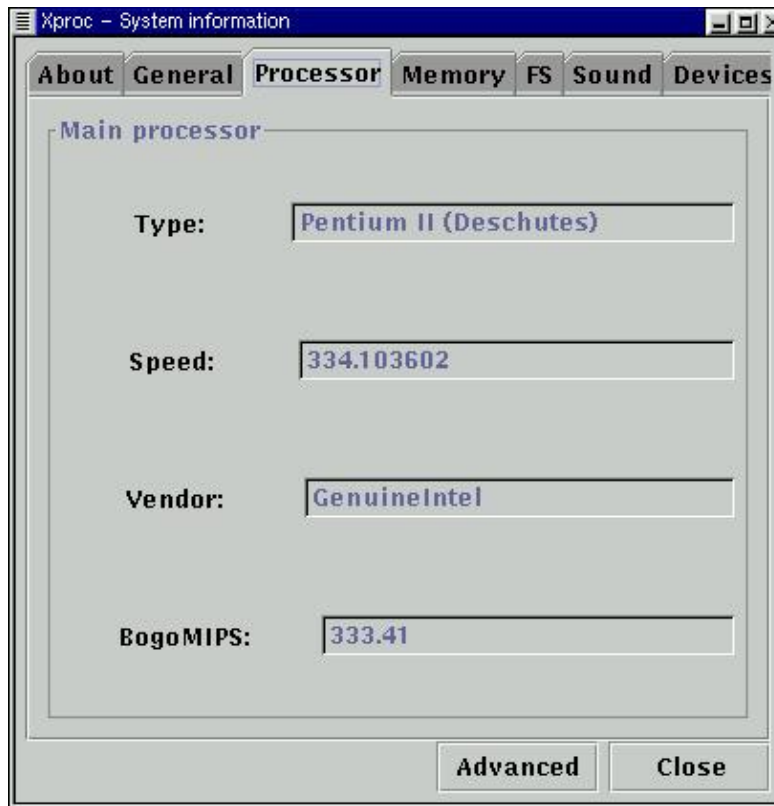
For example if you want to access an ISA PnP sound card, you will need to compile sound as a module in the kernel, and use isapnptools before loading the module.

Tip 7: Find hardware information

When the Linux system boots, it will try to detect the hardware installed in the computer. It will then make a fake file system called `procfs` and will store important information about your system in it.

You can get information about your system simply by browsing the directory `/proc`. The files in there will contain information such as the processor you have, the amount of memory and the file systems the kernel currently supports.

A usefull application exists to browse the `/proc` file system. It is called `Xproc` and is available from <http://devplanet.fastethernet.net/files.html>:



Tip 8: Blinking leds on the keyboard

Keyboard leds are pretty boring. Usually you know if the num lock is on or if you are writing in upper case letters. Could we make them do something more useful? Of course.

The keyboard leds can be controlled by a device driver called the misc driver. That driver can control all kinds of misc things. You could write your own driver to make them blink or light up at any system event.

A program called tleds is available from <http://www.hut.fi/~jlohikos/tleds.html>. That program will have them blink based on network usage. One will blink when packets go in, and the other when packets go out.

Tip 9: Reading a foreign file system

File systems are defined in the kernel. The kernel supports many file systems, but they need to be compiled in, or compiled as a module. When you compile a file system in the kernel, all you need to do is use mount with the -t option and the right file system type.

If you have compiled a driver for a file system as a module, then you need to load the module first.

Here are a few of the available file systems:

- msdos: This is the FAT file system used by DOS.
- vfat: This is the FAT32 file system used by Windows 95 and Windows 98.
- ext2: This is the default Linux file system.
- iso9660: This is the default CD-ROM format.

Tip 10: Can't mount root fs

When you boot a system, this is an error that will halt the system. This error means that the kernel can't mount the root file system, so it can't get all of its configuration files.

There are a few cases where this happens:

- No IDE support in the kernel. If your main hard drive is an IDE, and you have recompiled without including "Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support" or some other basic IDE device drivers.
- File system not supported. Usually the root file system should be of type ext2. You need to make sure that ext2 is compiled in the kernel, and not as a module because you can't load modules without first mounting the root file system.
- The drive is not ready. If you have removed the drive, the hard disk died or the BIOS didn't detect the device for some reason, then the kernel won't be able to mount it.

Tip 11: Linux on a 286?

Linux is a multi-user, multitasking operating system which requires a 386 processor or more to run. This is because the lower PC processors don't have what is needed for the Linux kernel.

Linux being open source, is being ported to many architectures. And one group decided to modify the kernel so it would run on a 286 processor. The project page is <http://www.uk.linux.org/ELKS-Home/index.html>.

Tip 12: Linux without a hard drive

Modern Linux distributions require around 100 megs to 200 megs of hard disk space to install. But is it possible to run Linux on a system without a hard drive? Yes it is.

The Linux Router Project is a full-featured Linux distribution that fits on one diskette. It was made for routers, and use modules to add the software packages you need, including DNS servers, Web servers, email and routing. You can find more information on the Linux Router Project at <http://www.linuxrouter.org>.

Tip 13: Shutdown and power off

Linux, like most other operating systems, must be shutdown in a specified manner. You can't turn the power off on a computer running Linux, or you may lose some data.

Here are the steps that need to be done when you want to shutdown:

- The operating system needs to stop all the running processes and logout the users.
- Various servers need to be shutdown in a proper way.
- All the mounted file systems need to be unmounted safely and unwritten data need to be stored on the disk.

The system can then be turned off safely.

To accomplish all these tasks, the shutdown command exists. That command has a lot of options, and you should explore them before trying anything at random. Another way to shutdown a Linux system is to set the runlevel to 0, the default shutdown level, with the init program.

Tip 14: LPD started but no device found

The printer daemon is called LPD. It will be started at boot time and assume that a printer is connected to the printer port. But a problem may occur when the daemon is started and no device is found.

The most common cause for this problem is a configuration problem in the kernel. Make sure that parallel port, PC-style parallel port, and printer support is enabled in the configuration, and that modules are loaded.

Tip 15: Read files from FAT32 drives

The Linux utility `mount` can read any file system that the kernel supports. Since version 2.0.34, the kernel supports FAT32, which is the main file system used by Windows 98.

