



Optimizing Your System

Ubuntu should prove to be as responsive in day-to-day operation as Windows, if not more so. But if you run into any performance issues, or if you simply want to get the most out of your system, then this chapter is for you. The chapter doesn't cover essential knowledge, so you can skip it if you're satisfied with how your system runs. More often than not it discusses hacks—clever methods of making things work in a nonstandard fashion. But as your experience of Ubuntu might have already taught you, such hacks are the lifeblood of Linux. One of the strengths of Linux is the ability to delve under the hood and change absolutely any aspect of the way it works.

Speeding Up Booting

Since Ubuntu 6.10 (Edgy Eft), Ubuntu has been using a boot routine called Upstart that effectively optimizes itself. You can learn more about Upstart at <http://upstart.ubuntu.com>. However, you can still make a performance tweak by enabling startup scripts to run in parallel, instead of one after the other. This works well for multithreaded, multicore processors, like Intel's Core 2 Duo and AMD's Athlon 64 X2. It won't make the computer start instantly, but it will save a few seconds each time you boot.

To configure running startup scripts in parallel, you need to edit the `/etc/init.d/rc` file. You can load this file into the Gedit text editor by typing the following in a terminal window:

```
gksu gedit /etc/init.d/rc
```

Look for the line that begins with `CONCURRENCY` near the top of the file, as shown in Figure 30-1, and change the value from `none` to `shell`. Save the file and reboot the computer. If you see no improvement in boot speed, you can change this value back to `CONCURRENCY=none` if you wish.

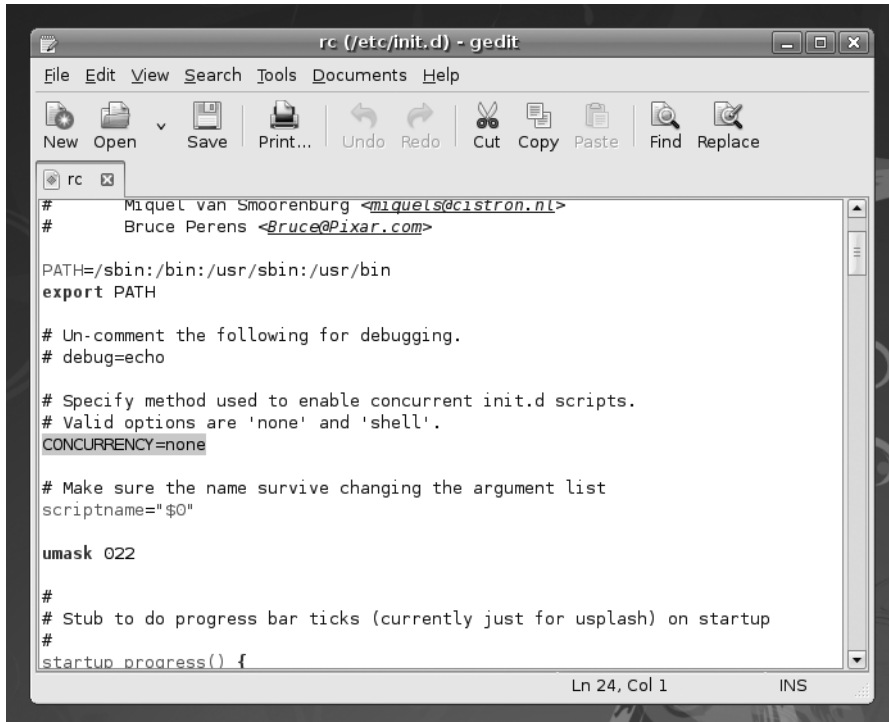


Figure 30-1. Parallel execution of scripts in the boot process can be enabled by editing the `/etc/init.d/rc` file.

