Linux Filesystem Hierarchy

Version 0.65

Binh Nguyen

<linuxfilesystem(at)yahoo(dot)com(dot)au>

2004−07−30

This document outlines the set of requirements and guidelines for file and directory placement under the Linux operating system according to those of the FSSTND v2.3 final (January 29, 2004) and also its actual implementation on an arbitrary system. It is meant to be accessible to all members of the Linux community, be distribution independent and is intended to discuss the impact of the FSSTND and how it has managed to increase the efficiency of support interoperability of applications, system administration tools, development tools, and scripts as well as greater uniformity of documentation for these systems.

Table of Contents

Table of Contents

Source and pre−formatted versions available

The source code and other machine readable formats of this book can be found on the Internet at the Linux Documentation Project home page http://www.tldp.org The latest version of this document can be found at http://cvsview.tldp.org/index.cgi/LDP/guide/docbook/Linux−Filesystem−Hierarchy/

Chapter 1. Linux Filesystem Hierarchy

1.1. Foreward

When migrating from another operating system such as Microsoft Windows to another; one thing that will profoundly affect the end user greatly will be the differences between the filesystems.

What are filesystems?

A *filesystem* is the methods and data structures that an operating system uses to keep track of files on a disk or partition; that is, the way the files are organized on the disk. The word is also used to refer to a partition or disk that is used to store the files or the type of the filesystem. Thus, one might say I have two filesystems meaning one has two partitions on which one stores files, or that one is using the extended filesystem, meaning the type of the filesystem.

The difference between a disk or partition and the filesystem it contains is important. A few programs (including, reasonably enough, programs that create filesystems) operate directly on the raw sectors of a disk or partition; if there is an existing file system there it will be destroyed or seriously corrupted. Most programs operate on a filesystem, and therefore won't work on a partition that doesn't contain one (or that contains one of the wrong type).

Before a partition or disk can be used as a filesystem, it needs to be initialized, and the bookkeeping data structures need to be written to the disk. This process is called *making a filesystem*.

Most UNIX filesystem types have a similar general structure, although the exact details vary quite a bit. The central concepts are *superblock*, *inode*, *data block*, *directory block*, and *indirection block*. The superblock contains information about the filesystem as a whole, such as its size (the exact information here depends on the filesystem). An inode contains all information about a file, except its name. The name is stored in the directory, together with the number of the inode. A directory entry consists of a filename and the number of the inode which represents the file. The inode contains the numbers of several data blocks, which are used to store the data in the file. There is space only for a few data block numbers in the inode, however, and if more are needed, more space for pointers to the data blocks is allocated dynamically. These dynamically allocated blocks are indirect blocks; the name indicates that in order to find the data block, one has to find its number in the indirect block first.

Like UNIX, Linux chooses to have a single hierarchical directory structure. Everything starts from the root directory, represented by /, and then expands into sub−directories instead of having so−called 'drives'. In the Windows environment, one may put one's files almost anywhere: on C drive, D drive, E drive etc. Such a file system is called a hierarchical structure and is managed by the programs themselves (program directories), not by the operating system. On the other hand, Linux sorts directories descending from the root directory / according to their importance to the boot process.

If you're wondering why Linux uses the frontslash / instead of the backslash \ as in Windows it's because it's simply following the UNIX tradition. Linux, like Unix also chooses to be case sensitive. What this means is that the case, whether in capitals or not, of the characters becomes very important. So this is not the same as THIS. This feature accounts for a fairly large proportion of problems for new users especially during file transfer operations whether it may be via removable disk media such as floppy disk or over the wire by way of FTP.

The filesystem order is specific to the function of a file and not to its program context (the majority of Linux filesystems are 'Second Extended File Systems', short 'EXT2' (aka 'ext2fs' or 'extfs2') or are themselves subsets of this filesystem such as ext3 and Reiserfs). It is within this filesystem that the operating system determines into which directories programs store their files.

If you install a program in Windows, it usually stores most of its files in its own directory structure. A help file for instance may be in C:\Program Files\[program name]\ or in C:\Program Files\[program−name]\help or in C:\Program Files\[program −name]\humpty\dumpty\doo. In Linux, programs put their documentation into /usr/share/doc/[program−name], man(ual) pages into /usr/share/man/man[1−9] and info pages into /usr/share/info. They are merged into and with the system hierarchy.

As all Linux users know, unless you mount a partition or a device, the system does not know of the existence of that partition or device. This might not appear to be the easiest way to provide access to your partitions or devices, however it offers the advantage of far greater flexibility when compared to other operating systems. This kind of layout, known as the unified filesystem, does offer several advantages over the approach that Windows uses. Let's take the example of the /usr directory. This sub−directory of the root directory contains most of the system executables. With the Linux filesystem, you can choose to mount it off another partition or even off another machine over the network using an innumerable set of protocols such as NFS (Sun), Coda (CMU) or AFS (IBM). The underlying system will not and need not know the difference. The presence of the /usr directory is completely transparent. It appears to be a local directory that is part of the local directory structure.

Compliance requires that:

```
+---------+-----------------+-------------+
|         | shareable       | unshareable |
+---------+-----------------+-------------+
|static   | /usr            | /etc        |
|         | /opt            | /boot       |
+---------+-----------------+-------------+
|variable | /var/mail       | /var/run    |
|         | /var/spool/news | /var/lock   |
+---------+-----------------+-------------+

"Shareable" files are defined as those that can be stored on one host and
used on others. "Unshareable" files are those that are not shareable. For
example, the files in user home directories are shareable whereas device
lock files are not. "Static" files include binaries, libraries,
documentation files and other files that do not change without system
administrator intervention. "Variable" files are defined as files that
are not static.
```

Another reason for this unified filesystem is that Linux caches a lot of disk accesses using system memory while it is running to accelerate these processes. It is therefore vitally important that these buffers are flushed (get their content written to disk), before the system closes down. Otherwise files are left in an undetermined state which is of course a very bad thing. Flushing is achieved by 'unmounting' the partitions during proper system shutdown. In other words, don't switch your system off while it's running! You may get away with it quite often, since the Linux file system is very robust, but you may also wreak havoc upon important files. Just hit ctrl−alt−del or use the proper commands (e.g. shutdown, poweroff, init 0). This will shut down the system in a decent way which will thus, guarantee the integrity of your files.

Many of us in the Linux community have come to take for granted the existence of excellent books and documents about Linux, an example being those produced by the Linux Documentation Project. We are used to having various packages taken from different sources such as Linux FTP sites and distribution CD−ROMs integrate together smoothly. We have come to accept that we all know where critical files like mount can be found on any machine running Linux. We also take for granted CD−ROM based distributions that can be run directly from the CD and which consume only a small amount of physical hard disk or a RAM disk for some variable files like /etc/passwd, etc. This has not always been the case.

During the adolescent years of Linux during the early to mid−90s each distributor had his own favorite scheme for locating files in the directory hierarchy. Unfortunately, this caused many problems. The *Linux File System Structure* is a document, which was created to help end this anarchy. Often the group, which creates this document or the document itself, is referred to as the FSSTND. This is short for file system standard". This document has helped to standardize the layout of file systems on Linux systems everywhere. Since the original release of the standard, most distributors have adopted it in whole or in part, much to the benefit of all Linux users.

Since the first draft of the standard, the FSSTND project has been coordinated by Daniel Quinlan and development of this standard has been through consensus by a group of developers and Linux enthusiasts. The FSSTND group set out to accomplish a number of specific goals. The first goal was to solve a number of problems that existed with the current distributions at the time. Back then, it was not possible to have a shareable /usr partition, there was no clear distinction between /bin and /usr/bin, it was not possible to set up a diskless workstation, and there was just general confusion about what files went where. The second goal was to ensure the continuation of some reasonable compatibility with the de−facto standards already in use in Linux and other UNIX−like operating systems. Finally, the standard had to gain widespread approval by the developers, distributors, and users within the Linux community. Without such support, the standard would be pointless, becoming just another way of laying out the file system.

Fortunately, the FSSTND has succeeded though there are also some goals that the FSSTND project did not set out to achieve. The FSSTND does not try to emulate the scheme of any specific commercial UNIX operating system (e.g. SunOS, AIX, etc.) Furthermore, for many of the files covered by the FSSTND, the standard does not dictate whether the files should be present, merely where the files should be if they are present. Finally, for most files, the FSSTND does not attempt to dictate the format of the contents of the files. (There are some specific exceptions when several different packages may need to know the file formats to work together properly. For example, lock files that contain the process ID of the process holding the lock.) The overall objective was to establish the location where common files could be found, if they existed on a particular machine. The FSSTND project began in early August 1993. Since then, there have been a number of public revisions of this document. The latest, v2.3 was released on January 29, 2004.

If you're asking "What's the purpose of all this? Well, the answer depends on who you are. If you are a Linux user, and you don't administrate your own system then the FSSTND ensures that you will be able to find programs where you'd expect them to be if you've already had experience on another Linux machine. It also ensures that any documentation you may have makes sense. Furthermore, if you've already had some experience with Unix before, then the FSSTND shouldn't be too different from what you're currently using, with a few exceptions. Perhaps the most important thing is that the development of a standard brings Linux to a level of maturity authors and commercial application developers feel they can support.

If you administer your own machine then you gain all the benefits of the FSSTND mentioned above. You may also feel more secure in the ability of others to provide support for you, should you have a problem. Furthermore, periodic upgrades to your system are theoretically easier. Since there is an agreed−upon standard for the locations of files, package maintainers can provide instructions for upgrading that will not leave extra, older files lying around your system inhabiting valuable disk space. The FSSTND also means that there is more support from those providing source code packages for you to compile and install yourself. The provider knows, for example, where the executable for sed is to be found on a Linux machine and can use that in his installation scripts or Makefiles.

If you run a large network, the FSSTND may ease many of your NFS headaches, since it specifically addresses the problems which formerly made shared implementations of /usr impractical. If you are a distributor, then you will be affected most by the Linux FSSTND. You may have to do a little extra work to make sure that your distribution is FSSTND−compliant, but your users (and hence your business) will gain by it. If your system is compliant, third party add−on packages (and possibly your own) will integrate smoothly with your system. Your users will, of course, gain all the benefits listed above, and many of your support headaches will be eased. You will benefit from all the discussion and thought that has been put into the FSSTND and avoid many of the pitfalls involved in designing a filesystem structure yourself. If you adhere to the FSSTND, you will also be able to take advantage of various features that the FSSTND was designed around. For example, the FSSTND makes "live" CD−ROMs containing everything except some of the files in the / and /var directories possible. If you write documentation for Linux, the FSSTND makes it much easier to do so, which makes sense to the Linux community. You no longer need to worry about the specific location of lock files on one distribution versus another, nor are you forced to write documentation that is only useful to the users of a specific distribution. The FSSTND is at least partly responsible for the recent explosion of Linux books being published.

If you are a developer, the existence of the FSSTND greatly eases the possibility for potential problems. You can know where important system binaries are found, so you can use them from inside your programs or your shell scripts. Supporting users is also greatly eased, since you don't have to worry about things like the location of these binaries when resolving support issues. If you are the developer of a program that needs to integrate with the rest of the system, the FSSTND ensures that you can be certain of the steps to meet this end. For example, applications such as kermit, which access the serial ports, need to know they can achieve exclusive access to the TTY device. The FSSTND specifies a common method of doing this so that all compliant applications can work together. That way you can concentrate on making more great software for Linux instead of worrying about how to detect and deal with the differences in flavors of Linux. The widespread acceptance of the FSSTND by the Linux community has been crucial to the success of both the standard and operating system. Nearly every modern distribution conforms to the Linux FSSTND. If your implementation isn't at least partially FSSTND compliant, then it is probably either very old or you built it yourself. The FSSTND itself contains a list of some of the distributions that aim to conform to the FSSTND. However, there are some distributions that are known to cut some corners in their implementation of FSSTND.