To read the FAT32 file system you need to specify the `-tvfat` option to `mount`. Here is an example:

```
mount -tvfat /dev/hdb1 /mnt
```

Tip 16: TV on Linux

A number of cards exist allowing you to watch and record TV on your PC. Most come with software for Windows only, like so many things, but it is possible to do the same thing on Linux.

Linux comes with several drivers which make up the Video4Linux drivers. Several cards are supported by these drivers, and a list of them is available at <http://roadrunner.swansea.linux.org.uk/v4l.shtml>. This is the driver side. You also need software to use the devices.

Several programs are available to watch TV, capture images and even Web applications. A list of some of the programs is available at <http://www.thp.uni-koeln.de/~rjkm/linux/bttv.html> including datasheets.

Tip 17: Device drivers

Hardware devices are not of much use without device drivers. Fortunately, the Linux kernel, like every other operating system, comes with a lot of them.

When you configure a kernel, the menu from which you must choose which devices you have in your computer is actually a list of device drivers available to you. Here is how to configure your kernel:

```
cd /usr/src/linux
make menuconfig
```

Linux, still not being as popular as some other operating systems, can't support all of the existing hardware. If you have a device that is not in the list, then you will need to search for it on the Web. Some drivers may exist for Linux and not be in the default kernel. But if a device is not currently supported by any driver for Linux, then you will have to wait for someone to make one, or make one yourself.

If you want to write your own driver, the best place to start is at The Kernel Hacker's Guide, available at <http://khg.redhat.com/HyperNews/get/khg.html>

Tip 18: Mouse problems

The mouse is a very important part of a computer. In X, you can't do much without it. Even in console some applications will allow you to use it. You must first configure it.

Like every other device, the mouse has a device file in the `/dev` directory. Depending on your mouse type, it will be either `/dev/psaux` for a PS/2 mouse, or `/dev/cua0` for a serial mouse on COM1.

To make it easy to use in the future, you should link the right device file to `/dev/mouse`:

```
ln -s /dev/cua0 /dev/mouse
```

Then, when you configure X Window or any other program that requires a mouse, you can specify `/dev/mouse`.

Tip 19: International keyboards

The X Window System comes by default with an english keymap on most distributions. If you have a keyboard from another country, it may be a problem.

The XKB extension was provided for that purpose. If you look in your X configuration file, usually in `/etc/X11` called `XF86Config`, you will find a section about the keyboard. There, you may specify several parameters, including the keymap and keycodes to use.

For a list of available keymaps, you can look in your X directory. Most of those keymaps are also available from the graphical configuration program, `XF86Setup`.

Software

What this chapter covers

Software is what makes the system work. Without software, a computer system would be useless. We cover important software issues here, and how small programs found on remote Web sites can solve most of your problems.

Tip 1: Background image in X Window

By default, the background in X Window is a grey background. Some window managers have their own options to set other backgrounds, and some don't.

A program you can use to set your background is a program that comes with most Linux distributions. It's called XV, and you can specify which background image you want with this line:

```
xv -quit -root image.gif
```

This will set the background to be the image.gif file. You can add that line in your .xinitrc file, in your home directory, so it is executed each time you launch X Window.

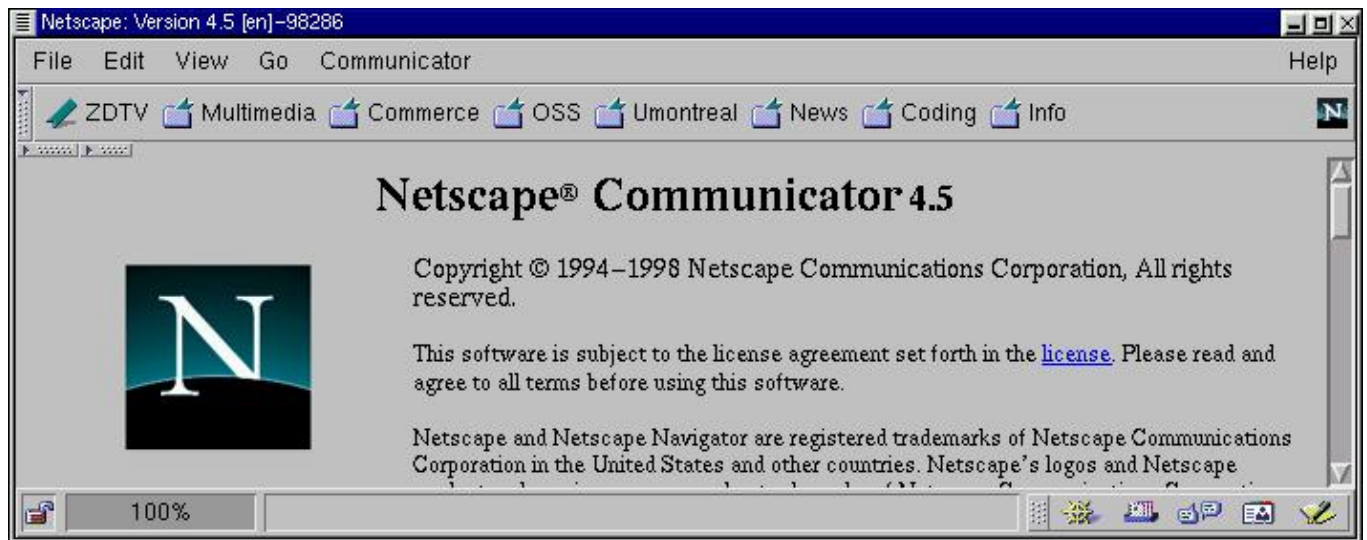
Tip 2: Customize Netscape Communicator

Netscape Communicator has one nice feature that few people use. When you download it, it has a Personal Toolbar with buttons that most of us may not find really useful, so we either turn it off or don't look at it.

I found it to be a much more useful tool than the Bookmarks. What appears on that bar is simply what's inside the Personal Folder inside your Bookmarks.

You can create folders in it for news sites, Linux sites, etc and move your bookmarks into them (with the Edit Bookmarks option).

You then have your own personal portal in your browser:



Tip 3: POP3 in Pine

Many Linux users use Pine as their mail client. Lets see how you can use Pine both with local mail and with POP3 mail, without using fetchmail or some other tool to get POP mail first.