Reducing the Boot Menu Delay

Getting rid of the GRUB boot menu delay can save some waiting around in the early stages of the boot process. The delay can be reduced to one second, or even eradicated completely. Of course, in such a case, you won't be able to choose which operating system you want to load if you're dual-booting with Windows. Even if Ubuntu is the only operating system on your computer, without the boot menu delay, you won't have the chance to boot into recovery mode or a previously installed Linux kernel from the GRUB menu. So you need to consider whether this is a worthwhile time-saving measure.

The boot menu delay is stated in the `/boot/grub/menu.lst` file. You can load this into the Gedit text editor by typing the following in a terminal window:

```
gksu gedit /boot/grub/menu.lst
```

Look for the line that begins with `timeout`, as shown in Figure 30-2, and change the value to whatever you wish. The units are counted in seconds, so a value of 3 equates to three seconds. A value of 0 (zero) will mean the boot menu won't appear at all, which is not recommended, for the reasons just mentioned. Generally speaking, a delay of one second (1) gives you just enough time to press a key at the appropriate time. This will

cancel the countdown, meaning the boot menu will stay on your screen until you select an option.

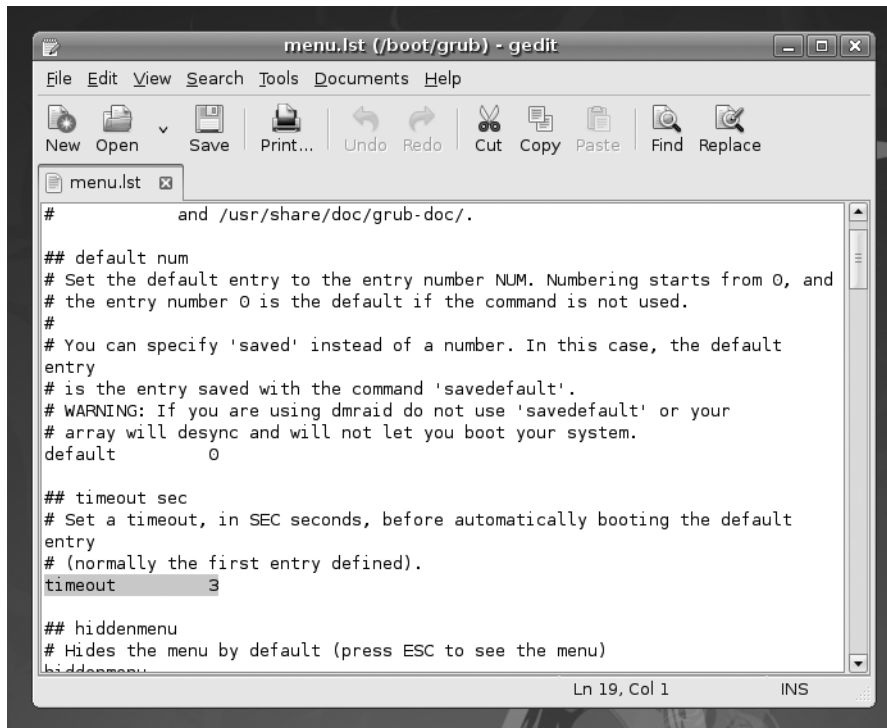


Figure 30-2. You can stop the GRUB menu from hanging around for so long by changing the timeout value in its configuration file.

When you've finished, save the file, and quit Gedit.

Enabling Automatic Login

The first component of user interaction Ubuntu requires is the input of a username and password. By default, Ubuntu will pause at the login screen indefinitely until the details are entered. Following this, the GNOME desktop will load, which can take another 30 seconds on an older computer.

It's possible to make Ubuntu log in to any particular account automatically, which will save time during boot (and will also allow you to activate your computer, leave the room, and come back to the computer ready for instant use).

However, you should do this only if you're confident your computer is in a secure location. After all, enabling automatic login will mean that anyone who has physical access to your computer will be logged in automatically and have access to your data.

Note Once automatic login is enabled, you will still be able to log out by clicking System ► Quit and clicking the Log Out button. You will see the login screen, which you can use to switch to other user accounts.