By no means does this mean that the standard itself is complete. There are still unresolved issues such as the organization of architecture−independent scripts and data files /usr/share. Up until now, the i386 has been the primary platform for Linux, so the need for standardization of such files was non−existent.

The rapid progress in porting Linux to other architectures (MC680x0, Alpha, MIPS, PowerPC) suggests that this issue will soon need to be dealt with. Another issue that is under some discussion is the creation of an /opt directory as in SVR4. The goal for such a directory would be to provide a location for large commercial or third party packages to install themselves without worrying about the requirements made by FSSTND for the other directory hierarchies. The FSSTND provides the Linux community with an excellent reference document and has proven to be an important factor in the maturation of Linux. As Linux continues to evolve, so will the FSSTND.

Now, that we have seen how things should be, let's take a look at the real world. As you will see, the implementation of this concept on Linux isn't perfect and since Linux has always attracted individualists who tend to be fairly opinionated, it has been a bone of contention among users for instance which directories certain files should be put into. With the arrival of different distributions, anarchy has once again descended upon us. Some distributions put mount directories for external media into the / directory, others into /mnt. Red Hat based distributions feature the /etc/sysconfig sub−hierarchy for configuration files concerning input and network devices. Other distributions do not have this directory at all and put the appropriate files elsewhere or even use completely different mechanisms to do the same thing. Some distributions put KDE into /opt/, others into /usr.

But even within a given file system hierarchy, there are inconsistencies. For example, even though this was never the intention of the XFree86 group, XFree86 does indeed have its own directory hierarchy.

These problems don't manifest themselves as long as you compile programs yourself. You can adapt configure scripts or Makefiles to your system's configuration or to your preference. It's a different story if you install pre−compiled packages like RPMs though. Often these are not adaptable from one file system hierarchy to another. What's worse: some RPMs might even create their own hierarchy. If you, say, install a KDE RPM from the SuSE Linux distribution on your Mandrake system, the binary will be put into /opt/kde2/bin. And thus it won't work, because Mandrake expects it to be in /usr/bin. There are of course ways to circumvent this problem but the current situation is clearly untenable. Thus, all the leading Linux distributors have joined the Linux Standard Base project, which is attempting to create a common standard for Linux distributions. This isn't easy, since changing the file system hierarchy means a lot of work for distributors so every distributor tries to push a standard which will allow them to keep as much of their own hierarchy as possible. The LSB will also encompass the proposals made by the Filesystem Hierarchy Standard project (FHS, former FSSTND).

1.2. The Root Directory

To comply with the FSSTND the following directories, or symbolic links to directories, are required in /.

```
       /bin       Essential command binaries
       /boot      Static files of the boot loader
       /dev       Device files
       /etc       Host-specific system configuration
       /lib       Essential shared libraries and kernel modules
       /media     Mount point for removeable media
       /mnt       Mount point for mounting a filesystem temporarily
       /opt       Add-on application software packages
       /sbin      Essential system binaries
       /srv       Data for services provided by this system
       /tmp       Temporary files
       /usr       Secondary hierarchy
       /var       Variable data
```

The following directories, or symbolic links to directories, must be in /, if the corresponding subsystem is installed:

```
     / -- the root directory
     /home User home directories (optional)
     /lib<qual> Alternate format essential shared libraries
                     (optional)
     /root Home directory for the root user (optional)
```

Each directory listed above is described in detail in separate subsections further on in this document.

The reference system will be based upon Debian 3.0r0 (Woody), 2.4.18 kernel configured to a Redhat kernel−2.4.18−i686.config file.

Hardware

◊ Intel Celeron 766 Processor
◊ MSI MS−6309 V.2.0 Mainboard
◊ 512MB PQI PC133 SDRAM
◊ 16x Lite−On LTD−165H DVD−ROM
◊ 40x24x10 Sony CRX175A1 CD−RW
◊ NVIDIA RIVA 32MB TNT2 M64
◊ D−Link DFE−530TX 10/100 NIC
◊ Realtek RTL8029(AS) 10 NIC
◊ Lucent Mars2 Linmodem
◊ C−Media CMI8738 PCI Audio Device
◊ Miro DC−30 VIVO
◊ Aopen KF−45A Miditower Case
◊ Acer Accufeel Keyboard
◊ Genius Netscroll+ Mouse
◊ Compaq MV500 Presario Monitor

Software

◊ Windows XP on /dev/hda1
◊ FreeBSD 4.2 on /dev/hda2
◊ Redhat 8.0 on /dev/hda5
◊ Debian 3.0r0 on /dev/hda6
◊ Mandrake 9.1 on /dev/hda7
◊ Swap partition on /dev/hda8

As we all know Linux file system starts with /, the root directory. All other directories are 'children' of this directory. The partition which the root file system resides on is mounted first during boot and the system will not boot if it doesn't find it. On our reference system, the root directory contains the following sub−directories:

bin/ dev/ home/ lost+found/ proc/ sbin/ usr/ cdrom/ opt/ vmlinuz boot/ etc/ lib/ mnt/ root/ tmp/ var/ dvd/ floppy/ initrd/ /tftpboot

In days past it was also the home directory of 'root' but now he has been given his own directory for reasons that will be explained further on in this document.

1.3. /bin

Unlike /sbin, the bin directory contains several useful commands that are of use to both the system administrator as well as non−privileged users. It usually contains the shells like bash, csh, etc.... and commonly used commands like cp, mv, rm, cat, ls. For this reason and in contrast to /usr/bin, the binaries in this directory are considered to be essential. The reason for this is that it contains essential system programs that must be available even if only the partition containing / is mounted. This situation may arise should you need to repair other partitions but have no access to shared directories (ie. you are in single user mode and hence have no network access). It also contains programs which boot scripts may depend on.

Compliance to the FSSTND means that there are no subdirectories in /bin and that the following commands, or symbolic links to commands, are located there.

```
cat      Utility to concatenate files to standard output
chgrp    Utility to change file group ownership
```

```
chmod    Utility to change file access permissions
chown    Utility to change file owner and group
cp       Utility to copy files and directories
date     Utility to print or set the system data and time
dd       Utility to convert and copy a file
df       Utility to report filesystem disk space usage
dmesg    Utility to print or control the kernel message buffer
echo     Utility to display a line of text
false    Utility to do nothing, unsuccessfully
hostname Utility to show or set the system's host name
kill     Utility to send signals to processes
ln       Utility to make links between files
login    Utility to begin a session on the system
ls       Utility to list directory contents
mkdir    Utility to make directories
mknod    Utility to make block or character special files
more     Utility to page through text
mount    Utility to mount a filesystem
mv       Utility to move/rename files
ps       Utility to report process status
pwd      Utility to print name of current working directory
rm       Utility to remove files or directories
rmdir    Utility to remove empty directories
sed      The `sed' stream editor
sh       The Bourne command shell
stty     Utility to change and print terminal line settings
su       Utility to change user ID
sync     Utility to flush filesystem buffers
true     Utility to do nothing, successfully
umount   Utility to unmount file systems
uname    Utility to print system information

If /bin/sh is not a true Bourne shell, it must be a hard or symbolic link to
the real shell command.

The rationale behind this is because sh and bash mightn't necessarily behave
in the same manner. The use of a symbolic link also allows users to easily
see that /bin/sh is not a true Bourne shell.

The [ and test commands must be placed together in either /bin or /usr/bin.

The requirement for the [ and test commands to be included as binaries
(even if implemented internally by the shell) is shared with the POSIX.2
standard.

The following programs, or symbolic links to programs, must be in /bin if the
corresponding subsystem is installed:

csh     The C shell (optional)
ed      The `ed' editor (optional)
tar     The tar archiving utility (optional)
cpio    The cpio archiving utility (optional)
gzip    The GNU compression utility (optional)
gunzip  The GNU uncompression utility (optional)
zcat    The GNU uncompression utility (optional)
netstat The network statistics utility (optional)
ping    The ICMP network test utility (optional)

If the gunzip and zcat programs exist, they must be symbolic or hard links to
gzip. /bin/csh may be a symbolic link to /bin/tcsh or /usr/bin/tcsh.

The tar, gzip and cpio commands have been added to make restoration of a
```

```
system possible (provided that / is intact).

Conversely, if no restoration from the root partition is ever expected,
then these binaries might be omitted (e.g., a ROM chip root, mounting /usr
through NFS). If restoration of a system is planned through the network,
then ftp or tftp (along with everything necessary to get an ftp connection)
must be available on the root partition.
```

1.4. /boot

This directory contains everything required for the boot process except for configuration files not needed at boot time (the most notable of those being those that belong to the GRUB boot−loader) and the map installer. Thus, the /boot directory stores data that is used before the kernel begins executing user−mode programs. This may include redundant (back−up) master boot records, sector/system map files, the kernel and other important boot files and data that is not directly edited by hand. Programs necessary to arrange for the boot loader to be able to boot a file are placed in /sbin. Configuration files for boot loaders are placed in /etc. The system kernel is located in either / or /boot (or as under Debian in /boot but is actually a symbolically linked at / in accordance with the FSSTND).

/boot/boot.0300
: Backup master boot record.

/boot/boot.b
: This is installed as the basic boot sector. In the case of most modern distributions it is actually a symbolic link to one of four files /boot/boot−bmp.b, /boot/boot−menu.b, /boot/boot−text.b, /boot/boot−compat.b which allow a user to change the boot−up schema so that it utilises a splash screen, a simple menu, a text based interface or a minimal boot loader to ensure compatibility respectively. In each case re−installation of lilo is necessary in order to complete the changes. To change the actual 'boot−logo' you can either use utilities such as fblogo or the more refined bootsplash.

/boot/chain.b
: Used to boot non−Linux operating systems.

/boot/config−kernel−version
: Installed kernel configuration. This file is most useful when compiling kernels on other systems or device modules. Below is a small sample of what the contents of the file looks like.

```
        CONFIG_X86=y
        CONFIG_MICROCODE=m
        CONFIG_X86_MSR=m
        CONFIG_MATH_EMULATION=y
        CONFIG_MTRR=y
        CONFIG_MODULES=y
        CONFIG_MODVERSIONS=y
        CONFIG_SCSI_DEBUG=m
        CONFIG_I2O=m
        CONFIG_ARCNET_ETH=y
        CONFIG_FMV18X=m
        CONFIG_HPLAN_PLUS=m
        CONFIG_ETH16I=m
        CONFIG_NE2000=m
        CONFIG_HISAX_HFC_PCI=y
        CONFIG_ISDN_DRV_AVMB1_C4=m
        CONFIG_USB_RIO500=m
        CONFIG_QUOTA=y
        CONFIG_AUTOFS_FS=m
        CONFIG_ADFS_FS=m
```

```
        CONFIG_AFFS_FS=m
        CONFIG_HFS_FS=m
        CONFIG_FAT_FS=y
        CONFIG_MSDOS_FS=y
        CONFIG_UMSDOS_FS=m
        CONFIG_FBCON_VGA=m
        CONFIG_FONT_8x8=y
        CONFIG_FONT_8x16=y
        CONFIG_SOUND=m
        CONFIG_SOUND_CMPCI=m
        CONFIG_AEDSP16=m
```

As you can see, it's rather simplistic. The line begins with the configuration option and whether it's configured as part of the kernel, as a module or not at all. Lines beginning with a # symbol are comments and are not interpreted during processing.

/boot/os2_d.b
: Used to boot to the 0S/2 operating system.

/boot/map
: Contains the location of the kernel.