First you will want to use multiple config files:

```
pine -p localmail
pine -p popserver1
pine -p popserver2
```

You need to configure Pine to use your POP3 server. In the program go to Setup, Config. You'll want to set something like this in your inbox-path:

```
{pop.server.com/pop3/user=myid}INBOX
```

Now restart Pine and it will ask you for your password and connect to the remote server, and use it just like if you were accessing local mail.

Tip 4: Multiple accounts in Pine

You can use multiple POP3, IMAP or other incoming mail accounts in the same config file by editing the line "incoming-folders=" from .pinerc in your local home directory.

The syntax is:

```
incoming-folders= "Nick1" {pop1.domain.com/pop3/user=login}INBOX,  
"Nick2" {pop2.somewhere.com/pop3/user=login}INBOX
```

This will create a new collection in Pine with these folders in it, plus the mail folder on the local system.

Tip 5: Running Java programs

Java is an interpreted language. Usually Java is found on the Web, in small programs called applets. But many Java applications exist. They are like applets, but require a Java Virtual Machine to run on your system. Netscape Communicator and any Java-enabled browser can run Java applets, but what if you want to run Java programs?

Java programs are files ending with .class and must be run in a JVM. The Java Development Kit comes with a JVM. What you need is the Linux port of the JDK that you can find at <http://www.blackdown.org>. Once installed, you can run any Java application using the line:

```
java Program
```

Where java is the JVM, from the JDK package, and Program is the class found inside the Program.class file. Note that you do not have to specify the .class part of the file to run it.

Tip 6: Virtual hosts in Apache

Apache, the popular Web server for Linux and Unix, allows you to host virtual hostnames with multiple IP addresses. You can set one IP to each hostname. But what if you want to host multiple hostnames on a single IP? Apache allows you to do it.

The trick is a single command that goes in the httpd.conf configuration file:

```
NameVirtualHost 1.2.3.4
```

Replacing 1.2.3.4 with your real IP address, this will allow Apache to know on which IP it should serve the virtual hostnames. Then you can add the virtual commands for every hostname you want to host:

```
<VirtualHost virtual.host.com>  
ServerAdmin webmaster@virtual.host.com  
DocumentRoot /home/httpd/virtual  
ServerName virtual.host.com  
</VirtualHost>
```

This will add virtual.host.com in your list of virtual hosts and serve the pages in /home/httpd/virtual. Of course you need to have the actual virtual.host.com hostname pointing to that system.

Tip 7: Libc versus Glibc

A few years ago, every Linux system used the libc version 5.x library. The libc library is the main library that all Unix C programs use. The functions they access are defined in that library.

Recently a new version of the library was introduced by the GNU project called GNU libc, or glibc, which is also called libc 6. Some distributions and older systems won't have glibc, and they won't support glibc binaries. This doesn't change anything for binaries you compile, but binaries you download that are linked to the GNU libc will not work.

There are 2 ways to get them to work. The first is to upgrade. You can upgrade to a newer version that does support glibc, or change your distribution to one that already has glibc as its main library. The other way is to install glibc on your current system. This requires you to know what you are doing and to backup your system first. See the GNU page at <http://www.gnu.org> page for more information on GNU products.

Tip 8: Aliases with Qmail

Qmail is a mail program that allows users to send and receive emails. It also has a lot of good features, one of them being aliases. You can add aliases to users who do not have an account on the system. Here is how to do it:

- Create an account called "alias" if it doesn't exist, and go in its home directory.
- Edit a file called `.qmail-username` where username is the alias you want to add.
- Add the address in the file with a `&` in front of it.

Here is an example of a `.qmail-username` file:

```
&me@mydomain.com  
&meagain@some.other.domain.net
```

This will forward any mail going to user "username" to the 2 listed email addresses.

Tip 9: Samba with Windows 98 or NT 4

Samba is a product allowing you to connect your Linux systems to Windows systems for file and printer sharing. By default the SMB protocol used by Samba sends passwords in clear text. Windows 98 and NT use encrypted passwords.

If you can't seem to connect from a Windows system on a Samba server, it might well be because the Samba server doesn't know about the encrypted passwords. You need to read the documentation about encryption in the Samba package. There is a utility to convert a Linux `/etc/passwd` file to encrypted mode so Samba can support Windows encrypted passwords.

Note that Samba also works with Windows 3.1 and Windows 95, with no encryption.

Tip 10: KDE drag and drop icons

KDE is a window manager for the X Window system. It comes with a toolbar at the bottom of the screen and default buttons. You can drag icons from your desktop to the toolbar. The problem is that there is no option to add icons.

The trick is to drag a file on the toolbar and then edit it. Open KFM and drag the program you want on the toolbar. You can then edit it and change the settings of the new icon in the toolbar:



Tip 11: Find files

Find is a very useful and powerful utility. It is often used by system administration and in shell scripts. Here are 2 commands that might be useful:

```
find / -perm -4000 -print
```

This command will find every file on the system that is `suid`. This means that when you run it you will be running it as an other user. For example, `traceroute` is a utility that needs to be run as `root`. To allow users to run it, systems administrators will set it `suid root` so it will be run as `root` even if a user starts it. This can be useful, but can also be a big security risk if the utility has a security hole in it.

Here is another interesting command:

```
find / -atime +10 -print
```

This command will find all the files accessed more than 10 days ago. Commands like this one can be useful to find old files that need to be backuped or erased.

Tip 12: asm or linux include files not found

On older Linux systems, you may have some errors about include files not found in the asm/ or linux/ directory when you try to compile some programs. This means that the kernel source code was not installed. Both asm/ and linux/ directories are links to the kernel source code.

To install them on your system, you need to get the kernel source code. Then you need to link the directories:

```
ln -s /usr/src/linux/include/asm /usr/include/asm
ln -s /usr/src/linux/include/linux /usr/include/linux
```

This will link the 2 directories in the include tree allowing applications to find them.

