Slackware Linux Basics

For Slackware Linux 9.1

Daniël de Kok

Slackware Linux Basics: For Slackware Linux 9.1

by Daniël de Kok

Table of Contents

Chapter 1. About this book

About this book

This book aims to provide an introduction to Slackware Linux. It addresses people who have little or no GNU/Linux experience. It aims to cover the Slackware Linux installation, basic GNU/Linux commands and the configuration of Slackware Linux. As you can see the book is still work in progress, but the first bits are released in the “release early, release often” spirit.

This book is written by Daniël de Kok and is freely available under the terms of the GNU Free Documentation License. It is continually under development, not just to keep up with the latest Slackware versions, but also to refine the documentation, and extend it where it is deemed necessary. The latest version can be found at http://www.slackfiles.net/

We wish everybody a good time with Slackware Linux, and we hope this book is useful for you.

Chapter 2. An introduction to Slackware Linux

What is Linux?

Linux is a Unix-like kernel which is written by Linus Torvalds and other developers, who communicate using the internet. Linux runs on many different architectures, for example on many IA32, IA64, Alpha, m68k, SPARC and PowerPC machines. The latest kernel and more information can be found at: http://www.kernel.org

Linux is often confused with the GNU/Linux system. Linux is only a kernel, not a complete operating system. GNU/Linux consists of the GNU operating system with the Linux kernel. Please read the following section for a more detailed explanation of GNU/Linux.

What is GNU/Linux?

At the beginning of the eighties Richard Stallman started an ambitious project with the goal to write a free Unix-like operating system. The name of this system is GNU (GNU is Not Unix). At the beginning of the nineties most important components of the GNU oparating system were written, except for the kernel, which is still under development under the name HURD. HURD consists of some servers which provide Unix-like kernel functionality. In turn these servers run under the Mach microkernel. At the beginning at the nineties the HURD team still had to wait till the Mach sources were released as free software. In the meanwhile Linus started filling the gap with the Linux kernel. GNU/Linux thus refers to the GNU system running on the Linux kernel. Right now the HURD kernel is also in a usable state and can be downloaded in the form of the GNU/HURD operating system. The Debian (http://www.debian.org/) project has even developed a version of the GNU operating system which works with the NetBSD (http://www.netbsd.org/) kernel. We should call "Linux distributions" "GNU/Linux distributions", because GNU is a substantial part of most distributions.

What is Slackware Linux?

Slackware Linux is a GNU/Linux distribution which is maintained and developed by Patrick Volkerding. In contrast to many other distributions Slackware adheres to the so-called KISS (keep it simple stupid) principle. This means that Slackware does not have complex graphical tools for configuring a system. For newbies this can be somwhat harsh, but it provides more transperancy and flexibility. Besides that you will get to learn GNU/Linux to the bones with Slackware Linux.

Another distinguishing aspect of Slackware Linux, that also "complies" with the KISS principe is the Slackware package manager. Slackware does not have complex package manager like RPM. Packages are normal tgz (tar/gzip) files, mostly with an additional installation script and a package description. Tgz is much more powerful than RPM for novice users and avoids dependency problems. Another famous feature of Slackware are the BSD-like initialization scripts of Slackware Linux. Slackware has one initialization script for each runlevel insteas of a script for eacht daemon. It allow you to tweak with your system easily, without the need to write net init scripts yourself.

The packages in Slackware Linux are compiled with as little modifications as possible. This means you can use most general GNU/Linux documentation.

Slackware Linux on CD-ROM

Slackware Linux can be purchased at quite many (internet) shops. It is important to make a distinction between the official CD-ROM set and cheap copies. When you buy the official CD set you are financially supporting the development of Slackware Linux. So, if you would like to see continuing development of Slackware Linux, buy the CD set!

Chapter 3. Sources of help

On your system

Linux HOWTO's

The famous Linux HOWTOs are a collection of documents which cover specific parts of a GNU/Linux system. Most HOWTOs are distribution independent, and therefore very useful for using them with Slackware Linux. The "linux-howtos" package in from the "f" diskset contains the HOWTO collection. After installing this package the HOWTOs can be found in the `/usr/doc/Linux-HOWTOs/` directory. Slackware also contains a collection of Mini-HOWTOs, which are shorter and cover narrower topics. The Mini-HOWTOs can be found in the `/usr/doc/Linux-mini-HOWTOs/` directory after installing the "linux-mini-howtos" package.

On the internet

alt.os.linux.slackware

alt.os.linux.slackware is a Slackware Linux newsgroup. You can read newsgroups with a newsreader like tin or knode. Be careful: it is expected that you have read all necessary documentation before posting to this newsgroup. If you have not the chance of getting "flamed" is really big.