/boot/vmlinuz, /boot/vmlinuz−kernel−version
: Normally the kernel or symbolic link to the kernel.

/boot/grub
: This subdirectory contains the GRUB configuration files including boot−up images and sounds. GRUB is the GNU GRand Unified Bootloader, a project which intends to solve all bootup problems once and for all. One of the most interesting features, is that you don't have to install a new partition or kernel, you can change all parameters at boot time via the GRUB Console, since it knows about the filesystems.

/boot/grub/device.map
: Maps devices in /dev to those used by grub. For example, (/dev/fd0) is represented by /dev/fd0 and (hd0, 4) is referenced by /dev/hda5.

/boot/grub/grub.conf, /boot/grub/menu.lst
: Grub configuration file.

/boot/grub/messages
: Grub boot−up welcome message.

/boot/grub/splash.xpm.gz
: Grub boot−up background image.

1.5. /dev

/dev is the location of special or device files. It is a very interesting directory that highlights one important aspect of the Linux filesystem − everything is a file or a directory. Look through this directory and you should hopefully see hda1, hda2 etc.... which represent the various partitions on the first master drive of the system. /dev/cdrom and /dev/fd0 represent your CD−ROM drive and your floppy drive. This may seem strange but it will make sense if you compare the characteristics of files to that of your hardware. Both can be read from and written to. Take /dev/dsp, for instance. This file represents your speaker device. Any data written to this file will be re−directed to your speaker. If you try 'cat /boot/vmlinuz > /dev/dsp' (on a properly configured system) you should hear some sound on the speaker. That's the sound of your kernel! A file sent to /dev/lp0 gets printed. Sending data to and reading from /dev/ttyS0 will allow you to communicate with a device attached there − for instance, your modem.

The majority of devices are either block or character devices; however other types of devices exist and can be created. In general, 'block devices' are devices that store or hold data, 'character devices' can be thought of as devices that transmit or transfer data. For example, diskette drives, hard drives and CD−ROM drives are all block devices while serial ports, mice and parallel printer ports are all character devices. There is a naming scheme of sorts but in the vast majority of cases these are completely illogical.

```
total 724
lrwxrwxrwx    1 root     root           13 Sep 28 18:06 MAKEDEV -> /sbin/MAKEDEV
crw-rw----    1 root     audio     14,  14 Oct  7 16:26 admmidi0
crw-rw----    1 root     audio     14,  30 Oct  7 16:26 admmidi1
lrwxrwxrwx    1 root     root           11 Oct  7 16:26 amidi -> /dev/amidi0
crw-rw----    1 root     audio     14,  13 Oct  7 16:26 amidi0
crw-rw----    1 root     audio     14,  29 Oct  7 16:26 amidi1
crw-rw----    1 root     audio     14,  11 Oct  7 16:26 amixer0
crw-rw----    1 root     audio     14,  27 Oct  7 16:26 amixer1
drwxr-xr-x    2 root     root         4096 Sep 28 18:05 ataraid
lrwxrwxrwx    1 root     root           11 Oct  7 16:26 audio -> /dev/audio0
crw-rw----    1 root     audio     14,   4 Oct  7 16:26 audio0
crw-rw----    1 root     audio     14,  20 Oct  7 16:26 audio1
crw-rw----    1 root     audio     14,   7 Mar 15  2002 audioctl
lrwxrwxrwx    1 root     root            9 Oct 14 22:51 cdrom -> /dev/scd1
lrwxrwxrwx    1 root     root            9 Oct 14 22:52 cdrom1 -> /dev/scd0
crw-------    1 root     tty        5,   1 Jan 19 20:47 console
lrwxrwxrwx    1 root     root           11 Sep 28 18:06 core -> /proc/kcore
crw-rw----    1 root     audio     14,  10 Oct  7 16:26 dmfm0
crw-rw----    1 root     audio     14,  26 Oct  7 16:26 dmfm1
crw-rw----    1 root     audio     14,   9 Oct  7 16:26 dmmidi0
crw-rw----    1 root     audio     14,  25 Oct  7 16:26 dmmidi1
lrwxrwxrwx    1 root     root            9 Oct  7 16:26 dsp -> /dev/dsp0
crw-rw----    1 root     audio     14,   3 Oct  7 16:26 dsp0
crw-rw----    1 root     audio     14,  19 Oct  7 16:26 dsp1
crw--w----    1 root     video     29,   0 Mar 15  2002 fb0
crw--w----    1 root     video     29,   1 Mar 15  2002 fb0autodetect
crw--w----    1 root     video     29,   0 Mar 15  2002 fb0current
crw--w----    1 root     video     29,  32 Mar 15  2002 fb1
crw--w----    1 root     video     29,  33 Mar 15  2002 fb1autodetect
crw--w----    1 root     video     29,  32 Mar 15  2002 fb1current
lrwxrwxrwx    1 root     root           13 Sep 28 18:05 fd -> /proc/self/fd
brw-rw----    1 root     floppy     2,   0 Mar 15  2002 fd0
brw-rw----    1 root     floppy     2,   1 Mar 15  2002 fd1
crw--w--w-    1 root     root       1,   7 Sep 28 18:06 full
brw-rw----    1 root     disk       3,   0 Mar 15  2002 hda
brw-rw----    1 root     disk       3,  64 Mar 15  2002 hdb
brw-rw----    1 root     disk      22,   0 Mar 15  2002 hdc
brw-rw----    1 root     disk      22,  64 Mar 15  2002 hdd
drwxr-xr-x    2 root     root        12288 Sep 28 18:05 ida
prw-------    1 root     root            0 Jan 19 20:46 initctl
brw-rw----    1 root     disk       1, 250 Mar 15  2002 initrd
drwxr-xr-x    2 root     root         4096 Sep 28 18:05 input
crw-rw----    1 root     dialout   45, 128 Mar 15  2002 ippp0
crw-rw----    1 root     dialout   45,   0 Mar 15  2002 isdn0
crw-rw----    1 root     dialout   45,  64 Mar 15  2002 isdnctrl0
crw-rw----    1 root     dialout   45, 255 Mar 15  2002 isdninfo
crw-------    1 root     root      10,   4 Mar 15  2002 jbm
crw-r-----    1 root     kmem       1,   2 Sep 28 18:06 kmem
brw-rw----    1 root     cdrom     24,   0 Mar 15  2002 lmscd
crw-------    1 root     root      10,   0 Mar 15  2002 logibm
brw-rw----    1 root     disk       7,   0 Sep 28 18:06 loop0
brw-rw----    1 root     disk       7,   1 Sep 28 18:06 loop1
crw-rw----    1 root     lp         6,   0 Mar 15  2002 lp0
crw-rw----    1 root     lp         6,   1 Mar 15  2002 lp1
crw-rw----    1 root     lp         6,   2 Mar 15  2002 lp2
crw-r-----    1 root     kmem       1,   1 Sep 28 18:06 mem
```

```
lrwxrwxrwx    1 root     root           10 Oct  7 16:26 midi -> /dev/midi0
crw-rw----    1 root     audio     14,   2 Oct  7 16:26 midi0
crw-rw----    1 root     audio     14,  18 Oct  7 16:26 midi1
lrwxrwxrwx    1 root     root           11 Oct  7 16:26 mixer -> /dev/mixer0
crw-rw-rw-    1 root     root      14,   0 Nov 11 16:22 mixer0
crw-rw----    1 root     audio     14,  16 Oct  7 16:26 mixer1
lrwxrwxrwx    1 root     root           11 Oct  7 06:50 modem -> /dev/ttyLT0
crw-rw----    1 root     audio     31,   0 Mar 15  2002 mpu401data
crw-rw----    1 root     audio     31,   1 Mar 15  2002 mpu401stat
crw-rw----    1 root     audio     14,   8 Oct  7 16:26 music
crw-rw-rw-    1 root     root       1,   3 Sep 28 18:06 null
crw-rw-rw-    1 root     root     195,   0 Jan  6 03:03 nvidia0
crw-rw-rw-    1 root     root     195,   1 Jan  6 03:03 nvidia1
crw-rw-rw-    1 root     root     195, 255 Jan  6 03:03 nvidiactl
crw-rw----    1 root     lp         6,   0 Mar 15  2002 par0
crw-rw----    1 root     lp         6,   1 Mar 15  2002 par1
crw-rw----    1 root     lp         6,   2 Mar 15  2002 par2
-rw-r--r--    1 root     root       665509 Oct  7 16:41 pcm
crw-r-----    1 root     kmem       1,   4 Sep 28 18:06 port
crw-rw----    1 root     dip      108,   0 Sep 28 18:07 ppp
crw-------    1 root     root      10,   1 Mar 15  2002 psaux
crw-rw-rw-    1 root     root       1,   8 Sep 28 18:06 random
crw-rw----    1 root     root      10, 135 Mar 15  2002 rtc
brw-rw----    1 root     cdrom     11,   0 Mar 15  2002 scd0
brw-rw----    1 root     cdrom     11,   1 Mar 15  2002 scd1
brw-rw----    1 root     disk       8,   0 Mar 15  2002 sda
brw-rw----    1 root     disk       8,   1 Mar 15  2002 sda1
brw-rw----    1 root     disk       8,   2 Mar 15  2002 sda2
brw-rw----    1 root     disk       8,   3 Mar 15  2002 sda3
brw-rw----    1 root     disk       8,   4 Mar 15  2002 sda4
brw-rw----    1 root     disk       8,  16 Mar 15  2002 sdb
brw-rw----    1 root     disk       8,  17 Mar 15  2002 sdb1
brw-rw----    1 root     disk       8,  18 Mar 15  2002 sdb2
brw-rw----    1 root     disk       8,  19 Mar 15  2002 sdb3
brw-rw----    1 root     disk       8,  20 Mar 15  2002 sdb4
crw-rw----    1 root     audio     14,   1 Oct  7 16:26 sequencer
lrwxrwxrwx    1 root     root           10 Oct  7 16:26 sequencer2 -> /dev/music
lrwxrwxrwx    1 root     root            4 Sep 28 18:05 stderr -> fd/2
lrwxrwxrwx    1 root     root            4 Sep 28 18:05 stdin -> fd/0
lrwxrwxrwx    1 root     root            4 Sep 28 18:05 stdout -> fd/1
crw-rw-rw-    1 root     tty        5,   0 Sep 28 18:06 tty
crw-------    1 root     root       4,   0 Sep 28 18:06 tty0
crw-------    1 root     root       4,   1 Jan 19 14:59 tty1
crw-rw----    1 root     dialout   62,  64 Oct  7 06:50 ttyLT0
crw-rw----    1 root     dialout    4,  64 Mar 15  2002 ttyS0
crw-rw----    1 root     dialout    4,  65 Mar 15  2002 ttyS1
crw-rw----    1 root     dialout    4,  66 Mar 15  2002 ttyS2
crw-rw----    1 root     dialout    4,  67 Mar 15  2002 ttyS3
crw-rw----    1 root     dialout  188,   0 Mar 15  2002 ttyUSB0
crw-rw----    1 root     dialout  188,   1 Mar 15  2002 ttyUSB1
cr--r--r--    1 root     root       1,   9 Jan 19 20:46 urandom
drwxr-xr-x    2 root     root         4096 Sep 28 18:05 usb
prw-r-----    1 root     adm             0 Jan 19 14:58 xconsole
crw-rw-rw-    1 root     root       1,   5 Sep 28 18:06 zero
```

Some common device files as well as their equivalent counterparts under Windows that you may wish to remember are:

/dev/ttyS0 (First communications port, COM1)
: First serial port (mice, modems).

/dev/psaux (PS/2)

PS/2 mouse connection (mice, keyboards).

/dev/lp0 (First printer port, LPT1)
: First parallel port (printers, scanners, etc).