Tip 13: ICQ on Linux

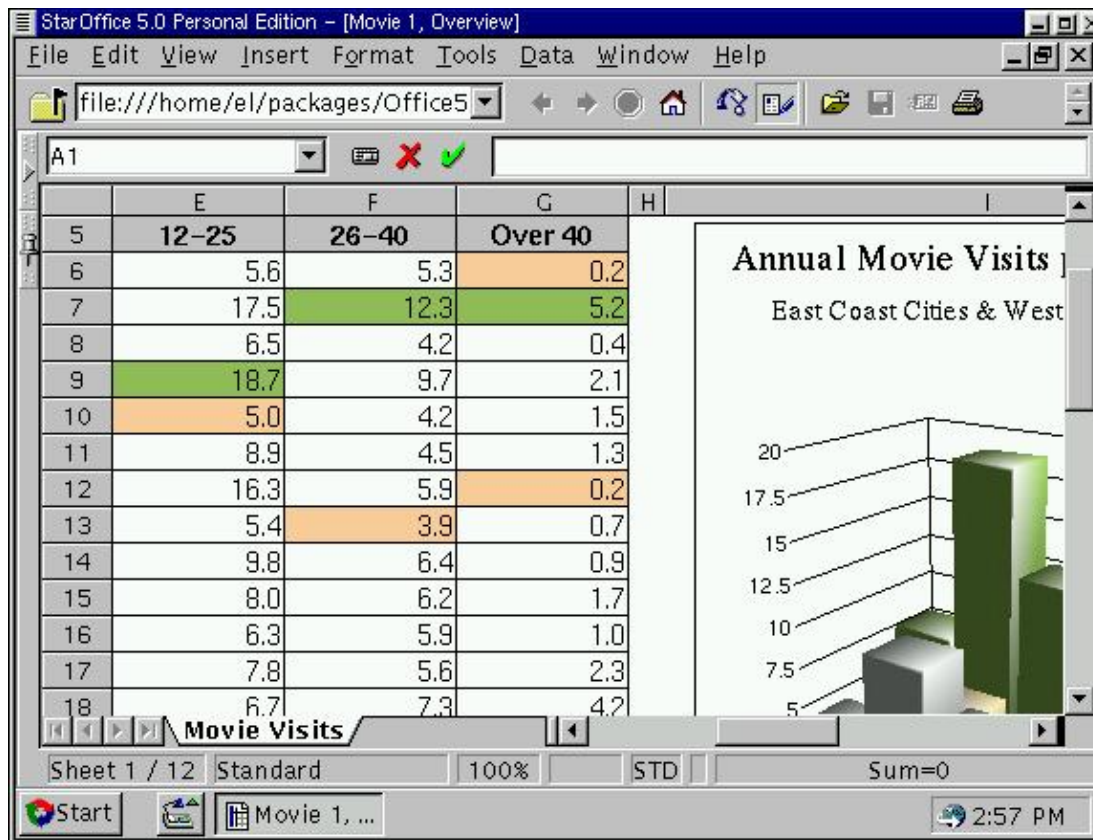
The original ICQ client does not currently run on Linux. Fortunately, there are clones that do run on Linux and offer the ICQ service to users. A popular ICQ program is Licq available from <http://licq.wibble.net>. Another popular one is Micq available at <http://micq.chatzone.org>. Other ICQ programs are also available for Linux. This is licq:



Tip 14: Reading foreign documents

Linux comes with various freely available word processors and office suites. Unfortunately, Microsoft Office doesn't come with a Linux native port. But you can still read and write Word and Excel files.

Word Perfect is a Corel word processor product that has a Linux port. The latest version is Word Perfect 8. It is available freely for personal use on Linux, and will read and write Word files. It is available from <http://linux.corel.com>. Another program is Star Office. This product from Star Division looks very much like Microsoft Office and offers similar features. It will also read and write all kinds of file formats. Star Office is available freely for personal use from <http://www.stardivision.com>:



Tip 15: Scanning with Linux

A number of scanning programs exist for Linux. A popular one currently is SANE which stands for "Scanner Access Now Easy" and is available from <http://www.mostang.com/sane>. It is an application programming interface that provides standardized access to scanners.

Several graphical frontends are also available from the SANE Web page.

Tip 16: Real audio and video

Real Networks has released their Real Player for Linux. Unfortunately, many users had problems setting it up. When running an older kernel, you may need to disable the 16bits sound playback. When running the Linux kernel 2.2 with Real Player 5.0, you may need to use a workaround to get the Real Player to work. The workaround is available from the Web.

Real Networks provides the Real Player for Linux and for RedHat Linux. The RedHat Linux version comes in the RPM format. Here it is:



Tip 17: Emulation

Linux is source compatible with Unix. This means that all of the Unix programs should work on Linux when compiled correctly, with little or no change to the source. Unix does provide a wide variety of software programs, but some programs are only available on non-Unix systems.

A number of emulators are available on Linux. We'll see 4 of them:

- One of them is called WINE and stands for WINE Is Not an Emulator. It used to stand for Windows Emulator. WINE will run various Windows 16bits and 32bits applications. The home page for WINE is at <http://www.winehq.com>.
- To emulate DOS programs, a program called DOSemu exists. That program comes with an X Window interface and a console interface. It will run most DOS programs. You may want to run graphic-intensive programs like DOS games in the console interface of DOSemu. Like WINE, DOSemu is a free product.
- To emulate MacOS programs, a commercial program called Executor exists. It will run a MacOS-like shell in X Window and will run various MacOS programs.
- A new commercial program is now available for every OS. It is called VMware and will create a virtual PC, allowing you to run nearly any operating system, including DOS, Windows NT and FreeBSD. It is very stable and comes with a 30 day free license.

Tip 18: Shared library not found

Binaries need to load shared libraries at runtime. They are files on your system that provide functions to the program. Sometimes the program will report that a shared library was not found, and it will abort.

To find which libraries a program needs, you can use the `ldd` program. It will list all those libraries and those missing. If you know that a library is installed, maybe you need to add it to the current library path. The path used to look for shared libraries is in `/etc/ld.so.conf`. You can modify that file and add any path you want, then as root run the `ldconfig` program.

Tip 19: Hard to erase files

File names in Linux can have many letters and numbers in them. Usually, names should not have spaces in them, although Linux can deal with them. There are some characters that should not be used in files, like "/" and "~".

Some programs unfortunately will create strange looking file names, often as temporary files. Using the `rm` command, you should be able to remove them, but it may be hard when strange alphanumeric characters are used. Here are a few ways you should try to get rid of a file with a strange name:

- First you should try the following:

```
rm -f "file name"
```
- If this doesn't work you should try the console program `mc`.
- With graphical file managers, you should be able to pick an icon and remove it, regardless of the file name.