To enable automatic login, click System ► Administration ► Login Window. In the Login Window Preferences window that appears, click the Security tab, and put a check alongside Enable Automatic Login. Then choose the account that you would like to log in automatically from the User drop-down list, as shown in Figure 30-3. (If there's only one user on your system, there will be only one entry in this list.) Click the Close button at the bottom right of the Login Window Preferences window to save your changes and exit.

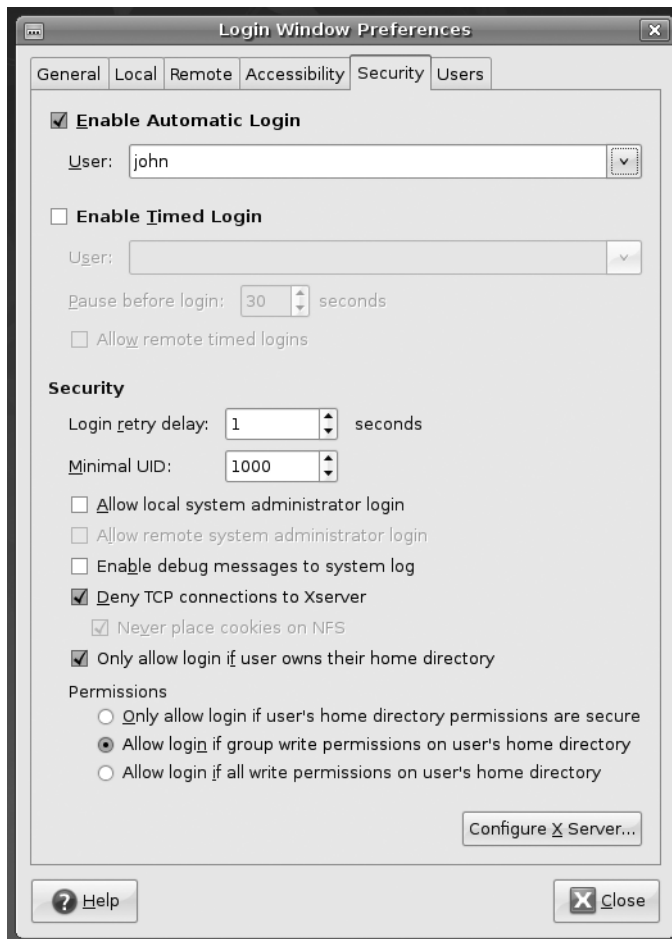


Figure 30-3. Automatic login can save precious seconds when booting, but represents a security risk.

Managing GNOME Sessions

Under Windows, you might be used to controlling which programs start up at the same time as the desktop, by adding or deleting entries in the Start menu's Startup program group. When running Ubuntu, you can control which system applets are started automatically, as well as your personal choice of desktop applications.

To control which programs start up with the GNOME desktop, use the Sessions program (`gnome-session-manager`). To run the program, click System ► Preferences ► Sessions.

Clicking the Startup Programs tab shows the programs that start when GNOME does, as shown in Figure 30-4. You can disable a startup program in the list by unchecking the Enabled check box beside it.