/dev/dsp (First audio device)
: The name DSP comes from the term digital signal processor, a specialized processor chip optimized for digital signal analysis. Sound cards may use a dedicated DSP chip, or may implement the functions with a number of discrete devices. Other terms that may be used for this device are digitized voice and PCM.

/dev/usb (USB Devices)
: This subdirectory contains most of the USB device nodes. Device name allocations are fairly simplistic so no elaboration is be necessary.

/dev/sda (C:\, SCSI device)
: First SCSI device (HDD, Memory Sticks, external mass storage devices such as CD−ROM drives on laptops, etc).

/dev/scd (D:\, SCSI CD−ROM device)
: First SCSI CD−ROM device.

/dev/js0 (Standard gameport joystick)
: First joystick device.

Devices are defined by type, such as 'block' or 'character', and 'major' and 'minor' number. The major number is used to categorize a device and the minor number is used to identify a specific device type. For example, all IDE device connected to the primary controller have a major number of 3. Master and slave devices, as well as individual partitions are further defined by the use of minor numbers. These are the two numbers precede the date in the following display:

ls −l /dev/hd*

```
brw-rw----    1 root     disk       3,   0 Mar 15  2002 /dev/hda
brw-rw----    1 root     disk       3,   1 Mar 15  2002 /dev/hda1
brw-rw----    1 root     disk       3,  10 Mar 15  2002 /dev/hda10
brw-rw----    1 root     disk       3,  11 Mar 15  2002 /dev/hda11
brw-rw----    1 root     disk       3,  12 Mar 15  2002 /dev/hda12
brw-rw----    1 root     disk       3,  13 Mar 15  2002 /dev/hda13
brw-rw----    1 root     disk       3,  14 Mar 15  2002 /dev/hda14
brw-rw----    1 root     disk       3,  15 Mar 15  2002 /dev/hda15
brw-rw----    1 root     disk       3,  16 Mar 15  2002 /dev/hda16
brw-rw----    1 root     disk       3,  17 Mar 15  2002 /dev/hda17
brw-rw----    1 root     disk       3,  18 Mar 15  2002 /dev/hda18
brw-rw----    1 root     disk       3,  19 Mar 15  2002 /dev/hda19
brw-rw----    1 root     disk       3,   2 Mar 15  2002 /dev/hda2
brw-rw----    1 root     disk       3,  20 Mar 15  2002 /dev/hda20
brw-rw----    1 root     disk       3,   3 Mar 15  2002 /dev/hda3
brw-rw----    1 root     disk       3,   4 Mar 15  2002 /dev/hda4
brw-rw----    1 root     disk       3,   5 Mar 15  2002 /dev/hda5
brw-rw----    1 root     disk       3,   6 Mar 15  2002 /dev/hda6
brw-rw----    1 root     disk       3,   7 Mar 15  2002 /dev/hda7
brw-rw----    1 root     disk       3,   8 Mar 15  2002 /dev/hda8
brw-rw----    1 root     disk       3,   9 Mar 15  2002 /dev/hda9
brw-rw----    1 root     disk       3,  64 Mar 15  2002 /dev/hdb
brw-rw----    1 root     disk       3,  65 Mar 15  2002 /dev/hdb1
brw-rw----    1 root     disk       3,  74 Mar 15  2002 /dev/hdb10
brw-rw----    1 root     disk       3,  75 Mar 15  2002 /dev/hdb11
brw-rw----    1 root     disk       3,  76 Mar 15  2002 /dev/hdb12
brw-rw----    1 root     disk       3,  77 Mar 15  2002 /dev/hdb13
brw-rw----    1 root     disk       3,  78 Mar 15  2002 /dev/hdb14
brw-rw----    1 root     disk       3,  79 Mar 15  2002 /dev/hdb15
```

```
brw-rw----    1 root     disk       3,  80 Mar 15  2002 /dev/hdb16
brw-rw----    1 root     disk       3,  81 Mar 15  2002 /dev/hdb17
brw-rw----    1 root     disk       3,  82 Mar 15  2002 /dev/hdb18
brw-rw----    1 root     disk       3,  83 Mar 15  2002 /dev/hdb19
brw-rw----    1 root     disk       3,  66 Mar 15  2002 /dev/hdb2
brw-rw----    1 root     disk       3,  84 Mar 15  2002 /dev/hdb20
brw-rw----    1 root     disk       3,  67 Mar 15  2002 /dev/hdb3
brw-rw----    1 root     disk       3,  68 Mar 15  2002 /dev/hdb4
brw-rw----    1 root     disk       3,  69 Mar 15  2002 /dev/hdb5
brw-rw----    1 root     disk       3,  70 Mar 15  2002 /dev/hdb6
brw-rw----    1 root     disk       3,  71 Mar 15  2002 /dev/hdb7
brw-rw----    1 root     disk       3,  72 Mar 15  2002 /dev/hdb8
brw-rw----    1 root     disk       3,  73 Mar 15  2002 /dev/hdb9
brw-rw----    1 root     disk      22,   0 Mar 15  2002 /dev/hdc
brw-rw----    1 root     disk      22,  64 Mar 15  2002 /dev/hdd
```

The major number for both hda and hdb devices is 3. Of course, the minor number changes for each specific partition. The definition of each major number category can be examined by looking at the contents of the /usr/src/linux/include/linux/major.h file. The devices.txt also documents major and minor numbers. It is located in the /usr/src/linux/Documentation directory. This file defines the major numbers. Almost all files devices are created by default at the install time. However, you can always create a device using the mknod command or the MAKEDEV script which is located in the /dev directory itself. Devices can be created with this utility by supplying the device to be created, the device type (block or character) and the major and minor numbers. For example, let's say you have accidentally deleted /dev/ttyS0 (COM1 under Windows), it can be recreated using the following command

mknod ttyS0 c 4 64

For those of us who are rather lazy you can simply run the MAKEDEV script as such

MAKEDEV *

which will create all devices known.

If is possible that /dev may also contain a MAKEDEV.local for the creation of any local device files.

In general and as required by the FSSTND, MAKEDEV will have provisions for creating any device that may be found on the system, not just those that a particular implementation installs.

For those of you who are wondering why Linux is using such a primitive system to reference devices its because we haven't been able to devise a sufficiently sophisticated mechanism which provides enough advantages over the current system in order to achieve widespread adoption.

To date (as of kernel version 2.4), the best attempt has been made by Richard Gooch of the CSIRO. It's called devfsd and has been a part of the kernel for a number of years now. It has been sanctioned by the kernel developers and Linus himself and details of its implementation can be found at /usr/src/linux/Documentation/filesystems/devfs/README. Below is an excerpt from this document.

Devfs is an alternative to "real" character and block special devices on your root filesystem. Kernel device drivers can register devices by name rather than major and minor numbers. These devices will appear in devfs automatically, with whatever default ownership and protection the driver specified. A daemon (devfsd) can be used to override these defaults. Devfs has been in the kernel since 2.3.46.

NOTE that devfs is entirely optional. If you prefer the old disc−based device nodes, then simply leave CONFIG_DEVFS_FS=n (the default). In this case, nothing will change. ALSO NOTE that if you do enable devfs, the defaults are such that full compatibility is maintained with the old devices names.

There are two aspects to devfs: one is the underlying device namespace, which is a namespace just like any mounted filesystem. The other aspect is the filesystem code which provides a view of the device namespace. The reason I make a distinction is because devfs can be mounted many times, with each mount showing the same device namespace. Changes made are global to all mounted devfs filesystems. Also, because the devfs namespace exists without any devfs mounts, you can easily mount the root filesystem by referring to an entry in the devfs namespace.

The cost of devfs is a small increase in kernel code size and memory usage. About 7 pages of code (some of that in __init sections) and 72 bytes for each entry in the namespace. A modest system has only a couple of hundred device entries, so this costs a few more pages. Compare this with the suggestion to put /dev on a ramdisc.

On a typical machine, the cost is under 0.2 percent. On a modest system with 64 MBytes of RAM, the cost is under 0.1 percent. The accusations of "bloatware" levelled at devfs are not justified.

As of kernel version 2.6, devfs has been marked obsolete and has now been replaced by udev. A system very similar (at least from a the end user's point of view) to devfs but which works entirely in userspace. An overview of udev can be found at
http://www.kroah.com/linux/talks/ols_2003_udev_paper/Reprint−Kroah−Hartman−OLS2003.pdf

---

1.6. /etc

This is the nerve center of your system, it contains all system related configuration files in here or in its sub−directories. A "configuration file" is defined as a local file used to control the operation of a program; it must be static and cannot be an executable binary. For this reason, it's a good idea to backup this directory regularly. It will definitely save you a lot of re−configuration later if you re−install or lose your current installation. Normally, no binaries should be or are located here.

/etc/X11/
: This directory tree contains all the configuration files for the X Window System. Users should note that many of the files located in this directory are actually symbolic links to the /usr/X11R6 directory tree. Thus, the presence of these files in these locations can not be certain.

/etc/X11/XF86Config, /etc/X11/XF86Config−4
: The 'X' configuration file. Most modern distributions possess hardware autodetection systems that enable automatic creation of a valid file. Should autodetection fail a configuration file can also be created manually provided that you have sufficient knowledge about your system. It would be considered prudent not to attempt to type out a file from beginning to end. Rather, use common configuration utilities such as xf86config, XF86Setup and xf86cfg to create a workable template. Then, using suitable documentation commence optimization through intuition and/or trial and error. Options that can be configured via this file include X modules to be loaded on startup, keyboard, mouse, monitor and graphic chipset type. Often, commercial distributions will include their own X configuration utilities such as XDrake on Mandrake and also Xconfiguration on Redhat. Below is a sample X configuration file from the reference system

```
### BEGIN DEBCONF SECTION
# XF86Config-4 (XFree86 server configuration file) generated by dexconf, the
```

```
# Debian X Configuration tool, using values from the debconf database.
#
# Edit this file with caution, and see the XF86Config-4 manual page.
# (Type "man XF86Config-4" at the shell prompt.)
#
# If you want your changes to this file preserved by dexconf, only
# make changes
# before the "### BEGIN DEBCONF SECTION" line above, and/or after the
# "### END DEBCONF SECTION" line below.
#
# To change things within the debconf section, run the command:
#   dpkg-reconfigure xserver-xfree86
# as root.  Also see "How do I add custom sections to a dexconf-
# generated
# XF86Config or XF86Config-4 file?" in /usr/share/doc/xfree86-
# common/FAQ.gz.

Section "Files"
        FontPath        "unix/:7100"
# local font server
        # if the local font server has problems,
# we can fall back on these
        FontPath        "/usr/lib/X11/fonts/misc"
        FontPath        "/usr/lib/X11/fonts/cyrillic"
        FontPath        "/usr/lib/X11/fonts/100dpi/:unscaled"
        FontPath        "/usr/lib/X11/fonts/75dpi/:unscaled"
        FontPath        "/usr/lib/X11/fonts/Type1"
        FontPath        "/usr/lib/X11/fonts/Speedo"
        FontPath        "/usr/lib/X11/fonts/100dpi"
        FontPath        "/usr/lib/X11/fonts/75dpi"
EndSection

Section "Module"
        Load        "GLcore"
        Load        "bitmap"
        Load        "dbe"
        Load        "ddc"
        Load        "dri"
        Load        "extmod"
        Load        "freetype"
        Load        "glx"
        Load        "int10"
        Load        "pex5"
        Load        "record"
        Load        "speedo"
        Load        "type1"
        Load        "vbe"
        Load        "xie"
EndSection

Section "InputDevice"
        Identifier        "Generic Keyboard"
        Driver                "keyboard"
        Option                "CoreKeyboard"
        Option                "XkbRules"        "xfree86"
        Option                "XkbModel"        "pc104"
        Option                "XkbLayout"        "us"
EndSection

Section "InputDevice"
        Identifier        "Configured Mouse"
        Driver                "mouse"
```

```
        Option              "CorePointer"
        Option              "Device"                "/dev/psaux"
        Option              "Protocol"              "NetMousePS/2"
        Option              "Emulate3Buttons"       "true"
        Option              "ZAxisMapping"          "4 5"
EndSection

Section "InputDevice"
        Identifier      "Generic Mouse"
        Driver              "mouse"
        Option              "SendCoreEvents"        "true"
        Option              "Device"                "/dev/input/mice"
        Option              "Protocol"              "ImPS/2"
        Option              "Emulate3Buttons"       "true"
        Option              "ZAxisMapping"          "4 5"
EndSection

Section "Device"
        Identifier      "Generic Video Card"
        Driver              "nv"
#        Option              "UseFBDev"              "true"
        Option              "UseFBDev"              "false"
EndSection

Section "Monitor"
        Identifier      "Generic Monitor"
        HorizSync       30-38
        VertRefresh       43-95
        Option              "DPMS"
EndSection

Section "Screen"
        Identifier      "Default Screen"
        Device              "Generic Video Card"
        Monitor              "Generic Monitor"
        DefaultDepth        16
        SubSection "Display"
                Depth               1
                Modes               "800x600" "640x480"
        EndSubSection
        SubSection "Display"
                Depth               4
                Modes               "800x600" "640x480"
        EndSubSection
        SubSection "Display"
                Depth               8
                Modes               "800x600" "640x480"
        EndSubSection
        SubSection "Display"
                Depth               15
                Modes               "800x600" "640x480"
        EndSubSection
        SubSection "Display"
                Depth               16
                Modes               "800x600" "640x480"
        EndSubSection
        SubSection "Display"
                Depth               24
                Modes               "800x600" "640x480"
        EndSubSection
EndSection
```

```
Section "ServerLayout"
        Identifier        "Default Layout"
        Screen               "Default Screen"
        InputDevice        "Generic Keyboard"
        InputDevice        "Configured Mouse"
        InputDevice        "Generic Mouse"
EndSection

Section "DRI"
        Mode        0666
EndSection

### END DEBCONF SECTION
```