Tip 20: Files permissions

When you try to run a file it may refuse to work with an error like "Permission denied" and when you try to view another file it may also say that you don't have permission to view it. These all come down to file permissions, a basic feature of Unix.

There are 3 types of permissions: read, write and execute. When you list files it will say which permission the files have:

```
ls -l file.dat
-rw-r--r--  1 root      users          1656 Mar 22 00:27 file.dat
```

The first part of that line is the permissions. They are, in order, the user permissions, the group permissions and others permissions, where r means read, w means write and x means execute. For this file, the user, root, has read and write permission (rw-), the group, users, can only read the file (r--) and everyone else can also only read the file (r--).

Other letters may appear. The first letter is - for a normal file, d for a directory and c or b for a device. In place of x you may see a letter s. This means that when you start a program, it will run as its owner.

Tip 21: Changing file permissions

To change a file's permissions, you need to use a program called `chmod`. With that command you can change one or multiple file permissions. Here are a few examples:

```
$ chmod 755 file
$ ls -l file
-rwxr-xr-x  1 root      users      1656 Mar 22 00:27 file
$ chmod 700 file
$ ls -l file
-rwx-----  1 root      users      1656 Mar 22 00:27 file
$ chmod 664 file
$ ls -l file
-rw-rw-r--  1 root      users      1656 Mar 22 00:27 file
```

The numbers are based on the 3 types of permissions. Read = 4, write = 2 and execute = 1. A permission of 755 means the user will have read, write and execute permissions ($4 + 2 + 1 = 7$), and everyone else will have read and execute permissions ($4 + 1 = 5$).

Tip 22: An international background

To set a background in X Window, most of the time you will use XV or a similar program to set a static image. But there is a program out there that allows you to put something useful on your background.

Xearth, available from <http://www.cs.colorado.edu/~tuna/xearth/index.html>, is a program that will display a map of the world on your background, with some useful information. The main use is to display where the sun currently lights the planet. It will also display the current time, and stars in the sky if you want to.

Several options exist. By default, Xearth will show a round planet, but you can make it display a rectangular map:

```
xearth -proj merc
```

Other options are available, including mono display, the number of colors, and the running priority.

Tip 23: Powerful file transfer system

Downloading several directories from an FTP server, or mirroring a web site, can be hard to do using traditional file transfer programs.

A program called wget is available from the GNU archive at <ftp://ftp.gnu.org>. This is a powerful program allowing you to do advanced functions like fetching recursive directories and even retrieving full web sites.

A graphical frontend using the GTK library is available from <http://usuarios.meridian.es/frimost1/gwget/gwget.htm>.

Tip 24: Editing in text editors

One common complaint among new Linux users is that vi or other text editors are too hard to use. That may be true, but they aren't the only available editors.

Experts used to tell new users to learn ed or vi and that they would not regret it. It was true in the past, but it is no longer necessary to learn hard to use editors to do powerful text editing.

Graphical editors most likely came with your distribution. You may not have installed them, but they are on your CD-ROM. Some examples are pico, a console full screen text editor, and emacs, which work on console and X Window, and is a very powerful text editor.

Desktop interfaces such as GNOME and KDE each come with their own text editors too. You can even find more editors on the Web.

Tip 25: Documentation and manual

Software under Linux rarely comes with printed documentation. Some do, like the GIMP and Blender, but most only come with online documentation. This way, you can get the documentation at the same time that you download the program.

There are 2 traditional ways to provide documentation under Unix and Linux:

The first is man pages. These are small files containing information about every command you have on your system. For example, if you want to know what the command `df` does, simply type:

```
man df
```

The man system works with level of commands, from 1 to 8, plus other extensions. For full details about the man program, simply see its manual page:

```
man man
```

The second way to provide help is with the info system. These are usually much bigger files (the libc info files have more than 10,000 lines of text).

To access info files, simply type the info command. It is a bit harder to use, but you can get help by typing `h`.

Networking

What this chapter covers

Linux is a networking operating system. Even so, problems related to networking are not uncommon. PPP dialup can be hard to setup, local network IP address issues can be hard to understand. We try to solve most common problems here with tips on how to make your networking and Internet experience a good one.

Tip 1: Easy PPP dialup

By default, most Linux distributions come with text-based configuration files and console menu-based configuration programs for PPP dialup. Often, this simply isn't enough for most new users.

Several GUI programs exist and can be used to configure your PPP dialup. One is X-isp, available from <http://users.hol.gr/~dbouras/>. You may still need to edit the pap-secrets file in /etc/ppp.

Another is the Gnome PPP dialer. It comes with Gnome and can be used with no text file editing. Be sure to put a "*" in the "Remote address" field if it should be set dynamically.

Here is a quick guide to setup the Gnome PPP program. First make sure the pppd program is suid root. Then, enter your user name, password and "*" in the remote IP field. Enter the DNS server, and set your IP address or set to dynamic. This should work for most Internet providers. Check the PPP-HOWTO for a complete overview of the PPP programs.

This is the Gnome PPP interface:



Tip 2: Internet for your LAN

Many companies have local networks, and it is becoming even more popular in homes, where 2 or 3 systems are linked together using ethernet wires. Most providers only offer 1 IP address per connection, which means that you can't put all of the systems on the Internet at the same time, unless you use a special gateway.

Linux comes with something called IP-Masquerade. With it, you can assign LAN IPs, which are IPs that can't go on the Internet directly, and then use the Linux system as a gateway. All you need to implement this is a Linux system with 2 ethernet cards, and an Internet connection. The Linux system has IP-Masquerade enabled, and the LAN systems will have access to the Internet using the Linux gateway as a transparent proxy server.

You can find more on the subject by reading the IP-Masq HOWTO and other networking documentation at <http://metalab.unc.edu/pub/Linux/docs>. You also need to make sure your Internet provider accepts this kind of network on their connection.

Tip 3: Domains to search in

When you try to access a Web site, or any remote site, you need to specify the full hostname. This means the machine name plus the domain name:

```
lynx my.yahoo.com
```

If you have a few domain names that you access a lot, you can make your life easier. You can edit `/etc/resolv.conf` and add the domains there:

```
search domain.net yahoo.com
```

This means that the system will search in those domains for hostnames. From now on type:

```
lynx my
```

The system will now look for "my", if that hostname doesn't exist it will look for "my.domain.net", and at last "my.yahoo.com".