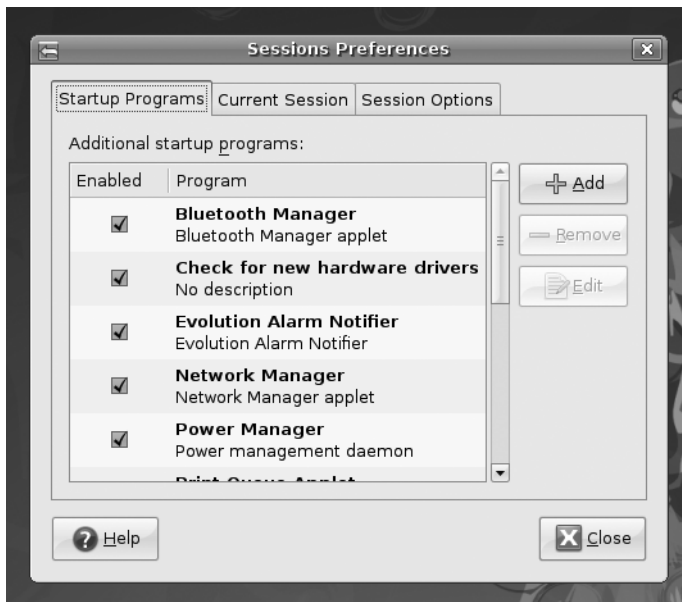


Figure 30-4. In the Sessions Preferences window, you can control which programs start when GNOME does.

On our test system, the following 12 entries were in the list; your computer might have more entries.

Bluetooth Manager: This program is responsible for the Bluetooth applet that can be used to send and receive files, browse other Bluetooth devices, set up Bluetooth services, and manage Bluetooth service settings. You can disable this applet if your system is not Bluetooth-enabled, or if you just don't want to use Bluetooth at all.

Check for new hardware drivers: This tool scans your hardware devices and checks if new third-party proprietary drivers can be used with them. If a new driver is compatible, this tool provides a facility to download, install, and configure the driver in your computer. If your hardware devices are working perfectly without third-party drivers, or you're uncomfortable using proprietary software, it is safe to disable this program. Otherwise, keep the tool enabled so that you can receive driver bug fixes and updates.

Evolution Alarm Notifier: As its name suggests, this utility ties into Evolution's calendar function in order to notify you of events, such as an appointment that you don't want to forget. If you don't use Evolution, or don't use its calendar function, this applet can be disabled.

Network Manager: This applet manages your Wi-Fi, Ethernet, and VPN connections. This is useful for laptop users who connect to several Wi-Fi networks on the go, as discussed in Chapter 8. You can disable this program if your network settings are static and always use the same IP address, DNS, and gateway settings when connecting to a particular local area network.

Power Manager: This program controls all aspects of GNOME's power management, including the useful hibernate feature that can save the contents of the system's RAM to make for quicker startup. If your computer is a notebook, this tool should be considered essential. If you have a desktop PC and are looking to save power using hibernate, suspend, or screen blanking after a period of inactivity, you won't want to disable Power Manager. Disabling it will remove the Suspend and Hibernate options from the System ► Quit dialog box.

Print Queue Applet: This applet provides an interface for you to cancel or repeat printer jobs. It is also responsible for automatically setting up a printer for use when you plug in a printer. You can disable this applet if you will never print from your computer.

Tracker and Tracker Applet: Tracker is the background process that indexes the contents and metadata of your home directory for fast searching. Tracker Applet is the tool for searching your data and customizing Tracker settings. By default, indexing is disabled unless you enable the indexing options in the Tracker Preferences window (System ► Preferences ► Search and Indexing). You should leave indexing disabled if using it slows down your computer; however, the benefits of obtaining fast results when searching your data are undeniable.

Note The Tracker Applet is still undergoing testing, and it is not known whether it will remain part of future Ubuntu releases.

Update Notifier: This is the Update Manager tool. You shouldn't disable this applet, because it performs the essential task of checking if any system updates are available. It runs in the background once started, and hardly impacts startup time at all. However, if you absolutely must prune valuable milliseconds from startup, you can disable it. You can then check for updates manually, whenever you desire, by clicking System ► Administration ► Update Manager.

User folders update: This tool pop ups during the start of your session if you have recently changed the default language (System ► Administration ► Language Support) of your desktop. The tool gives you an option to translate the folder names Desktop, Templates, Public, Documents, Music, Pictures, and Videos in your home directory to the new default language. You can disable this tool if you don't change your language settings.