As you can see, the layout of the file is quite simple and tends to be quite standard across most distributions. At the top are the locations of the various font files for X (note − X will not start if you do not specify a valid font), next is the "Modules" section. It details what modules are to be loaded upon startup. The most well known extensions are probably GLX (required for 3D rendering of graphics and games) and Xinerama which allows users to expand their desktop over several monitors. Next are the various "Device" sections which describe the type of hardware you have. Improper configuration of these subsections can lead to heartache and trauma with seemingly misplaced keys, bewitched mice and also constant flashing as X attempts to restart in a sometimes never ending loop. In most cases when all else fails the vesa driver seems to be able to initialise most modern video cards. In the "Screen" section it is possible to alter the default startup resolution and depth. Quite often it is possible to alter these attributes on the fly by using the alt−ctrl−+ or alt−ctrl− set of keystrokes. Lastly are the "ServerLayout" and "DRI" sections. Users will almost never touch the "DRI" section and only those who wish to utilise the Xinerama extensions of X will require having to change any of the ServerLayout options.

/etc/X11/Xmodmap

In general your default keyboard mapping comes from your X server setup. If this setup is insufficient and you are unwilling to go through the process of reconfiguration and/or you are not the superuser you'll need to use the xmodmap program. This is the utility's global configuration file.

/etc/X11/xkb/

The various symbols, types, geometries of keymaps that the X server supports can be found in this directory tree.

/etc/X11/lbxproxy/

Low Bandwidth X (LBX) proxy server configuration files. Applications that would like to take advantage of the Low Bandwidth extension to X (LBX) must make their connections to an lbxproxy. These applications need know nothing about LBX, they simply connect to the lbxproxy as if it were a regular X server. The lbxproxy accepts client connections, multiplexes them over a single connection to the X server, and performs various optimizations on the X protocol to make it faster over low bandwidth and/or high latency connections. It should be noted that such compression will not increase the pace of rendering all that much. Its primary purpose is to reduce network load and thus increase overall network latency. A competing project called DXPC (Differential X Protocol Compression) has been found to be more efficient at this task. Studies have shown though that in almost all cases ssh tunneling of X will produce far better results than through any of these specialised pieces of software.

/etc/X11/proxymngr/

X proxy services manager initialisation files. proxymngr is responsible for resolving requests from xfindproxy (in the xbase−clients package) and other similar clients, starting new proxies when appropriate, and keeping track of all the available proxy services.

/etc/X11/xdm/

X display manager configuration files. xdm manages a collection of X servers, which may be on the local host or remote machines. It provides services similar to those provided by init, getty, and login on character−based terminals: prompting for login name and password, authenticating the user, and running a session. xdm supports XDMCP (X Display Manager Control Protocol) and can also be used to run a chooser process which presents the user with a menu of possible hosts that offer XDMCP display management. If the xutils package is installed, xdm can use the sessreg utility to register login sessions to the system utmp file; this, however, is not necessary for xdm to function.

/etc/X11/xdm/xdm−config
: This is the master 'xdm' configuration file. It determines where all other 'xdm' configuration files will be located. It is almost certain to be left undisturbed.

/etc/X11/gdm/
: GNOME Display Manager configuration files. gdm provides the equivalent of a "login:" prompt for X displays− it pops up a login window and starts an X session. It provides all the functionality of xdm, including XDMCP support for managing remote displays. The greeting window is written using the GNOME libraries and hence looks like a GNOME application− even to the extent of supporting themes! By default, the greeter is run as an unprivileged user for security.

/etc/X11/gdm/gdm.conf
: This is the primary configuration file for GDM. Through it, users can specify whether they would like their system to automatically login as a certain user, background startup image and also if they would like to run their machine as somewhat of a terminal server by using the XDMCP protocol.

/etc/X11/fonts
: Home of xfs fonts.

/etc/X11/fs/
: X font server configuration files. xfs is a daemon that listens on a network port and serves X fonts to X servers (and thus to X clients). All X servers have the ability to serve locally installed fonts for themselves, but xfs makes it possible to offload that job from the X server, and/or have a central repository of fonts on a networked machine running xfs so that all the machines running X servers on a network do not require their own set of fonts. xfs may also be invoked by users to, for instance, make available X fonts in user accounts that are not available to the X server or to an already running system xfs.

/etc/X11/fs/config
: This is the 'xfs' initialisation file. It specifies the number of clients that are allowed to connect to the 'xfs' server at any one time, the location of log files, default resolution, the location of the fonts, etc.

```
# font server configuration file
# $Xorg: config.cpp,v 1.3 2000/08/17 19:54:19 cpqbld Exp $

# allow a maximum of 10 clients to connect to this font server
client-limit = 10
# when a font server reaches its limit, start up a new one
clone-self = on
# log messages to /var/log/xfs.log (if syslog is not used)
error-file = /var/log/xfs.log
# log errors using syslog
use-syslog = on
# turn off TCP port listening (Unix domain connections are still permitted)
no-listen = tcp
# paths to search for fonts
catalogue = /usr/lib/X11/fonts/misc/,/usr/lib/X11/fonts/cyrillic/,
/usr/lib/X11/fonts/100dpi/:unscaled,/usr/lib/X11/fonts/75dpi/:unscaled,
/usr/lib/X11/fonts/Type1/,/usr/lib/X11/fonts/CID,
/usr/lib/X11/fonts/Speedo/,/usr/lib/X11/fonts/100dpi/,
/usr/lib/X11/fonts/75dpi/
# in decipoints
default-point-size = 120
# x1,y1,x2,y2,...
default-resolutions = 100,100,75,75
```

```
# font cache control, specified in kB
cache-hi-mark = 2048
cache-low-mark = 1433
cache-balance = 70
```

/etc/X11/twm
: Home of configuration files for twm. The original Tabbed Window Manager.

/etc/X11/xinit/
: xinit configuration files. 'xinit' is a configuration method of starting up an X session that is designed to used as part of a script. Normally, this is used at larger sites as part of a tailored login process.

/etc/X11/xinit/xinitrc
: Global xinitrc file, used by all X sessions started by xinit (startx). Its usage is of course overridden by a .xinitrc file located in the home directory of a user.

/etc/adduser.conf
: 'adduser' configuration. The adduser command can create new users, groups and add existing users to existing groups. Adding users with adduser is much easier than adding them by hand. Adduser will choose appropriate UID and GID values, create a home directory, copy skeletal user configuration from /etc/skel, allow you to set an initial password and the GECOS field. Optionally a custom script can be executed after this commands. See adduser(8) and adduser.conf(5) for full documentation.

/etc/adjtime
: Has parameters to help adjust the software (kernel) time so that it matches the RTC.

/etc/aliases
: This is the aliases file – it says who gets mail for whom. It was originally generated by `eximconfig', part of the exim package distributed with Debian, but it may edited by the mail system administrator. See exim info section for details of the things that can be configured here. An aliases database file (aliases.db) is built from the entries in the aliases files by the newaliases utility.

/etc/alternatives
: It is possible for several programs fulfilling the same or similar functions to be installed on a single system at the same time. For example, many systems have several text editors installed at once. This gives choice to the users of a system, allowing each to use a different editor, if desired, but makes it difficult for a program to make a good choice of editor to invoke if the user has not specified a particular preference.

  The alternatives system aims to solve this problem. A generic name in the filesystem is shared by all files providing interchangeable functionality. The alternatives system and the system administrator together determine which actual file is referenced by this generic name. For example, if the text editors ed(1) and nvi(1) are both installed on the system, the alternatives system will cause the generic name /usr/bin/editor to refer to /usr/bin/nvi by default. The system administrator can override this and cause it to refer to /usr/bin/ed instead, and the alternatives system will not alter this setting until explicitly requested to do so.

  The generic name is not a direct symbolic link to the selected alternative. Instead, it is a symbolic link to a name in the alternatives directory, which in turn is a symbolic link to the actual file referenced. This is done so that the system administrator's changes can be confined within the /etc directory.

/etc/apt
: This is Debian's next generation front–end for the dpkg package manager. It provides the apt–get utility and APT dselect method that provides a simpler, safer way to install and upgrade packages. APT features complete installation ordering, multiple source capability and several other unique features, see the Users Guide in /usr/share/doc/apt/guide.text.gz

/etc/apt/sources.list

```
deb cdrom:[Debian GNU/Linux 3.0 r0 _Woody_ - Official i386 Binary-7 (20020718)]/
          unstable contrib main non-US/contrib non-US/main
```

```
deb cdrom:[Debian GNU/Linux 3.0 r0 _Woody_ - Official i386 Binary-6 (20020718)]/
          unstable contrib main non-US/contrib non-US/main
deb cdrom:[Debian GNU/Linux 3.0 r0 _Woody_ - Official i386 Binary-5 (20020718)]/
          unstable contrib main non-US/contrib non-US/main
deb cdrom:[Debian GNU/Linux 3.0 r0 _Woody_ - Official i386 Binary-4 (20020718)]/
          unstable contrib main non-US/contrib non-US/main
deb cdrom:[Debian GNU/Linux 3.0 r0 _Woody_ - Official i386 Binary-3 (20020718)]/
          unstable contrib main non-US/contrib non-US/main
deb cdrom:[Debian GNU/Linux 3.0 r0 _Woody_ - Official i386 Binary-2 (20020718)]/
          unstable contrib main non-US/contrib non-US/main
deb cdrom:[Debian GNU/Linux 3.0 r0 _Woody_ - Official i386 Binary-1 (20020718)]/
          unstable contrib main non-US/contrib non-US/main

# deb http://security.debian.org/ stable/updates main
```

Contains a list of apt−sources from which packages may be installed via APT.