Chapter 4. Installing Slackware Linux

Booting the installation CD-ROM

The easiest method for booting the installation system is by using the installation CD-ROM. The Slackware installation CD-ROM is a bootable CD, which means that the BIOS can boot the CD, just like it can boot, for example, a floppy disk. Most modern systems have a BIOS which supports CD-ROM booting.

If the CD is booted when you have the CD inserted in the CD-ROM drive during the system boot, the boot sequence is probably not correctly configured in the BIOS. Enter the BIOS setup (usually by this can be done by holding the "Del" or "Esc" key when the BIOS screen appears) and make sure the CD-ROM is on the top of the list in the boot sequence. If you are using a SCSI CD-ROM you may have to set the boot sequence in the SCSI BIOS instead of the system BIOS. Consult the SCSI card manual for more information.

After booting the installation system, you will be asked whether you are using a special (national) keyboard layout or not. If you have a normal US/International keyboard, which are the most common, you can just press "Enter" at this question. After that the login prompt will appear. Log on as “root”, no password will be requested. After logging on the shell is started, and you can start installing Slackware Linux. The installation procedure will be explained briefly in this chapter.

Partitioning a hard disk

Installing Slackware Linux requires at least one Linux partition, creating a swap partition is also recommended. To be able to create a partition there has to be free unpartitioned space on the disk. There are some programs that can resize partitions. For example, FIPS can resize FAT partitions. Commercial programs like Partition Magic can also resize other partition types.

After booting the Slackware Linux CD-ROM and logging on, there are two partitioning programs at your disposal: **fdisk** and **cfdisk**. **cfdisk** is the easiest of both, because it is controlled by a menu interface. This section describes the **cfdisk** program.

To partition the first harddisk you can simply execute **cfdisk**. If you want to partition another disk or a SCSI disk you have to specify which disk you want to partition (**cfdisk /dev/device**). An ATA hard disks have the following device naming: `/dev/hdn`, “n” is replaced by a character. E.g. the “primary master” is named `/dev/hda`, the “secondary slave” is named `/dev/hdd`. SCSI disks are named in the following way: `/dev/sdn`, “n” is replaced by the device character (the first SCSI disk = a, the fourth SCSI disk = d).

After starting **cfdisk** currently existing partitions are shown, as well as the amount of free space. The list of partitions can be navigated with the “up” and “down” arrow keys. At the bottom of the screen some commands are displayed, which can be browsed with the “left” and “right” arrow keys. A command can be executed with the <Enter> key.

You can create a Linux partition by selecting “Free Space” and executing the “New” command. **cfdisk** will ask you whether you want to create a primary or logical partition. The number of primary partitions is limited to four. Linux can be installed on both primary and logical partitions. If you want to install other operating systems besides Slackware Linux that require primary partitions, it is a good idea to install Slackware Linux onto a logical partition. The type of the new partition is automatically set to “Linux Native”, so it is not necessary to set the partition type.

The creation of a swap partition involves the same steps as a normal Linux partition, but the type of the partition has to be changed to “Linux Swap” after the partition is created. The suggested size of the swap partition depends on your own needs. The swap partition is used to store programs if the main (RAM) memory is full. If you have a harddisk of a reasonable size, it is a good idea to make a 256MB or 512MB swap partition, which should be enough for normal usage. After creating the partition the partition type can be changed to “Linux Swap”, by selecting the “Type” command. The **cfdisk** program will ask for the type number, “Linux Swap” partitions have type number 82. Normally number 82 is already selected, so you can go ahead by pressing the <Enter> key.

If you are satisfied with the partitioning you can save the changes by executing the “Write” command. This operation has to be confirmed by entering **yes**. After saving the changes you can quite **cfdisk** with the **Quit** command. It is a good idea to reboot the computer before starting the installation, to make sure that the partitioning changes are active. Press <ctrl> + <alt> + <del> to shut Linux down and restart the computer.

Chapter 5. Files and directories

Introduction

Unix-like operating systems use a hierarchical filesystem to store files and directories. Directories can contain files and other directories, the top directory (`/`) is named the root directory (not to be confused with the `/root` directory). Most filesystems also support file links (which provide alternative names for a file) and soft links. Other filesystems can be “connected” to an arbitrary directory. This process is named “mounting”, and the directory in which the filesystem is mounted is named the “mount point”.

This chapter covers the basic navigation of the filesystem, commands which are used to remove and create directories, filesystem permissions, links and mounting.