Visual Assistance: This utility runs assistive technologies such as the Orca screen reader, magnifier, and Braille application if these tools were enabled in the Assistive Technologies Preferences window (System ► Preferences ► Assistive Technologies). See Chapter 10 for a discussion of these tools. If you are not using assistive technologies, you can disable this applet.

Volume Manager: This program isn't, as you might think, related to the sound system. Instead it allows the automounting of removable storage (known in Linux as volumes), such as digital cameras and CD or DVD-ROM discs. This is a critical function of Ubuntu and, without it running, you face the uphill struggle of manually mounting removable storage devices at the command prompt. You can disable it if you want, but it's definitely not advisable.

The Current Session tab of the Sessions Preferences window provides a list of programs that are running at that exact moment. You can use this tab to test the effect of removing applications from the startup list, by highlighting the application, clicking the Remove button, and then clicking the Apply button. This will force the program to quit, so make sure you save any open files before closing an application this way, to avoid loss of data.

The Session Options tab provides two ways to set your preferred applications to run automatically when you log in, instead of starting each application that you need manually. If you choose the Automatically Remember Running Applications When Logging Out option, applications that are running as you log out will be started automatically when you log in next time. If you choose the Remember Currently Running Applications option, applications that were running just before you clicked this option will be started automatically when you log back in.

Tip The Startup Programs tab of the Sessions Preferences window contains an Add button, which lets you add any program you like to the GNOME startup. You could add Evolution, for example, so that it starts automatically whenever you log in. But it's easier to add currently running applications using the option on the Session Options tab.

STOP WAITING FOR AN ADDRESS

If you use an Ethernet or Wi-Fi connection to access your network, you might find that Ubuntu spends a few seconds during each boot acquiring an Internet address. This will be characterized by a long pause while nothing seems to be happening. Therefore, one way to provide an instant speed boost is to give your computer a static IP address. Chapter 8 explains how to configure your network interface.

However, to assign a static address, you'll need to find out what IP address range your router (or other DHCP server) uses. This can be discovered by looking at the router's configuration software. Sometimes, this is accessed via a web browser. Look for the section of the web interface headed DHCP Configuration or similar.

Normally, IPv4 local area network addresses are in the 192.168.x.x range, where x.x can be any series of numbers from 1.1 to 255.255. For instance, you may find that your router uses the 192.168.1.2-255 range. In this case, assigning a static IP address that will work with the router is simply a matter of choosing an IP address in this range. However, this router may hand out addresses sequentially from 2 upward, so it's best to choose an address it's unlikely to reach, even if you happen to have many computers connected to the network. Starting at 50 is a good idea, so you could assign the address 192.168.1.50.

Don't forget that when defining static IP addresses, you'll need to manually supply the gateway, subnet, and DNS addresses. In the example, the gateway would be 192.168.1.1 (the address of the LAN interface on the router), and the subnet would be 255.255.255.0. The DNS address on a small home network will probably be the same as the gateway address, because the router will usually be set to forward DNS requests by default. This isn't always the case, though, so be sure to check.

BUILD YOUR OWN READAHEAD PROFILE

Ubuntu includes a feature called *readahead*, which is able to order the list of files to be loaded during bootup by their locations on the hard disk. A default *readahead* list is installed on a standard Ubuntu installation. This is created on a generic PC, but you can build your own version of the list, customized for your own computer.

Here are the steps to create your own *readahead* list:

1. Reboot Ubuntu, and at the boot menu, highlight the Ubuntu entry and press E. If Ubuntu is the only operating system on your computer (that is, your computer doesn't dual-boot with Windows), you might need to press Esc to see the boot menu when prompted.
2. Highlight the second line, beginning with the word `kernel`, and press E again.
3. Using the right-arrow key, move the cursor to the end of the line. Insert a space, type `profile`, and press Enter. The following is how the entire line read on our test PC; yours may be slightly different (note that the beginning of the line was cropped off because of the resolution of the screen):

```
< quiet splash profile
```

4. Press B to boot the computer. This boot will take longer than usual, because the boot profile is being rebuilt. When the computer has booted up, and all disk activity has stopped (which might take a minute or two after the desktop has appeared), reboot your computer. You should find that bootup is faster.