/etc/asound.conf

ALSA (Advanced Linux Sound Architecture) configuration file. It is normally created via alsactl or other third−party sound configuration utilities that may be specific to a distribution such as sndconfig from Redhat.

/etc/at.deny

Users denied access to the at daemon. The 'at' command allows user to execute programs at an arbitrary time.

/etc/autoconf

Configuration files for autoconf. 'autoconf' creates scripts to configure source code packages using templates. To create configure from configure.in, run the autoconf program with no arguments. autoconf processes configure.ac with the m4 macro processor, using the Autoconf macros. If you give autoconf an argument, it reads that file instead of configure.ac and writes the configuration script to the standard output instead of to configure. If you give autoconf the argument −, it reads the standard input instead of configure.ac and writes the configuration script on the standard output.

The Autoconf macros are defined in several files. Some of the files are distributed with Autoconf; autoconf reads them first. Then it looks for the optional file acsite.m4 in the directory that contains the distributed Autoconf macro files, and for the optional file aclocal.m4 in the current directory. Those files can contain your site's or the package's own Autoconf macro definitions. If a macro is defined in more than one of the files that autoconf reads, the last definition it reads overrides the earlier ones.

/etc/bash.bashrc

System wide functions and aliases' file for interactive bash shells.

/etc/bash_completion

Programmable completion functions for bash 2.05a.

/etc/chatscripts/provider

This is the chat script used to dial out to your default service provider.

/etc/cron.d, /etc/cron.daily, /etc/cron.weekly, /etc/cron.monthly

These directories contain scripts to be executed on a regular basis by the cron daemon.

/etc/crontab

'cron' configuration file. This file is for the cron table to setup the automatic running of system routines. A cron table can also be established for individual users. The location of these user cron table files will be explained later on.

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file.
# This file also has a username field, that none of the other crontabs do.

SHELL=/bin/sh
```

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user command
25 6 * * * root test -e /usr/sbin/anacron || run-parts --report /etc/cron.daily
47 6 * * 7 root test -e /usr/sbin/anacron || run-parts --report /etc/cron.weekly
52 6 1 * * root test -e /usr/sbin/anacron || run-parts --report /etc/cron.monthly
#
```

/etc/csh.login
: System−wide .login file for csh(1). This file is sourced on all invocations of the shell. It contains commands that are to be executed upon login and sometimes aliases also.

/etc/csh.logout
: System−wide .logout file for csh(1). This file is sourced on all invocations of the shell. It contains commands that are to be executed upon logout.

/etc/csh.cshrc
: System−wide .cshrc file for csh(1). This file is sourced on all invocations of the shell. This file should contain commands to set the command search path, plus other important environment variables. This file should not contain commands that produce output or assume the shell is attached to a tty.

/etc/cups
: Configuration files for the Common UNIX Printing System (CUPS). Files here are used to define client−specific parameters, such as the default server or default encryption settings.

/etc/deluser.conf
: 'deluser' configuration files. The deluser command can remove users and groups and remove users from a given group. Deluser can optionally remove and backup the user's home directory and mail spool or all files on the system owned by him. Optionally a custom script can also be executed after each of the commands.

/etc/devfs
: This daemon sets up the /dev filesystem for use. It creates required symbolic links in /dev and also creates (if so configured, as is the default) symbolic links to the "old" names for devices.

/etc/devfs/conf.d/
: 'devfsd' configuration files. This daemon sets up the /dev filesystem for use. It creates required symbolic links in /dev and also creates (if so configured, as is the default) symbolic links to the "old" names for devices.

/etc/dhclient.conf, /etc/dhclient−script
: 'dhclient' configuration file and 'dhclient' script files respectively. It configures your system so that it may act as a client on a DHCP based network. It is essential to connect to the Internet nowadays.

/etc/dict.conf

```
#  /etc/dict.conf Written by Bob Hilliard <hilliard@debian.org>
#  1998/03/20.  Last revised Sun, 22 Nov 1998 18:10:04 -0500 This is
#  the configuration file for /usr/bin/dict.  In most cases only the
#  server keyword need be specified.

#  This default configuration will try to access a dictd server on
#  the local host, failing that, it will try the public server.  In
#  many cases this will be slow, so you should comment out the line
#  for the server that you don't want to use. To use any other
#  server, enter its IP address in place of "dict.org".

#  Refer to the dict manpage (man dict) for other options that could
#  be inserted in here.

server localhost
server dict.org
```

dict is a client for the Dictionary Server Protocol (DICT), a TCP transaction based query/response protocol that provides access to dictionary definitions from a set of natural language dictionary databases.

/etc/dosemu.conf
: Configuration file for the Linux DOS Emulator. DOSEMU is a PC Emulator application that allows Linux to run a DOS operating system in a virtual x86 machine. This allows you to run many DOS applications. It includes the FreeDOS kernel, color text and full keyboard emulation (via hotkeys) via terminal, built−in X support, IBM character set font, graphics capability at the console with most compatible video cards, DPMI support so you can run DOOM, CDROM support, builtin IPX and pktdrvr support. Note − 'dosemu' is simply a ported version of Corel's own PC−DOS.

/etc/email−addresses
: Part of the exim package. This file contains email addresses to use for outgoing mail. Any local part not in here will be qualified by the system domain as normal. It should contain lines of the form:

```
user: someone@isp.com
otheruser: someoneelse@anotherisp.com
```

Exim is an MTA that is considered to be rather easier to configure than smail or sendmail. It is a drop−in replacement for sendmail, mailq and rsmtp. Advanced features include the ability to reject connections from known spam sites, and an extremely efficient queue processing algorithm.

/etc/esound.conf
: ESD configuration files. The Enlightened sound daemon is designed to mix together several digitized audio streams for playback by a single device. Like nasd, artsd and rplay it also has the capability to play sounds remotely.

/etc/exports
: The control list of systems who want to access the system via NFS, a the list of directories that you would like to share and the permissions allocated on each share.

```
# /etc/exports: the access control list for filesystems which may be
# exported to NFS clients.  See exports(5).
## LTS-begin ##

#
# The lines between the 'LTS-begin' and the 'LTS-end' were added
# on: Sun Feb 23 05:54:17 EST 2003 by the ltsp installation script.
# For more information, visit the ltsp homepage
# at http://www.ltsp.org
#

/opt/ltsp/i386                  192.168.0.0/255.255.255.0(ro,no_root_squash)
/var/opt/ltsp/swapfiles         192.168.0.0/255.255.255.0(rw,no_root_squash)

#
# The following entries need to be uncommented if you want
# Local App support in ltsp
#
#/home                  192.168.0.0/255.255.255.0(rw,no_root_squash)

## LTS-end ##
```

/etc/fdprm
: Floppy disk parameter table. Describes what different floppy disk formats look like. Used by setfdprm.

/etc/fstab
: The configuration file for 'mount' and now 'supermount'. It lists the filesystems mounted automatically at startup by the mount −a command (in /etc/rc or equivalent startup file). Under Linux, also contains information about swap areas used automatically by swapon −a.

```
# /etc/fstab: static file system information.
#
# The following is an example. Please see fstab(5) for further details.
# Please refer to mount(1) for a complete description of mount options.
#
# Format:
# <file system> <mount point> <type> <options> <dump> <pass>
#
# dump(8) uses the <dump> field to determine which file systems need
# to be dumped. fsck(8) uses the <pass> column to determine which file
# systems need to be checked--the root file system should have a 1 in
# this field, other file systems a 2, and any file systems that should
# not be checked (such as MS-DOS or NFS file systems) a 0.
#
# The `sw' option indicates that the swap partition is to be activated
# with `swapon -a'.
/dev/hda2 none swap sw 0 0
# The `bsdgroups' option indicates that the file system is to be mounted
# with BSD semantics (files inherit the group ownership of the directory
# in which they live). `ro' can be used to mount a file system read-only.
/dev/hda3 / ext2 defaults 0 1
/dev/hda5 /home ext2 defaults 0 2
/dev/hda6 /var ext2 defaults 0 2
/dev/hda7 /usr ext2 defaults,ro 0 2
/dev/hda8 /usr/local ext2 defaults,bsdgroups 0 2
# The `noauto' option indicates that the file system should not be mounted
# with `mount -a'. `user' indicates that normal users are allowed to mount
# the file system.
/dev/cdrom /cdrom iso9660 defaults,noauto,ro,user 0 0
/dev/fd0 /floppy minix defaults,noauto,user 0 0
/dev/fd1 /floppy minix defaults,noauto,user 0 0
# NFS file systems: server:
/export/usr /usr nfs defaults 0 0
# proc file system:
proc /proc proc defaults 0 0
```

/etc/ftpaccess
: Determines who might get ftp–access to your machine.

/etc/ftpchroot
: List of ftp users that need to be chrooted.

/etc/ftpuser
: List of disallowed ftp users.

/etc/gateways
: Lists gateways for 'routed'.

/etc/gettydefs
: Configures console–logins.

/etc/gnome–vfs–mime–magic
: MIME magic patterns as used by the Gnome VFS library.

/etc/group
: Similar to /etc/passwd. It lists the configured user groups and who belongs to them.

/etc/group–
: Old /etc/group file.

/etc/gshadow
: Contains encrypted forms of group passwords.

/etc/gshadow–
: Old /etc/gshadow file.

/etc/hostname

Contains the hostname of your machine (can be fully qualified or not).

/etc/host.conf

Determines the search order for look−ups (usually hosts bind, i.e. "check /etc/hosts first and then look for a DNS").

/etc/hosts

This file is used to define a system name and domain combination with a specific IP address. This file needs to always contain an entry for an IP address, if the machine is connected to the network.

```
### etherconf DEBCONF AREA. DO NOT EDIT THIS AREA OR INSERT TEXT BEFORE IT.
127.0.0.1 localhost ::1 localhost
ip6−localhost ip6−loopback
fe00::0 ip6−localnet
ff00::0 ip6−mcastprefix
ff02::1 ip6−allnodes
ff02::2 ip6−allrouters
ff02::3 ip6−allhosts
192.168.0.99 debian.localdomain.com debian
### END OF DEBCONF AREA. PLACE YOUR EDITS BELOW; THEY WILL BE PRESERVED.
192.168.0.1 ws001
```

/etc/hosts.allow

Part of the tcp−wrappers system to control access to your machine's services. It lists hosts that are allowed to access the system and specific daemons.

```
# /etc/hosts.allow: list of hosts that are allowed to access the
# system.
# See the manual pages hosts_access(5), hosts_options(5)
# and /usr/doc/netbase/portmapper.txt.gz
#
# Example: ALL: LOCAL @some_netgroup
# ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "portmap"
# for the daemon name. Remember that you can only use the keyword
# "ALL" and IP addresses (NOT host or domain names) for the
# portmapper. See portmap(8) and /usr/doc/portmap/portmapper.txt.gz
# for further information.
bootpd: 0.0.0.0 in.tftpd: 192.168.0.
portmap: 192.168.0.
rpc.mountd: 192.168.0.
rpc.nfsd: 192.168.0.
gdm: 192.168.0.
nasd: 192.168.0.
```

/etc/hosts.deny

part of the tcp−wrappers system to control access to your machine's services. It lists hosts that are not allowed to access the system.

```
# Example: ALL: some.host.name, .some.domain
# ALL EXCEPT in.fingerd: other.host.name, .other.domain
#
# If you're going to protect the portmapper use the name "portmap"
# for the daemon name. Remember that you can only use the keyword
# "ALL" and IP addresses (NOT host or domain names) for the
# portmapper. See portmap(8) and /usr/doc/portmap/portmapper.txt.gz
# for further information.
#
# The PARANOID wildcard matches any host whose name does not match
```

```
# its address. You may wish to enable this to ensure any programs
# that don't validate looked up hostnames still leave understandable
# logs. In past versions of Debian this has been the default.
# ALL: PARANOID
```

/etc/httpd
: Apache configuration files. Apache is a versatile, high−performance HTTP server. The most popular server in the world, Apache features a modular design and supports dynamic selection of extension modules at runtime. Its strong points are its range of possible customization, dynamic adjustment of the number of server processes, and a whole range of available modules including many authentication mechanisms, server−parsed HTML, server−side includes, access control, CERN httpd metafiles emulation, proxy caching, etc. Apache also supports multiple virtual homing.