The basics

pwd

pwd(1) is a simple utility which shows the directory you are currently working in. The **pwd** does not require any parameters. This is an example output of **pwd**:

```
$ pwd
/home/danieldk
```

ls

ls is similar to the **dir** command in DOS and Windows. **ls** can be used to display files and directories located in specific directories. Running the **ls** command without any parameters shows the contents of the current directory:

```
$ ls
slackware-beginselen  slackware-beginselen-20december2002.tar.gz
```

Naturally it is also possible to show the contents of other directories. You can do this by specifying the path as a parameter tot the **ls** command:

```
$ ls /
bin   dev  home  lost+found  opt   root  tmp  var
boot  etc  lib   mnt         proc  sbin  usr
```

A disadvantage of the default output is that it provides little information about files and directories. For example, it is not possible to see whether some entry is a file or directory, what size a file is, or who the owner of the file is. The **ls** has the "-l" parameter to show more information:

```
$ ls -l
total 20
drwxr-xr-x    7 danieldk users        4096 Dec 21 09:24 slackware-beginselen
```

```
-rw-r--r--    1 danieldk users       14317 Dec 21 08:35 slackware-beginselen-20december2002.
```

cd

Another important command is the **cd** command. It can be used to change the current working directory:

```
$ cd /home/danieldk/
```

With the **pwd** command you can see it worked:

```
$ pwd
/home/danieldk
```

mkdir

As you might have guessed, the **mkdir**(1) command can be used to create directories. For example:

```
$ pwd
/home/danieldk
$ mkdir test
$ cd test
$ pwd
/home/danieldk/test
```

It might happen that you want to create a directory in a parent directory which does not exist yet. For example, if you want to create the `test2/hello/` directory, but the `test2` directory does not yet exist. In this case you can make both directories with only one **mkdir** command:

```
$ mkdir -p test2/hello
```

rm

The **rm**(1) is used to remove files and directories. Let's look at a simple example:

```
$ rm hello.c
```

This commands removes the file `hello.c`. Sometimes the **rm** asks for a confirmation. You can ignore this with the '-f' parameter:

```
$ rm -f *
```

This command removes all files in the current directory without asking for confirmation. It is also possible to delete directories or even whole directory trees. **rm** provides the '-r' parameter to do this. Suppose we want to delete the `ogle` directory and all subdirectories and files in that directory, this can be done in the following way:

```
$ rm -r -f ogle/
```

Many commands allow you to combine parameters. The following example is equivalent to the last one:

```
$ rm -rf ogle/
```

Permissions

A short introduction

Every file in GNU/Linux has permissions. As you might have noticed, file permissions can be shown with the **ls -l** command:

```
$ ls -l logo.jpg
-rw-r--r--    1 danieldk users        9253 Dec 23 19:12 logo.jpg
```

The permissions are shown in the first column. Permissions that can be set are read(r), write(w) and execute(x). These permissions can we set for three classes: owner(u), group(g) and others(o). The permissions are visible in the second to ninth character in the first column. These nine characters are divided in three groups. The first three characters represent the permissions for the owner, the next three characters represent the permissions for the group, finally the last three characters represent the permissions for other users. Thus, the example file shown above can be written to by the owner and can be read by all three classes of users (owner, group and others).

Each GNU/Linux system has many distinct users (a list of users can be found in `/etc/passwd`), and a user can be a member of certain groups. This kind of user management provides makes it possible to manage detailed permissions for each file. In the example shown above *danieldk* is the owner of the file and group permissions apply to the group *users*. In this example groups rights do not differ from the rights of other users.

chown

The **chown**(1) command is used to set the file owner and to which group group permissions should apply to. Suppose we want to make *danieldk* the owner of the file `logo2.jpg`, this can be done with the **chown**:

```
$ chown danieldk logo2.jpg
```

Using the **ls** we can see that the owner is now *danieldk*:

```
$ ls -l logo2.jpg
-rw-r--r--    1 root     root        9253 Dec 29 11:35 logo2.jpg
$ chown danieldk logo2.jpg
$ ls -l logo2.jpg
-rw-r--r--    1 danieldk root        9253 Dec 29 11:35 logo2.jpg
```

But group permissions still apply for the *root* group. This group can be changed by adding a dot after the owner, followed by the name of the group (in this example the group is nedslackers):

```
$ chown danieldk.nedslackers logo2.jpg
$ ls -l logo2.jpg
-rw-r--r--    1 danieldk nedslackers        9253 Dec 29 11:35 logo2.jpg
```

It is also possible to change ownership recursively, this can be done with the recursive (-R) parameter:

```
$ chown -R danieldk.users oggs/
```

chmod

File permissions can be manipulated using the **chmod**(1) command. The most basic syntax of chmod is **chmod [u,g,o][+/-][r,w,x] filename**. The first parameter consists of the following elements: 1. which classes this manipulation permission applies to, 2. if the permissions should be added (+) or removed (-), and 3. which permissions should be manipulated. Suppose we want to make the file `memo` writable for the owner of the file and the groups for which the group permissions apply. This can be done with the following command:

```
$ chmod ug+w memo
```