Tip 4: Display IP rather than hostname

When dealing with networking issues, it often helps to be able to use only IP addresses rather than hostnames. Why? For 2 reasons. First, the name server might not always be available if routing is being changed. And most important, you may not have the time to wait for all the IP resolving to be done.

Fortunately, many networking utilities in Linux share a common option flag. The `-n` flag. It will allow you to make the utility display IP addresses rather than hostnames. Here are a few examples:

```
netstat -an  
traceroute 1.2.3.4 -n  
arp -n -a -i eth0 -a proxy
```

These commands were all given the `-n` flag and will display only IP addresses.

Tip 5: Is my modem a winmodem?

Winmodems are modems which lack some hardware. They use software drivers to emulate the hardware, and the CPU to do some tasks. Unfortunately the drivers provided by the winmodem manufacturers are Windows only.

There is no sure way to know if the modem should work in Linux or not, except trying it yourself, or ask someone else who has tried it. Fortunately, there is a Web site with a very long list of modems that are known to work in Linux, and those that are winmodems. The Web site can be found at <http://www.o2.net/~gromitkc/winmodem.html>.

Tip 6: Sharing files from a Windows system

The protocol to use to be able to share files with Windows systems is called SMB. Linux supports natively a lot of file systems and network protocols. SMB is one of them. To be able to mount a Windows file system to share files, that system needs to have file sharing enabled, and you need to enable SMB support in the kernel. You also need to download a program called Samba which allows you to share remote file systems, and build yourself a server. General information about how to connect Linux, Windows machines and Macs is on the Web at http://www.eats.com/linux_mac_win.html.

If SMB is not a possibility, you could use FTP. Linux by default has an FTP server turned on. Windows FTP servers are also available for free.

Tip 7: Sorry but this host is not in my list

Mail clients require you to specify your server for incoming mail, but they also need a server for outgoing mail. If you have sendmail or Qmail running, you can use localhost. If not, you may use your provider's mail server.

If you setup your own mail server, you may have problems sending mail from other systems on your network. Mail servers often block clients from using them for outgoing mail to prevent relaying. You need to specify the hosts where you will be sending mail from. Here is how to do it in Qmail, from the FAQ. You need to put the following line in /etc/hosts.allow:

```
tcp-env: ip1, ip2, ip3: setenv = RELAYCLIENT
```

Tip 8: Access to various networks

Big corporations often sub-divide their networks into small networks, hidden behind gateways. To access them, you need to tell your Linux system that there is a gateway to use to access the networks.

The `route` program makes it easy to add networks, hosts and gateways to your routing table. To add a default gateway, for example to access the Internet, you can set it as default with the following line:

```
route add default gw 10.0.0.1
```

This will work if you need to access the Internet via the 10.0.0.1 gateway. Now, if you want to access networks 10.1.0.0 and 10.2.0.0 through other gateways, here is what you will want to do:

```
route add -net 10.1.0.0 gw 10.0.0.10  
route add -net 10.2.0.0 gw 10.0.0.20
```

Tip 9: Accessing remote file systems

SMB is the most popular protocol to access Windows systems. But from the Unix world comes NFS. NFS is a way to share files that predates SMB and Samba, and comes compiled in most Linux distributions. To enable file sharing, you must have the `nfsd` and `mountd` daemons running. You also need to add the IPs of the systems you want to allow in `/etc/exports`.

To access remote file systems, you simply mount them like local hard drives. To mount `/usr/files` from `1.2.3.4` into `/mnt/files`, simply type:

```
mount -tnfs 1.2.3.4:/usr/files /mnt/files
```

The `-tnfs` parameter may be omitted.

Tip 10: Secure Web server

Electronic commerce is becoming very popular on the Internet. Companies will often pay thousands of dollars for commercial packages to deliver secure content to customers on the Web. You can setup one of the most popular Web servers, Apache, running on Linux and serving secure content, for free.

To setup Apache to deliver secure content, you will need to get a cryptographic package called OpenSSL, based on the SSLeay library. The place to start is at <http://www.apache-ssl.org>. From there, you can download the needed patches to make Apache into a secure web server.

Detailed instructions are available in the packages, but here is a quick step-by-step guide:

- First, you need to download 3 packages: Apache itself, the corresponding Apache-SSL patch and OpenSSL.
- Then you need to patch the Apache distribution and compile the SSL library.
- After editing the configuration file in the Apache directory, and setting the right paths and libraries to use, you can compile Apache and then create a test certificate.
- All you have to do now is install Apache and configure it to use your test certificate.

Note that while Apache and the SSLeay library are free, you may need to pay to get signed certificates from commercial companies. Also, due to export laws in various countries, you may want to check your local laws before using any encrypting product.

Tip 11: Secure alternative to telnet

Telnet is a protocol allowing you to connect to a remote system and run programs and commands on that system. It is very old and still very much in use today.

Unfortunately, a telnet client sends the user password as clear text, and the connection is not encrypted. On the other hand, a program called ssh exists that can replace both telnet and ftp in a secure, encrypted way.

Ssh stands for Secure Shell. It will encrypt each connection with a random key, so that it is impossible or at least very hard for a third party to decrypt the connection and find the password, or spy on you.

Tip 12: Speed problems on a PPP connection

PPPd is the PPP connection daemon. It will try to connect to a server using a specified speed. The default speed is 38400. If you use a serial connection, or a 56.7Kbps modem, it may not be enough. If you want to use all the available bandwidth, you need to increase that number. For example, for a serial connection, you want the speed set at 115200.

Another reason for speed drops is unwanted packets. You may want to filter unwanted packets out of your network, like some ICMP messages and chat connections.

A last possibility for speed drops is Denial of Service attacks. DoS attacks are unfortunately very real and they occur a lot. Malicious people that can't handle their problems elsewhere turn to the Internet and launch attacks against networks. An attack against one user will always affect several thousands of people. By using bandwidth of an Internet provider to cause trouble to any one user, the whole provider will be affected. To prevent such attacks, firewalls exist, and tracking tools were invented to deal with abusers. MCI has a tool called DoSTrack that can be of help if you are victim of such an attack. For more information about various DoS attacks, you should search the Web.