Prelinking

A lot of Ubuntu software relies on other pieces of code to work. These are sometimes referred to as *libraries*, which is a good indicator of their purpose: to provide functions that programs can check in and out whenever they need them, as if they were borrowing books from a library.

Whenever a program starts, it must look for these other libraries and load them into memory, so they're ready for use. This can take some time, particularly with larger and more complicated programs. Because of this, the concept of *prelinking* was invented. By a series of complicated tricks, the `prelink` program makes each bit of software you might run aware of the libraries it needs, so that memory can be better allocated.

Prelinking claims to boost program startup times by up to 50% or more, but the problem is that it's a *hack*—a programming trick designed to make your system work in a nonstandard way. Because of this, some programs are incompatible with prelinking. In fact, some might simply refuse to work unless prelinking is deactivated. At the time of this writing, such programs are in the minority. However, keep in mind that prelinking can be easily reversed if necessary. Alternatively, you might want to weigh whether it's actually worth setting up prelinking in the first place.

Configuring Prelinking

If you decide to go ahead with prelinking, you'll need to download the relevant software from the Ubuntu software repositories. Open the Synaptic Package Manager (System ► Administration ► Synaptic Package Manager), click the Search button, and type `prelink` into the search box. Mark `prelink` for installation, and then click Apply.

Before you can run a prelinking sweep of your system, you need to enable it in one of its configuration files. To do this, type the following in a terminal window:

```
gksu gedit /etc/default/prelink
```

Change the line that reads `PRELINKING=unknown` to `PRELINKING=yes`. Then save the file and quit Gedit.

To run a prelinking scan of your system, simply issue this command:

```
sudo prelink -a
```

This will prelink practically all the binary files on your system, and may take some time to complete. You may also see some error output, but you don't need to pay attention to it.

Prelinking was automatically added as a daily cron job when you installed it (see Chapter 32 for a description of cron), so any new programs you add will be automatically prelinked.

Deactivating Prelinking

Should you find prelinking makes a particular application malfunction or simply stop working, you can try undoing prelinking. To do this, find out where the main binary for the program resides, and issue the `prelink` command with the `--undo` command option. For example, to remove prelinking from the Gedit text editor program, you could type the following:

```
whereis gedit
```

This command will show that the `gedit` binary is found at the location `/usr/bin/gedit` in the file system. Next, attempt to undo prelinking on the binary:

```
sudo prelink --undo /usr/bin/gedit
```

However, this may not work, because some programs might rely on additional binaries on the system. Therefore, the solution might be to undo prelinking for the entire system, which you can do by typing the following:

```
sudo prelink -ua
```

After this, you should remove the `prelink` package, via the Synaptic Package Manager, to stop it from running again in the future (or manually remove its cron entry, as explained in Chapter 32).

OPTIMIZING THE KERNEL

You can download the Linux kernel source code and compile your own version of it. This gives you total control over the kernel configuration, so you can leave out parts you don't want in order to free memory. You can also set certain optimization settings, such as creating a version of the kernel specifically built for your model of CPU.

Although compiling a kernel is a simple procedure, there are many complex questions that you'll need to answer, and an in-depth knowledge of the way Linux works is necessary.

In addition, compiling your own kernel brings with it several issues. The first is that it may not work with any binary modules that you have installed, such as graphics cards or wireless drivers. You can opt to update these yourself, but this adds to the complexity.

The second problem is that Ubuntu is built around precompiled kernels. Several software packages expect to work with the precompiled kernel, and in addition, Ubuntu may occasionally download an updated prepackaged kernel automatically as part of the system update feature and override the one you've created.

If there are any security problems with the kernel version you compiled, you'll need to recompile a new kernel from scratch (or patch the one you have). This means you'll have to keep an eye on the security news sites and take action when necessary.