As you can see below the `memo` is now writable for the file owner and group:

```
$ ls -l notities
-r--r--r--    1 daniel   users          12 Mar  9 16:28 memo
bash-2.05b$ chmod ug+w memo
bash-2.05b$ ls -l notities
-rw-rw-r--    1 daniel   users          12 Mar  9 16:28 momo
```

Just like the **chown** command it is also possible to do recursive (-R) operations. In the following example the `secret/`, including subdirectories and files in this directory, is made unreadable for the group set for this directory and other users:

```
$ chmod -R go-r geheim/
```

Mounting filesystems

Introduction

Like most Unices Linux uses a technique named “mounting” to access filesystems. Mounting means that a filesystem is connected to a directory in the root filesystem. One could for example mount a CD-ROM drive to the `/mnt/cdrom` directory. Linux supports many kinds of filesystems, like Ext2, Ext3, ReiserFS, JFS, XFS, ISO9660 (used for CD-ROMs), UDF (used on some DVDs) and DOS/Windows filesystems, like FAT, FAT32 and NTFS. These filesystems can reside on many kinds of media, for example hard drives, CD-ROMs and Flash drives. This section explains how filesystems can be mounted and unmounted.

mount

The **mount**(8) is used to mount filesystems. The basic syntax is: “mount /dev/devname /mountpoint”. The device name can be any block device, like hard disks or CD-ROM drives. The mount point can be an arbitrary point in the root filesystem. Let’s look at an example:

```
# mount /dev/cdrom /mnt/cdrom
```

This mounts the /dev/cdrom on the /mnt/cdrom mountpoint. The /dev/cdrom device name is normally a link to the real CD-ROM device name (for example, /dev/hdc). As you can see, the concept is actually very simple, it just takes some time to learn the device names ;). Sometimes it is necessary to specify which kind of filesystem you are trying to mount. The filesystem type can be specified by adding the "-t" parameter:

```
# mount -t vfat /dev/sda1 /mnt/usbflash
```

This mounts the vfat filesystem on /dev/sda1 to /mnt/usbflash.

umount

The **umount**(1) command is used to unmount filesystems. **umount** accepts two kinds of parameters, mount points or devices. For example:

```
# umount /mnt/cdrom
# umount /dev/sda1
```

The first command unmounts the filesystem that was mounted on /mnt/cdrom, the second commands unmounts the filesystem on /dev/sda1.

Chapter 6. Process management

Introduction

Unix-like operating systems work with processes. A process is an unit the operating system schedules for CPU time and the memory manager manages memory for. Basically a process consists of program code (named text), data (used by a program) and a stack. The stack is used by the program to store variables. Programs are at least one process. A program/process can ask the system to create a new copy of itself, which is called a fork. For example, a web server could fork itself to let the new process handle a request.

A process can be parted in threads. The difference between forking a process and creating a thread is that different threads share the address space of the process. A forked process is a separate process with its own address space. Forking is more expensive in terms of memory requirement and CPU time.

A user can control a process by sending signals to the process. For example, the SIGTERM command is used to terminate a process, and the SIGHUP signal to restart a process.

Process basics

This section describes some basic commands that are used for process management.

ps

The **ps**(1) command is used to report which processes are currently active. By running **ps** without any parameters you can see which processes are active in the current user session. Let's look at an example:

```
$ ps
  PID TTY          TIME CMD
 1191 pts/2    00:00:00 bash
 1216 pts/2    00:00:00 ps
```

In this example the **bash** and **ps** commands are running. As you can see each process has a process ID (PID). You will need the process number if you want to send a signal to a process, for example a kill signal. The **ps** has many parameters to modify the output. For example, the `x` shows all processes without a controlling tty:

```
$ ps x
  PID TTY      STAT   TIME COMMAND
 1044 tty1     S      0:00 -bash
 1089 tty1     S      0:00 /bin/sh /usr/X11R6/bin/startx
 1100 tty1     S      0:00 xinit /home/daniel/.xinitrc --
 1108 tty1     S      0:00 /usr/bin/wmaker
 1113 tty1     S      0:00 sylpheed
 1114 tty1     S      0:00 /bin/sh /opt/firefox/run-mozilla.sh /opt/firefox/fire
 1120 tty1     S      0:52 /opt/firefox/firefox-bin
 1125 tty1     S      0:00 /usr/libexec/gconfd-2 20
 1146 tty1     S      0:00 xchat
```

```
 1161 tty1     S      0:00 xterm -sb
 1163 pts/0    S      0:00 bash
 1170 pts/0    S      0:00 vi proc.xml
 1189 tty1     S      0:00 xterm -sb
 1191 pts/2    S      0:00 bash
 1275 pts/2    R      0:00 ps x
```

Have a look at the **ps**(1) manual page for a summary of available parameters.

kill

The **kill**(1) sends a signal to a process. If no signal is specified the TERM signal is send, which asks a process to exit gracefully. Let's have a look at the normal mode of execution:

```
$ ps ax | grep mc
 1045 tty4     S      0:00 /usr/bin/mc -P /tmp/mc-daniel/mc.pwd.756
$ kill 1045
$ ps ax | grep mc
$
```

As you can see the **ps** is used to look for the **mc** process. There is one occurrence of **mc** running with PID 1045. This process is killed, and the second **ps** command shows that the process is indeed terminated.