Tip 13: Names and name servers

Internet hostnames and domains are resolved using the Domain Name System (DNS) using Name Servers (NS). These name servers are usually hosted by your Internet provider. You can also host your own name server, using the program named. Every name server, upon receiving a request to resolve a hostname, will ask an upstream name server if it doesn't know the answer. Your name server may ask your ISP's name server, which will ask the backbone's main name server, which will ask a root server.

Linux knows which name server to ask by looking in `/etc/resolv.conf`. In that file, a number of name servers may be specified in the following way:

```
nameserver 192.168.0.1
nameserver 205.237.65.254
```

The name server itself, named, has a configuration file which is usually `/etc/named.conf`. In that file, you configure the domain names you are responsible for, and the zone file to use. A nice introduction to running a name server is available in the various named man pages.

Various utilities are related to resolving hostnames. One is called `whois`, and will query the Internet main name servers to know who owns a domain:

```
whois linux.org
```

Another utility is called `nslookup`. That command will allow you to resolve hosts, and to get all kinds of information about a domain. See the man page for more.

Tip 14: Who owns this port

Several utilities exist to check which ports are open, who is connected to your system and even what process owns a port number.

First a few ground rules. Ports below 1024 are reserved for common services, and only root can use them. Standard port numbers can be found in `/etc/services`. The maximum number of ports is 65k, so you have more than enough Internet ports for all your services.

Here are some useful utilities. Netstat is a command that will list both the open ports and who is connected to your system. You should run it like this:

```
netstat -an | more
```

This way you can find out who is connected to which service.

Another interesting command is the `fuser` program. This program can tell you which user and process owns a port. For example, the following command will tell you who owns port 6000:

```
fuser -v -n tcp 6000
```

Tip 15: Network printers

By tradition in Unix most services come with networking capabilities. This includes the printing server. You don't need to get third party software to make a printing server.

The lpd daemon allows you to print to your local printer, but also allows others to print on it, if you allow them.

By default the printing software will read on port 515 on the UDP protocol. It will allow hosts listed in the `/etc/hosts.lpd` to print using your printer.

For a full overview of the printing service, you should check the printing howto on the Web.

Development

What this chapter covers

This last chapter covers development. A standard Linux system comes with many compilers and interpreters for a lot of languages. We cover most of them, and will see tips and tricks on how to use them for small scripts or powerful applications.

Tip 1: Graphical messages to the world

Often we write shell scripts and at one point we would like to display messages or otherwise interact with the user using a graphical way, without rewriting all our script in C.

A small program called gtk-shell was created for that purpose. It is a GTK-based program meant to be used in shell scripts, that can interact with the user, and even return values to the script. Let's see a very simple example:

```
xv -quit -root `gtk-shell -t "Which file do you want me to use?" -fs`
```

This will display a graphical browsing box with the specified title, where the user can select a file, and the file path will be returned to xv which will display it as background.

Gtk-shell can be used in many ways including displaying a message, asking for input, and more. Here is an output of "gtk-shell --help":

```
gtk-shell (C) 1999 Patrick Lambert under GPL
This program will use the GTK library (www.gtk.org) to ask for user input,
and can be used in scripts or called from other programs.
```

Command line options:

--help, -h	This help text
--version, -v	Show gtk-shell's version
--output <file>, -o	Set the output file, default is stdout
--title <title>, -t	Set the window title
--label <label>, -l	Show a label to the user
--file-selection, -fs	Display a file selection box
--size <x> <y>, -s	Set the size of the window
--position <x> <y>, -p	Set the position of the window on the screen
--query, -q	Display a query box
--query-value, -qv	Set a value in the query box
--append-eol, -eol	Appends an EOL char at output time
--exit-code <str>, -ec	String to display if user press Cancel
--view-file <file>, -vf	View a file in a text box
--edit-file <file>, -ef	Edit a file in a text box
--button <str>, -b	Add a button to quit
--choice <c1> <c2>..., -c	Display a choice box with the specified choices

```
Example: gtk-shell -l "Pick a choice" -c 1 2 3 4 -b Ok -eol
```

```
Example: xv -quit -root `gtk-shell -t "Which image?" -fs`
```

Gtk-shell is available at <http://devplanet.fastethernet.net/files.html>.

Tip 2: Code reuse

Why write code that we already wrote for previous programs? Well here is a small tip that might save you some time: Make a generic file with reusable functions.

Many languages make this easy to do. Object oriented languages like Java or C++ for example allow you to make a class with utility functions, and then import them in your main program. This way you can use the generic class in all your programs. Even in C you can do this by including the generic C file when compiling with a command like:

```
gcc -o myprogram myprogram.c generic.c
```

Your generic utility functions would be in generic.c and gcc would simply add the file to myprogram.

Tip 3: Makefile don't equal C

Makefiles are used in most Unix C or even C++ programs. But nowhere does it say that they can't be used for other languages. Make is a program installed with most Linux distributions, and on most Unix systems too. Makefiles make your program portable, and easy to compile. These files can be used in C, C++, Java, and any program that requires compilation.

Make has so many useful applications. Feel free to explore them for your programs.

Tip 4: Parsing the command line in BASH

Bash is a shell, but it's also a scripting language. You can make bash scripts, and you can pass parameters to it. You can access the parameters with \$1, \$2, ... But what if you need the whole command line?

Bash uses special variables for all kind of purposes. One of them is \$* which contains the whole command line. You can use it rather than the \$1, \$2 if all you need is a string of text, passed on the command line.

Tip 5: Don't grep grep

A useful tool in scripting is the "grep" function. This program will find a string in a file. It is often used also to find if a process is running:

```
ps ax | grep sendmail
```

That command will find if sendmail is running. The problem is that you might end up with an other entry for this command. Because you grep for "sendmail", you may well end up with this command showing because the "sendmail" string is in the command line. To avoid that, you can use the -v flag to grep:

```
ps ax | grep sendmail | grep -v grep
```

That command will find all the lines in the current process list that have the "sendmail" string in it, and will remove the lines containing the string "grep".

Tip 6: Move a text into upper case letters

When you want to do text manipulation, you can use Sed and Awk. These 2 tools which come on most Linux distributions, will allow you to modify text files in many ways.

To move a text file into upper case letters, you can use Awk in the following way:

```
awk '{ print toupper($0) }' old_file > new_file
```

Sed and Awk are useful for a lot of other uses, and are integrated in several products.