/etc/identd.conf
: TCP/IP IDENT protocol server. It implements the TCP/IP proposed standard IDENT user identification protocol (RFC 1413). identd operates by looking up specific TCP/IP connections and returning the username of the process owning the connection. It can also return other information besides the username.

```
# /etc/identd.conf - an example configuration file


#-- The syslog facility for error messages
# syslog:facility = daemon


#-- User and group (from passwd database) to run as
server:user = nobody

#-- Override the group id
# server:group = kmem

#-- What port to listen on when started as a daemon or from /etc/inittab
# server:port = 113

#-- The socket backlog limit
# server:backlog = 256

#-- Where to write the file containing our process id
# server:pid-file = "/var/run/identd/identd.pid"

#-- Maximum number of concurrent requests allowed (0 = unlimited)
# server:max-requests = 0

#-- Enable some protocol extensions like "VERSION" or "QUIT"
protocol:extensions = enabled

#-- Allow multiple queries per connection
protocol:multiquery = enabled

#-- Timeout in seconds since connection or last query. Zero = disable
# protocol:timeout = 120

#-- Maximum number of threads doing kernel lookups
# kernel:threads = 8

#-- Maximum number of queued kernel lookup requests
# kernel:buffers = 32

#-- Maximum number of time to retry a kernel lookup in case of failure
```

```
# kernel:attempts = 5



#-- Disable username lookups (only return uid numbers)
# result:uid-only = no

#-- Enable the ".noident" file
# result:noident = enabled

#-- Charset token to return in replies
# result:charset = "US-ASCII"

#-- Opsys token to return in replies
# result:opsys = "UNIX"

#-- Log all request replies to syslog (none == don't)
# result:syslog-level = none


#-- Enable encryption (only available if linked with a DES library)
# result:encrypt = no

#-- Path to the DES key file (only available if linked with a DES library)
# encrypt:key-file = "/usr/local/etc/identd.key"


#-- Include a machine local configuration file
# include = /etc/identd.conf
```

/etc/inetd.conf

Configuration of services that are started by the INETD TCP/IP super server. 'inetd' is now deprecated. 'xinetd' has taken its place. See /etc/xinet.conf for further details.

```
# /etc/inetd.conf:  see inetd(8) for further information.
#
# Internet server configuration database
#
#
# Lines starting with "#:LABEL:" or "#<off>#" should not
# be changed unless you know what you are doing!
#
# If you want to disable an entry so it isn't touched during
# package updates just comment it out with a single '#' character.
#
# Packages should modify this file by using update-inetd(8)
#
# <service_name> <sock_type> <proto>
# <flags> <user> <server_path>
# <args>
#
#:INTERNAL: Internal services
#echo stream  tcp nowait root internal
#echo dgram   udp wait root internal
#chargen stream tcp   nowait root internal
#chargen dgram udp    wait root internal
discard stream tcp nowait root internal
discard dgram udp wait root internal
daytime stream tcp nowait root internal
#daytime dgram udp wait root internal
time stream tcp nowait root internal
```

```
#time dgram udp wait root internal

#:STANDARD: These are standard services.
ftp stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.ftpd
telnet stream tcp nowait telnetd.telnetd /usr/sbin/tcpd
                                        /usr/sbin/in.telnetd

#:MAIL: Mail, news and uucp services.
smtp stream tcp nowait mail /usr/sbin/exim exim -bs
imap2  stream  tcp nowait root /usr/sbin/tcpd /usr/sbin/imapd
imap3  stream  tcp nowait root /usr/sbin/tcpd /usr/sbin/imapd

#:INFO: Info services
ident stream tcp wait identd /usr/sbin/identd identd
finger stream tcp nowait nobody /usr/sbin/tcpd
                                /usr/sbin/in.fingerd

#:BOOT: Tftp service is provided primarily for booting.
#Most sites run this only on machines acting as "boot servers."
tftp dgram udp wait nobody /usr/sbin/tcpd
                           /usr/sbin/in.tftpd -s /tftpboot
```

/etc/init.d

```
Order of scripts run in /etc/rc?.d
==================================

0. Overview.

   All scripts executed by the init system are located in /etc/init.d.
   The directories /etc/rc?.d (? = S, 0 .. 6) contain relative links to
   those scripts. These links are named S<2-digit-number><
   original-name> or K<2-digit-number><original-name>.

   If a scripts has the ".sh" suffix it is a bourne shell script and
   MAY be handled in an optimized manner. The behaviour of executing the
   script in an optimized way will not differ in any way from it being
   forked and executed in the regular way.

   The following runlevels are defined:

   N       System bootup (NONE).
   S       Single user mode (not to be switched to directly)
   0       halt
   1       single user mode
   2 .. 5  multi user mode
   6       reboot

1. Boot.

   When the systems boots, the /etc/init.d/rcS script is executed. It
   in turn executes all the S* scripts in /etc/rcS.d in alphabetical
   (and thus numerical) order. The first argument passed to the
   executed scripts is "start". The runlevel at this point is
   "N" (none).

   Only things that need to be run once to get the system in a consistent
   state are to be run. The rcS.d directory is NOT meant to replace rc.local.
   One should not start daemons in this runlevel unless absolutely
   necessary. Eg, NFS might need the portmapper, so it is OK to start it
   early in the boot process. But this is not the time to start the
```

```
   squid proxy server.

2. Going multiuser.

   After the rcS.d scripts have been executed, init switches to the
   default runlevel as specified in /etc/inittab, usually "2".

   Init then executes the /etc/init.d/rc script which takes care of
   starting the services in /etc/rc2.d.

   Because the previous runlevel is "N" (none) the /etc/rc2.d/KXXxxxx
   scripts will NOT be executed - there is nothing to stop yet,
   the system is busy coming up.

   If for example there is a service that wants to run in runlevel 4
   and ONLY in that level, it will place a KXXxxxx script in
   /etc/rc{2,3,5}.d to stop the service when switching out of runlevel 4.
   We do not need to run that script at this point.

   The /etc.rc2.d/SXXxxxx scripts will be executed in alphabetical
   order, with the first argument set to "start".

3. Switching runlevels.

   When one switches from (for example) runlevel 2 to runlevel 3,
   /etc/init.d/rc will first execute in alphabetical order all K
   scripts for runlevel 3 (/etc/rc3.d/KXXxxxx) with as first argument
   "stop" and then all S scripts for runlevel 3 (/etc/rc3.d/SXXxxxx)
   with as first argument "start".

   As an optimization, a check is made for each "service" to see if
   it was already running in the previous runlevel. If it was, and there
   is no K (stop) script present for it in the new runlevel, there is
   no need to start it a second time so that will not be done.

   On the other hand, if there was a K script present, it is assumed the
   service was stopped on purpose first and so needs to be restarted.

   We MIGHT make the same optimization for stop scripts as well-
   if no S script was present in the previous runlevel, we can assume
   that service was not running and we don't need to stop it either.
   In that case we can remove the "coming from level N" special case
   mentioned above in 2). But right now that has not been implemented.

4. Single user mode.

   Switching to single user mode is done by switching to runlevel 1.
   That will cause all services to be stopped (assuming they all have
   a K script in /etc/rc1.d). The runlevel 1 scripts will then switch
   to runlevel "S" which has no scripts - all it does is spawn
   a shell directly on /dev/console for maintenance.

5. Halt/reboot

   Going to runlevel 0 or 6 will cause the system to be halted or rebooted,
   respectively. For example, if we go to runlevel 6 (reboot) first
   all /etc/rc6.d/KXXxxxx scripts will be executed alphabetically with
   "stop" as the first argument.

   Then the /etc/rc6.d/SXXxxxx scripts will be executed alphabetically
   with "stop" as the first argument as well. The reason is that there
   is nothing to start any more at this point - all scripts that are
```

```
run are meant to bring the system down.

In the future, the /etc/rc6.d/SXXxxxx scripts MIGHT be moved to
/etc/rc6.d/K1XXxxxx for clarity.
```

/etc/inittab
: Boot−time system configuration/initialization script. Tells init how to handle runlevels. It sets the default runlevel. This is run first except when booting in emergency (−b) mode. It also enables a user to startup a getty session on an external device such as the serial ports. To add terminals or dial−in modem lines to a system, just add more lines to /etc/inittab, one for each terminal or dial−in line. For more details, see the manual pages init, inittab, and getty. If a command fails when it starts, and init is configured to restart it, it will use a lot of system resources: init starts it, it fails, init starts it, it fails, and so on. To prevent this, init will keep track of how often it restarts a command, and if the frequency grows to high, it will delay for five minutes before restarting again. /etc/inittab also has some special features that allow init to react to special circumstances. powerwait Allows init to shut the system down, when the power fails. This assumes the use of a UPS, and software that watches the UPS and informs init that the power is off. ctrlaltdel Allows init to reboot the system, when the user presses ctrl−alt−del on the console keyboard. Note that the system administrator can configure the reaction to ctrl−alt−del to be something else instead, e.g., to be ignored, if the system is in a public location. sysinit Command to be run when the system is booted. This command usually cleans up /tmp, for example. The list above is not exhaustive. See your inittab manual page for all possibilities, and for details on how to use the ones above. To set (or reset) initial terminal colours. The following shell script should work for VGA consoles: for n in 1 2 4 5 6 7 8; do setterm −fore yellow −bold on −back blue −store > /dev/tty$n done Substitute your favorite colors, and use /dev/ttyS$n for serial terminals. To make sure they are reset when people log out (if they've been changed) replace the references to getty (or mingetty or uugetty or whatever) in /etc/inittab with references to /sbin/mygetty. #!/bin/sh setterm −fore yellow −bold on −back blue −store > $1 exec /sbin/mingetty $@ An example /etc/inittab is provided below.

```
# /etc/inittab: init(8) configuration.
# $Id: etc.xml,v 1.10 2004/02/03 21:42:57 binh Exp $
# The default runlevel. id:2:initdefault:
# Boot−time system configuration/initialization script.
# This is run first except when booting in emergency (−b) mode.
si::sysinit:/etc/init.d/rcS
# What to do in single−user mode.
~~:S:wait:/sbin/sulogin
# /etc/init.d executes the S and K scripts upon change
# of runlevel.
#
# Runlevel 0 is halt.
# Runlevel 1 is single−user.
# Runlevels 2−5 are multi−user.
# Runlevel 6 is reboot.
l0:0:wait:/etc/init.d/rc 0 l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2 l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4 l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
# Normally not reached, but fallthrough in case of emergency.
z6:6:respawn:/sbin/sulogin
# What to do when CTRL−ALT−DEL is pressed.
ca:12345:ctrlaltdel:/sbin/shutdown −t1 −a −r now
# Action on special keypress (ALT−UpArrow).
#kb::kbrequest:/bin/echo "Keyboard Request
#−−edit /etc/inittab to let this work."
# What to do when the power fails/returns.
pf::powerwait:/etc/init.d/powerfail start
```

```
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop
# /sbin/getty invocations for the runlevels.
#
# The "id" field MUST be the same as the last
# characters of the device (after "tty").
#
# Format:
# <id>:<runlevels>:<action>:<process>
#
# Note that on most Debian systems tty7 is used by the X Window System,
# so if you want to add more getty's go ahead but skip tty7 if you run X.
#
1:2345:respawn:/sbin/getty 38400 tty1 2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3 4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5 6:23:respawn:/sbin/getty 38400 tty6
# Example how to put a getty on a serial line (for a terminal)
#
#T0:23:respawn:/sbin/getty -L ttyS0 9600 vt100
#T1:23:respawn:/sbin/getty -L ttyS1 9600 vt100
# Example how to put a getty on a modem line.
#
#T3:23:respawn:/sbin/mgetty -x0 -s 57600 ttyS3

Undocumented features

The letters A-C can be used to spawn a daemon listed in /etc/inittab. For
example, assuming you want to start getty on a port to receive a call, but
only after receiving a voice call first (and not all the time). Furthermore,
you want to be able to receive a data or a fax call and that when you get
the voice message you'll know which you want. You insert two new lines
in /etc/inittab, each with its own ID, and each with a runlevel such as A
for data and B for fax. When you know which you need, you simply spawn the
appropriate daemon by calling 'telinit A' or 'telinit B'.
The appropriate getty is put on the line until the first call is received.
When the caller terminates the connection, the getty drops because, by
definition, on demand will not respawn. The other two letters, S and Q, are
special. S brings you system to maintenance mode and is the same as changing
state to runlevel 1. The Q is used to tell init to reread inittab. The
/etc/inittab file can be changed as often as required, but will only be read
under certain circumstances: -One of its processes dies (do you need to
respawn another?) -On a powerful signal from a power daemon (or a command
line) -When told to change state by telinit The Q argument tells init to
reread the /etc/inittab file. Even though it is called the System V runlevel
system runlevels 7-9 are legitimate runlevels that can be used if necessary.
The administrator must remember to alter the inittab file though and also to
create the required rc?.d files.
```