As we said earlier the **kill** command can also be used to send other signals. The **kill -l** displays a list of signals that can be sent:

```
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL
 5) SIGTRAP      6) SIGABRT      7) SIGBUS       8) SIGFPE
 9) SIGKILL     10) SIGUSR1     11) SIGSEGV     12) SIGUSR2
13) SIGPIPE     14) SIGALRM     15) SIGTERM     17) SIGCHLD
18) SIGCONT     19) SIGSTOP     20) SIGTSTP     21) SIGTTIN
22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO
30) SIGPWR      31) SIGSYS
```

The SIGKILL signal is often used to kill processes that refuse to terminate with the default SIGTERM signal. The signal can be specified by using the number as a parameter, for example, the following command would send a SIGKILL signal to PID 1045:

```
$ kill -9 1045
```

It is also possible to specify the signal without the "SIG" letters as a parameter. In the following example the SIGHUP signal is sent to the **inetd** to restart it:

```
# ps ax | grep inetd
  727 ?        S      0:00 /usr/sbin/inetd
# kill -HUP 727
```

Chapter 7. Printer configuration

Introduction

GNU/Linux supports a large share of the available USB, parallel and network printers. Slackware Linux provides two printing systems, CUPS and LPRNG. This chapter covers the CUPS system.

Independent of which printing system you are going to use, it is a good idea to install some printer filter collections. These can be found in the “ap” diskset. If you want to have support for most printers, make sure the following packages are installed:

```
a2ps
enscript
espgs
gimp-print
gnu-gs-fonts
hpijs
ifhp
```

Both printing systems have their own advantages and disadvantages. If you do not have much experience with configuring printers under GNU/Linux, it is a good idea to use CUPS, because CUPS provides a comfortable web interface which can be accessed through a web browser.

Preparations

To be able to use cups the “cups” package from the “a” diskset has to be installed. After the installation CUPS can be started automatically during each system boot by making `/etc/rc.d/rc.cups` executable. This can be done with the following command:

```
# chmod a+x /etc/rc.d/rc.cups
```

After restarting the system CUPS will also be restarted automatically. You can start CUPS on a running system by executing the following command:

```
# /etc/rc.d/rc.cups start
```

Configuration

CUPS can be configured via a web interface. The configuration interface can be accessed with a web browser at the following URL: http://localhost:631/. Some parts of the web interface require that you authenticate yourself. If an authentication window pops up you can enter “root” as the user name, and fill in the root account password.

A printer can be added to the CUPS configuration by clicking on “Administrate”, and clicking on the “Add Printer” button after that. The web interface will ask for three options:

- *Name* - the name of the printer. Use a simple name, for example “epson”.
- *Location* - the physical location of the printer. This setting is not crucial, but handy for larger organizations.
- *Description* - a description of the printer, for example “Epson Stylus Color C42UX”.

You can proceed by clicking the “Continue” button. On the next page you can configure how the printer is connected. If you have an USB printer which is turned on, the web interface will show the name of the printer next to the USB port that is used. After configuring the printer port you can select the printer brand and model. After that the printer configuration is finished, and the printer will be added to the CUPS configuration.

An overview of the configured printers can be found on the “Printers” page. On this page you can also do some printer operations. For example, “Print Test Page” can be used to check the printer configuration by printing a test page.

Chapter 8. XFree86

X Configuration

The XFree86 configuration is stored in `/etc/X11/XF86Config`. Many distributions provide special configuration tools for X, but Slackware Linux only provide the standard XFree86 tools (which are actually quite easy to use). In most cases X can be configured automatically, but sometimes it is necessary to edit `/etc/X11/XF86Config` manually.

Automatical configuration

The XFree86 server provides an option to automatically generate a configuration file. XFree86 will load all available driver modules, and will try to detect the hardware, and generate a configuration file. Execute the following command to generate a XFree86 configuration file:

```
$ XFree86 -configure
```

If X does not output any errors, the generated configuration can be copied to the `/etc/X11` directory. And X can be started to test the configuration:

```
$ cp /root/XF86Config /etc/X11/
$ startx
```

Interactive configuration

XFree86 provides two tools for configuring X interactively, **xf86config** and **xf86config**. **xf86cfg** tries to detect the video card automatically, and starts an tool which can be used to tune the configuration. Sometimes **xf86cfg** switches to a video mode which is not supported by the monitor. In that case **xf86cfg** can also be used in text-mode, by starting it with **xf86cfg -textmode**.

xf86config differs from the tools described above, it does not detect hardware and will ask detailed questions about your hardware. If you only have little experience configuring XFree86 it is a good idea to avoid **xf86config**.