That said, compiling a kernel is an excellent way of learning how Linux works, and if it all goes well, the sense of achievement is enormous.

Some people choose to download the kernel source code from the official Linux kernel site, www.kernel.org. However, it makes more sense to download the official Ubuntu release, because this will be tailored for the way your system works. Using the Synaptic Package Manager, simply search for `linux-source`.

You can find several guides to compiling your own kernel online, but we recommend the following posting on the Ubuntu forums web site, which looks at compiling a kernel under Ubuntu: <https://help.ubuntu.com/community/Kernel/Compile>.

Freeing Up Disk Space

After using Ubuntu for some time, you might find that the disk begins to get full. You can keep an eye on disk usage by using the following command in a terminal window:

```
df -h
```

This will show the free space in terms of megabytes or gigabytes for each partition, also expressed as a percentage figure. If the disk does start to get full, you can take steps to make more space available.

Emptying the /tmp Folder

An easy way to regain disk space is to empty the /tmp folder. Like its counterpart in the Windows operating system, this is the folder in which temporary data is stored. Some applications clean up after themselves, but others don't, leaving behind many megabytes of detritus.

Because the /tmp folder is accessed practically every second the system is up and running, to empty it safely, it's necessary to switch to run level 1. This ensures few other programs are running, and avoids the risk of deleting data that is in use. First, switch to the text console by pressing Ctrl+Alt+F1. Then enter these commands to switch to run level 1:

```
sudo killall gdm
sudo init 1
```

A recovery menu will appear. Select the Drop to Root Shell Prompt option. Then enter the following to empty the /tmp folder and reboot:

```
rm -rf /tmp/*
reboot
```

Tip On a similar theme, don't forget to empty the desktop Trash. This can hold many megabytes of old data. If you see an error message about permissions when emptying the Trash, you can do so manually from a terminal window. Simply type `sudo rm -rf ~/.Trash/*` to get the job done.

Emptying the Cache of Package Files

You might also choose to clear out the Advanced Packaging Tool (APT) cache of old .deb package files. On a system that has been very frequently updated, this can free many megabytes (possibly gigabytes) of space.

You can empty the cache by typing the following command in a terminal window:

```
sudo rm -f /var/cache/apt/archives/*.deb
```

Caution Be careful to type the command to empty the APT cache exactly as it's written. Even inserting an additional space can lead to very bad consequences!

However, the `apt-get clean` and `apt-get autoclean` commands are considered a much safer way to remove unwanted package files.

If you want to restore any packages later on, simply locate them in the Synaptic Package Manager list, click the check box, and click Mark for Reinstallation. This will cause the package to be downloaded, installed, and configured.

Removing Unused Software

If you still need disk space, consider uninstalling unused programs. As you've learned, you can manage software through the Synaptic Package Manager (System ► Administration ► Synaptic Package Manager).

To remove a package, click its check box and select Mark for Removal. However, it's not a good idea to simply scroll down the list and remove anything that seems dispensable. Because of the way Linux works, many seemingly insignificant packages are actually vital to the running of the system. Instead, it's a better idea to look for programs on the Applications menu, and then return to the Synaptic Package Manager to remove them by searching for their names.

As always, removing software can create dependency problems, so you might find yourself limited in what software you can actually remove.

Tip If you want to remove all the desktop games, simply search for `gnome-games` and `gnome-games-data` in the Synaptic Package Manager, and mark them for removal.

Summary

In this chapter, we looked at streamlining your installation of Ubuntu. This involved speeding up the boot procedure by running boot scripts in parallel, and decreasing the boot menu delay, along with a handful of other tricks. We also looked at optimizing your hard disk settings to allow for greater speed when loading and saving files.

Additionally, we investigated prelinking programs so that they load faster, recompiling the kernel so that it's optimized for your system, and freeing disk space by various means.

In the next chapter, you'll learn how to perform backups to safeguard your data.