Tip 7: Using PASCAL on Linux

Linux comes with a lot of compilers and interpreters. These include programs for C, C++, Perl, TCL/TK and more. Unfortunately most Linux distributions don't come with a Pascal compiler. Is it possible to compile Pascal programs? It sure is.

Several projects were started to make a Pascal compiler for Linux. One is called GNU Pascal and is available from <http://agnes.dida.physik.uni-essen.de/~gnu-pascal>. That program will also run on any operating system that supports the GNU C compiler.

Tip 8: Segmentation fault

One of the most common problems when making software is errors like "Segmentation fault", also called SegFault. Here is what a SegFault is.

Virtual memory in a computer can be created in 2 ways: pages or segments. Paging means that the memory is divided in pages of equal size, containing words of memory in it. Segmentation means that every process has a segment of memory of needed size, with gaps of empty memory blocks between the segments.

The operating system knows the upper limit of every segment, and every segment begins at a virtual address 0. When a program accesses a memory block, it calls a virtual address that the Memory Management Unit (MMU) maps to a real address. If the operating system sees that the requested address doesn't match any valid address in the segment, it will send a signal to the process terminating it.

SegFaults are the direct result of a memory error. The program has a bad pointer, a memory leak or any kind of error that makes it access the wrong memory address. To correct these errors you need to check pointers and arrays for errors.

Tip 9: Who is online?

Many system administrators use scripts to help them in the process of managing a server. A common problem is finding out exactly what users are on the system and what they are doing.

Several tools are available on the system to see who is online, what processes are running, and pipeing them together can resolve many problems. Here are 2 small scripts that will show, first if a user is online, and then what he is running:

```
who | grep $1  
ps -aux | grep $1
```

The variable \$1 here means the first command line argument, from a shell script. The who command first checks if the user is online, then ps will show all the processes currently running under that user's UID.

Tip 10: Graphical toolkits

Most Unix programs are console based. This is mainly because several years ago, most computer users were on text-only stations linked to big mainframes.

Now, Linux is becoming a full graphical desktop, and needs graphical programs, or at least frontends to text-only programs. This is where graphical toolkits come in. Here is a list of some of the toolkits available to Linux programmers:

- [GTK](#) - This is currently a very popular toolkit, and is the base of the GNOME desktop. It is C based and easy to work with.
- [QT](#) - QT is a C++ toolkit made by Troll Tech. It is the base of the popular KDE desktop.
- [WXwindows](#) - This is a very complete C++ toolkit that is cross-platform. The same code should work on Windows, the MacOS, Unix (Motif and GTK) and others.
- [GraphApp](#) - This is also a cross-platform toolkit, but is one of the easiest libraries I ever saw. It supports a smaller set of widgets, but takes only hours to learn. It will work on Windows, the MacOS, Unix (Motif and Athena) and others.

Tip 11: IDE and visual interfaces

If you are a programmer that comes from the Windows world, you may be used to code in a visual interface or an IDE. Coding in a text editor and then going to the shell to compile may be the traditional way of Unix and Linux users, but it may not be what you are looking for.

You can use a visual interface in Linux too. Searching for "visual" or "IDE" at <http://www.freshmeat.net> will return a few programs available to you. But there may be a much better way.

The purpose of a visual interface is to make coding more efficient. There are many so-called programmer editors. These are text editors so powerful that you can do everything with them. This includes editing multiple files, compiling and even debugging your program.

One of them is emacs, available from the GNU project. It is so powerful that it enables you to edit text, compile, read mail, newsgroups, and even browse the Web. It may seem far more complex than your usual Windows visual editor, but it may prove really interesting in the end.

Tip 12: Free software and copyleft

Most software programs under Linux come under the GPL license. The so-called copyleft agreement. It may be confusing to new Linux programmers. Here are the main features of the license, and why you may or may not want to use it for your programs. The full license is available from the GNU site at <http://www.gnu.org>.

The GPL license provides software free. Not as in free of charge, but in freedom. You can charge money for free software or software under the GPL. What you can't do is restrict its use.

You must provide source code, or a way for the user to have access to that source code. This means that if you release a program as a binary only, it can't be under the GPL license and may not contain GPL code from other people.

If you use code available from other people under the GPL license, you can modify it and redistribute it under the same license, with source code. Also, you must keep the previous copyright notices, and write down in the program what changes you have made to it. This also means that anyone can modify your programs, as long as they keep your copyright notices there, and release their modifications under the same license.

Tip 13: Talking to the terminal

Under DOS, or other text based systems, there is only one way in and one way out. Under Linux and every Unix system, there is one way in but two ways out.

The terminal has 2 outputs, one for standard output and one for errors. By default, they both display on the same device: your monitor. But they can be redirected, for example, to different logs. This is why it's important for you to use them properly.

The standard output is called stdout, and printing to it will print to the screen, or where it is redirected. In C, it may look like this:

```
fprintf(stdout, "Test\n");
```

The standard error is called stderr:

```
fprintf(stderr, "Error\n");
```

To input, you need to read from the standard input, called stdin. They are all available to you from the stdio.h file.

Tip 14: Internet technologies

The Internet is of course a big phenomena, and Linux is a big part of it. Linux servers are very popular, and you can be a part of this revolution.

Among available technologies, here are the most popular ones:

- Java - The Java language is a new and powerful language made to be cross-platform. Since the Web must be available on all systems, this makes Java a very powerful tool. You can make Java applets that will run on the user's system and display information or interact with him.
- JavaScript - This is similar to Java, but in a Web page itself. It is embedded in HTML, to do small functions like rollover scripts, and making text appear on the screen.
- CGI - The CGI protocol is very much in use for servers with some sort of account system, like online shops and search engines. You can make CGI scripts in any language, and make your Web server run them on the server.

Tip 15: Library types

Two types of libraries exist on most operating systems: shared and static. On Windows, they are .DLL, for dynamically linked library, and .LIB for static library.

On Linux and most Unix, they are .so and .a files. The shared libraries, the .so files, are loaded at runtime. The .a files, or static libraries, are loaded at compile time and are no longer required to run the binary program.

When you make a program, you must decide if you will link it to a static library or a shared one. You will want a shared library in most cases because standard libraries are available on most systems, and would be too big to include in a binary file.

If you have a small library that is not one of the standard ones that you need, then you may decide to include it in your binary program. In that case, simply add it like any other object file in your compilation:

```
cc -o program file1.o file2.o library.a
```