Window manager

The "look and feel" of XFree86 is managed by a so-called window manager. Slackware Linux provides the following widely user window managers:

- WindowMaker: A relatively light window manager, which is part of the GNUStep project.
- BlackBlox: Light window manager, BlackBox has no dependencies except the X11 libraries.
- KDE: A complete desktop environment, including browser, e-mail program and an office suite (KOffice).

- GNOME: Like KDE a complete desktop environment. It is worth noting that Dropline Systems (http://www.dropline.net/) provides a special GNOME environment for Slackware.

If you are used to a desktop environment, using KDE or GNOME is a logical choice. But it is a good idea to try some of the lighter window managers. They are faster, and consumer less memory, besides that most KDE and GNOME applications are perfectly usable under other window managers.

On Slackware Linux the following command can be used to select a window manager:

```
$ xwmconfig
```

This configuration program shows the installed window managers, from which you can choose one. You can set the window manager globally by executing **xwmconfig** as root.

Chapter 9. Pkgtools

Introduction

Slackware does not use a complex package system, unlike many other Linux distributions. Package have the `.tgz` extension, and are usually ordinary tarballs which contain two extra files: an installation script and a package description file. Due to the simplicity of the packages the Slackware package tools do not have the means to handle dependencies. But many Slackware users prefer this approach, because dependencies often cause more problems than they solve.

Slackware has a few tools to handle packages. The most important tools will be covered in this chapter. To learn to understand the tools we need to have a look at package naming. Let's have a look at an example, imagine that we have a package with the filename `bash-2.05b-i386-2.tgz`. In this case the name of the package is bash-2.05b-i386-2. In the package name information about the package is seperated by the '-' character. A package name has the following meaning: "programname-version-architecture-packagerevision"

pkgtool

The **pkgtool** command provides a menu interface for some package operations. De most important menu items are "Remove" and "Setup". De "Remove" option presents a list of installed packages. You can select which packages you want to remove with the spacebar and confirm your choices with the return key. You can also deselect a package for removal with the spacebar.

The "Setup" options provides access to a few tools which can help you with configuring your system, for example: **netconfig**, **pppconfig** and **xwmconfig**.

installpkg

The **installpkg** command is used to install packages. **installpkg** needs a packagefile as a parameter. For example, if you want to install the package `bash-2.05b-i386-2.tgz` execute:

```
# installpkg bash-2.05b-i386-2.tgz
```

upgradepkg

upgradepkg can be used to upgrade packages. In contrast to **installpkg** it only installs packages when there is an older version available on the system. The command syntax is comparable to **installpkg**. For example, if you want to upgrade packages using package in a directory execute:

```
# upgradepkg *.tgz
```

As said only those packages will be installed of which an other version is already installed on the system.

removepkg

The **removepkg** can be used to remove installed packages. For example, if you want to remove the "bash" package (it is not recommended to do that!), you can execute:

```
# removepkg bash
```

As you can see only the name of the package is specified in this example. You can also remove a package by specifying its full name:

```
# removepkg bash-2.05b-i386-2
```

Chapter 10. Networking configuration

Hardware

Network cards (NICs)

Drivers for NICs are installed as kernel modules. The module for your NIC has to be loaded during the initialization of Slackware. On most systems the NIC is automatically detected and configured during the installation of Slackware Linux. You can reconfigure your NIC with the **netconfig** command. **netconfig** adds the driver (module) for the detected card to `/etc/rc.d/rc.netdevice`.

It is also possible to manually configure which modules should be loaded during the initialization of the system. This can be done by adding a **modprobe** line to `/etc/rc.d/rc.modules`. For example, if you want to load the module for 3Com 59x NICs (3c59x.o), add the following line to `/etc/rc.d/rc.modules`

```
/sbin/modprobe 3c59x
```

PCMCIA cards

Supported PCMCIA cards are detected automatically by the PCMCIA software. The pcmcia-cs packages from the "a" diskset provides PCMCIA functionality for Slackware Linux.

Configuration of interfaces

Network cards are available under Linux through so-called "interfaces". The **ifconfig**(8) command can be used to display the available interfaces:

```
# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:20:AF:F6:D4:AD
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1301 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1529 errors:0 dropped:0 overruns:0 carrier:0
          collisions:1 txqueuelen:100
          RX bytes:472116 (461.0 Kb)  TX bytes:280355 (273.7 Kb)
          Interrupt:10 Base address:0xdc00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:77 errors:0 dropped:0 overruns:0 frame:0
          TX packets:77 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:8482 (8.2 Kb)  TX bytes:8482 (8.2 Kb)
```

Network cards get the name ethn, in which n is a number, starting with 0. In the example above, the first network card (eth0) already has an IP address. But unconfigured interfaces have no IP address, the **ifconfig** will not show IP addresses for unconfigured interfaces. Interfaces can be configured in the `/etc/rc.d/rc.inet1.conf` file. You can simply read the comments, and fill in the required information. For example:

```
# Config information for eth0:
IPADDR[0]="192.168.1.1"
NETMASK[0]="255.255.255.0"
USE_DHCP[0]=""
DHCP_HOSTNAME[0]=""
```

In this example the IP address 192.168.1.1 with the 255.255.255.0 netmask is assigned to the first ethernet interface (eth0). If you are using a DHCP server you can change the *USE_DHCP=""* line to *USE_DHP[n]="yes"* (swap “n” with the interface number). Other variables, except *DHCP_HOSTNAME* are ignored when using DHCP. For example:

```
IPADDR[1]=""
NETMASK[1]=""
USE_DHCP[1]="yes"
DHCP_HOSTNAME[1]=""
```

The same applies to other interfaces. You can activate the new settings by rebooting the system or by executing **/etc/rc.d/rc.inet1**.

Resolving

Hostname

Each computer on the internet has a hostname. If you do not have a hostname that is resolvable with DNS, it is still a good idea to configure your hostname, because some software uses it. You can configure the hostname in `/etc/HOSTNAME`. A single line with the hostname of the machine will suffice. Normally a hostname has the following form: host.domain.tld, for example darkstar.slackfans.org. Be aware that the hostname has to be resolvable, meaning that Linux should be able to convert the hostname to an IP address. You can make sure the hostname is resolvable by adding it to `/etc/hosts`. Read the following section for more information about this file.

/etc/hosts

`/etc/hosts` is a table of IP addresses with associated hostnames. This file can be used to name computers in a small network. Let’s look at an example of the `/etc/hosts` file:

```
127.0.0.1        localhost
192.168.1.1             tazzy.slackfans.org tazzy
192.168.1.2             gideon.slackfans.org
```

The localhost line should always be present. It assigns the name "localhost" to a special interface, the loopback. In this example the names "tazzy.slackfans.org" and "tazzy" are assigned to the IP address

192.168.1.1, and the name "gideon.slackfans.org" is assigned to the IP address 192.168.1.2. On the system with this file both computers are available via the mentioned hostnames.

/etc/resolv.conf

The `/etc/resolv.conf` file is used to specify which nameservers the system should use. A nameserver converts hostnames to IP addresses. Your provider should have given you at least two name name server addresses (DNS servers). You can add these nameservers to `/etc/resolv.conf` by adding the line "nameserver ipaddress" for each nameserver. For example:

```
nameserver 192.168.1.1
nameserver 192.168.1.69
```

You can check wether the hostnames are tranlated correctly or not with the **host hostname** command. Swap "hostname" with an existing hostname, for example the website of your internet service provider.

IPv4 Forwarding

IPv4 forwarding connects two or more networks by sending packets which arrive on one interface to another interface. This makes it possible to let a Linux machine act as a router. For example, you can connect your home network with the internet. IPv4 forwarding can be enabled or disabled under Slackware Linux by changing the IPV4_FORWARD variable in `/etc/rc.d/rc.inet2`. The default setting is as follows:

```
IPV4_FORWARD=1
```

This means that IPv4 forwarding is enabled. You can disable forwarding by changing the IPV4_FORWARD value to 0. This setting can be enabled by rebooting the computer. It is also possible to enable IPv4 forwarding on a running system with the following command (0 disables forwarding, 1 enables forwarding):

```
# echo 0 > /proc/sys/net/ipv4/ip_forward
```

Be cautious! By default there are no active IP filters. This means that anyone can access other networks. Traffic can be filtered and logged with the iptables kernel packet filter. Iptables can be administrated through the **iptables** command. NAT (Network Address Translation) is also a subset of iptables, and can be controlled and enabled through the **iptables** command. NAT makes it possible to "hide" a network behind one IP address. This allows you to use the internet on a complete network with only one IP address.

Chapter 11. Networking services

The internet super server

There are two ways to offer TCP/IP services: by running server applications standalone as a daemon or by using the internet super server, **inetd**(8). **inetd** is a daemon which monitors a range of ports. If a client attempts to connect to a port **inetd** handles the connection and forwards the connection to the server software which handles that kind of connection. The advantage of this approach is that it adds an extra layer of security and it makes it easier to log incoming connections. The disadvantage is that it is somewhat slower than using a standalone daemon. It is thus a good idea to run a standalone daemon on, for example, a heavily loaded FTP server.

inetd can be configured using the `/etc/inetd.conf` file. Let's have a look at an example line from `inetd.conf`:

```
# File Transfer Protocol (FTP) server:
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  proftpd
```

This line specifies that **inetd** should accept FTP connections and pass them to **tcpd**. This may seem a bit odd, because **proftpd** normally handles FTP connections. You can also specify to use **proftpd** directly in `inetd.conf`, but Slackware normally passes the connection to **tcpd**. This program passes the connection to **proftpd** in turn, as specified. **tcpd** is used to monitor services and to provide host based access control.

Services can be disabled by adding the comment character (#) at the beginning of the line. It is a good idea to disable all services and enable services you need one at a time. After changing `/etc/inetd.conf` **inetd** needs to be restarted to activate the changes. This can be done by sending the HUP signal to the inetd process:

```
# ps ax | grep 'inetd'
   64 ?        S      0:00 /usr/sbin/inetd
# kill -HUP 64
```

Caching nameserver

A caching nameserver provides DNS services for a machine or a network, but does not provide DNS for a domain. That means it can only be used to convert hostnames to IP addresses. Setting up a nameserver with Slackware Linux is fairly easy, because BIND is configured as a caching nameserver by default. Enabling the caching nameserver takes just two steps: you have to install BIND and alter the initialization scripts. BIND can be installed by adding the bind package from the "n" diskset. After that bind can be started by executing the **named(8)** command. If want to start BIND by default, make the `/etc/rc.d/rc.bind` file executable. This can be done by executing the following command as root:

```
# chmod a+x /etc/rc.d/rc.bind
```

If you want to use the nameserver on the machine that runs BIND, you also have to alter `/etc/resolv.conf`.

The Apache webserver

Apache is the most widely userd webserver on the internet, and is, of course, part of Slackware Linux. Apache can be installed by adding the `apache` pacakge from the "n" diskset. If you also want to use PHP, the `php` ("n" diskset) and `mysql` ("ap" diskset) are also required. MySQL is required, because the precompiled PHP depends on MySQL shared libraries. You do not have to run MySQL itself. After installing Apache it can be started automatically while booting the system by making the `/etc/rc.d/rc.httpd` file executable. You can do this by executing:

```
# chmod a+x /etc/rc.d/rc.httpd
```

The Apache configuration can be altered in the `/etc/apache/httpd.conf` file. Apache can be stopped/started/restarted every moment with the **apachectl** command, and the "stop", "start" and "restart" parameters. For example, execute the following command to restart Apache:

```
# apachectl restart
/usr/sbin/apachectl restart: httpd restarted
```

Chapter 12. XFree86

X Configuration

The XFree86 configuration is stored in `/etc/X11/XF86Config`. Many distributions provide special configuration tools for X, but Slackware Linux only provide the standard XFree86 tools (which are actually quite easy to use). In most cases X can be configured automatically, but sometimes it is necessary to edit `/etc/X11/XF86Config` manually.

Automatical configuration

The XFree86 server provides an option to automatically generate a configuration file. XFree86 will load all available driver modules, and will try to detect the hardware, and generate a configuration file. Execute the following command to generate a XFree86 configuration file:

```
$ XFree86 -configure
```

If X does not output any errors, the generated configuration can be copied to the `/etc/X11` directory. And X can be started to test the configuration:

```
$ cp /root/XF86Config /etc/X11/
$ startx
```

Interactive configuration

XFree86 provides two tools for configuring X interactively, **xf86config** and **xf86config**. **xf86cfg** tries to detect the video card automatically, and starts an tool which can be used to tune the configuration. Sometimes **xf86cfg** switches to a video mode which is not supported by the monitor. In that case **xf86cfg** can also be used in text-mode, by starting it with **xf86cfg -textmode**.

xf86config differs from the tools described above, it does not detect hardware and will ask detailed questions about your hardware. If you only have little experience configuring XFree86 it is a good idea to avoid **xf86config**.

Window manager

The "look and feel" of XFree86 is managed by a so-called window manager. Slackware Linux provides the following widely user window managers:

- WindowMaker: A relatively light window manager, which is part of the GNUStep project.
- BlackBlox: Light window manager, BlackBox has no dependencies except the X11 libraries.
- KDE: A complete desktop environment, including browser, e-mail program and an office suite (KOffice).

- GNOME: Like KDE a complete desktop environment. It is worth noting that Dropline Systems (http://www.dropline.net/) provides a special GNOME environment for Slackware.

If you are used to a desktop environment, using KDE or GNOME is a logical choice. But it is a good idea to try some of the lighter window managers. They are faster, and consumer less memory, besides that most KDE and GNOME applications are perfectly usable under other window managers.

On Slackware Linux the following command can be used to select a window manager:

```
$ xwmconfig
```

This configuration program shows the installed window managers, from which you can choose one. You can set the window manager globally by executing **xwmconfig** as root.

Appendix A. GNU Free Documentation License

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The

Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.