/etc/inputrc
: Global inputrc for libreadline. Readline is a function that gets a line from a user and automatically edits it.

/etc/isapnp.conf
: Configuration file for ISA based cards. This standard is virtually redundant in new systems. The 'isapnptools' suite of ISA Plug−And−Play configuration utilities is used to configure such devices. These programs are suitable for all systems, whether or not they include a PnP BIOS. In fact, PnP BIOS adds some complications because it may already activate some cards so that the drivers can find them, and these tools can unconfigure them, or change their settings causing all sorts of nasty effects.

/etc/isdn
: ISDN configuration files.

/etc/issue

Output by getty before the login prompt. Usually contains a short description or welcoming message to the system. The contents are up to the system administrator. Debian GNU/\s 3.0 \n \l

/etc/issue.net
: Presents the welcome screen to users who login remotely to your machine (whereas /etc/issue determines what a local user sees on login). Debian GNU/%s 3.0 %h

/etc/kde
: KDE initialization scripts and KDM configuration.

/etc/kde/kdm
: Location for the K Desktop Manager files. kdm manages a collection of X servers, which may be on the local host or remote machines. It provides services similar to those provided by init, getty, and login on character−based terminals: prompting for login name and password, authenticating the user, and running a session. kdm supports XDMCP (X Display Manager Control Protocol) and can also be used to run a chooser process which presents the user with a menu of possible hosts that offer XDMCP display management.

/etc/kderc
: System wide KDE initialization script. Commands here executed every time the KDE environment is loaded. It's a link to /etc/kde2/system.kdeglobals

```
[Directories]
dir_config=/etc/kde2
dir_html=/usr/share/doc/kde/HTML
dir_cgi=/usr/lib/cgi-bin
dir_apps=/usr/share/applnk
dir_mime=/usr/share/mimelnk
dir_services=/usr/share/services
dir_servicetypes=/usr/share/servicetypes
[General]
TerminalApplication=x-terminal-emulator
```

/etc/ld.so.conf, /etc/ld.so.cache
: /etc/ld.so.conf is a file containing a list of colon, space, tab, newline, or comma separated directories in which to search for libraries. /etc/ld.so.cache containing an ordered list of libraries found in the directories specified in /etc/ld.so.conf. This file is not in human readable format, and is not intended to be edited.
'ldconfig' creates the necessary links and cache (for use by the run−time linker, ld.so) to the most recent shared libraries found in the directories specified on the command line, in the file /etc/ld.so.conf, and in the trusted directories (/usr/lib and /lib). 'ldconfig' checks the header and file names of the libraries it encounters when determining which versions should have their links updated. ldconfig ignores symbolic links when scanning for libraries.
'ldconfig' will attempt to deduce the type of ELF libs (ie. libc5 or libc6/glibc) based on what C libs if any the library was linked against, therefore when making dynamic libraries, it is wise to explicitly link against libc (use −lc).
Some existing libs do not contain enough information to allow the deduction of their type, therefore the /etc/ld.so.conf file format allows the specification of an expected type. This is only used for those ELF libs which we can not work out. The format is like this "dirname=TYPE", where type can be libc4, libc5 or libc6. (This syntax also works on the command line). Spaces are not allowed. Also see the −p option.
Directory names containing an = are no longer legal unless they also have an expected type specifier.
'ldconfig' should normally be run by the super−user as it may require write permission on some root owned directories and files. It is normally run automatically at bootup or manually whenever new shared libraries are installed.

/usr/X11R6/lib
: X libraries.

/usr/local/lib
: Local libraries.

/etc/lilo.conf
: Configuration file for the Linux boot loader 'lilo'. 'lilo' is the original OS loader and can load Linux and others. The 'lilo' package normally contains lilo (the installer) and boot−record−images to install Linux, OS/2, DOS and generic Boot Sectors of other Oses. You can use Lilo to manage your Master Boot Record (with a simple text screen, text menu or colorful splash graphics) or call 'lilo' from other boot−loaders to jump−start the Linux kernel.

```
Prompt #Prompt user to select
OS choice at boot timeout=300  # Amount of time to wait before default OS
                               # started (in ms)
default=Debian4 #Default OS to be loaded
vga=normal #VGA mode
boot=/dev/had #location of MBR
map=/boot/map #location of kernel
install=/boot/boot-bmp.b #File to be installed as boot sector
bitmap=/boot/debian.bmp #LILO boot image
bmp-table=30p,100p,1,10 #Colours
selectable bmp-colors=13,,0,1,,0 #Colours chosen
lba32 #Required on most new systems to overcome
      #1024 cylinder problem
image=/vmlinuz #name of kernel
image label=Debian #a label
read-only #file system to be mounted read only
root=/dev/hda6 #location of root filesystem

image=/boot/bzImage
label=Debian4
read-only
root=/dev/hda6

image=/mnt/redhat/boot/vmlinuz
label=Redhat
initrd=/mnt/redhat/boot/initrd-2.4.18-14.img
read-only
root=/dev/hda5
vga=788
append=" hdc=ide-scsi hdd=ide-scsi"

image=/mnt/mandrake/boot/vmlinuz
label="Mandrake"
root=/dev/hda7
initrd=/mnt/mandrake/boot/initrd.img
append="devfs=mount hdc=ide-scsi
acpi=off quiet"
vga=788
read-only

other=/dev/hda2
table=/dev/hda
loader=/boot/chain.b
label=FBSD
other=/dev/hda1
label=Windows
table=/dev/hda

other=/dev/fd0
```

```
label=floppy unsafe
```

/etc/local.gen
: This file lists locales that you wish to have built. You can find a list of valid supported locales at /usr/share/i18n/SUPPORTED. Other combinations are possible, but may not be well tested. If you change this file, you need to re−run locale−gen.

/etc/locale.alias
: Locale name alias data base.

/etc/login.defs
: Configuration control definitions for the login package. An inordinate number of attributes can be altered via this single file such as the location of mail, delay in seconds after a failed login, enabling display of fail log information, display of unknown username login failures, shell environment variables, etc....

/etc/logrotate.conf
: The logrotate utility is designed to simplify the administration of log files on a system which generates a lot of log files. Logrotate allows for the automatic rotation compression, removal and mailing of log files. Logrotate can be set to handle a log file daily, weekly, monthly or when the log file gets to a certain size. Normally, logrotate runs as a daily cron job.

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    create 0664 root utmp
    rotate 1
}

/var/log/btmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

# system-specific logs may be configured here
```

/etc/ltrace.conf
: Configuration file for ltrace (Library Call Tracer). It tracks runtime library calls in dynamically linked programs. 'ltrace' is a debugging program which runs a specified command until it exits. While the command is executing, ltrace intercepts and records the dynamic library calls which are called by the executed process and the signals received by that process. It can also intercept and print the system calls executed by the program. The program to be traced need not be recompiled for this, so you can

use it on binaries for which you don't have the source handy. You should install ltrace if you need a sysadmin tool for tracking the execution of processes.

/etc/magic
: Magic local data and configuration file for the file(1) command. Contains the descriptions of various file formats based on which file guesses the type of the file. Insert here your local magic data. Format is described in magic(5).

/etc/mail.rc
: Initialization file for 'mail'. 'mail' is an intelligent mail processing system which has a command syntax reminiscent of ed with lines replaced by messages. It's basically a command line version of Microsoft Outlook.

/etc/mailcap
: 'metamail' capabilities file. The mailcap file is read by the metamail program to determine how to display non−text at the local site. The syntax of a mailcap file is quite simple, at least compared to termcap files. Any line that starts with "#" is a comment. Blank lines are ignored. Otherwise, each line defines a single mailcap entry for a single content type. Long lines may be continued by ending them with a backslash character, \. Each individual mailcap entry consists of a content−type specification, a command to execute, and (possibly) a set of optional "flag" values.

/etc/mailcap.order
: The mailcap ordering specifications. The order of entries in the /etc/mailcap file can be altered by editing the /etc/mailcap.order file. Each line of that file specifies a package and an optional mime type. Mailcap entries that match will be placed in the order of this file. Entries that don't match will be placed later.

/etc/mailname
: Mail server hostname. Normally the same as the hostname.

/etc/menu, /etc/menu−methods
: The menu package was inspired by the install−fvwm2−menu program from the old fvwm2 package. However, menu tries to provide a more general interface for menu building. With the update−menus command from this package, no package needs to be modified for every X window manager again, and it provides a unified interface for both text−and X−oriented programs.

  When a package that wants to add something to the menu tree gets installed, it will run update−menus in its postinstall script. Update−menus then reads in all menu files in /etc/menu/ /usr/lib/menu and /usr/lib/menu/default, and stores the menu entries of all installed packages in memory. Once that has been done, it will run the menu−methods in /etc/menu−methods/*, and pipe the information about the menu entries to the menu−methods on stdout, so that the menu−methods can read this. Each Window Manager or other program that wants to have the debian menu tree, will supply a menu−method script in /etc/menu−methods/. This menu−method then knows how to generate the startup−file for that window manager. To facilitate this task for the window−manager maintainers, menu provides a install−menu program. This program can generate the startup files for just about every window manager.

/etc/mgetty+sendfax
: Configuration files for use of mgetty as the interface on the serial port. The mgetty routine special routine has special features for handling things such as dial up connections and fax connections.

/etc/mime.types
: MIME−TYPES and the extensions that represent them. This file is part of the "mime−support" package. Note: Compression schemes like "gzip", "bzip", and "compress" are not actually "mime−types". They are "encodings" and hence must _not_ have entries in this file to map their extensions. The "mime−type" of an encoded file refers to the type of data that has been encoded, not the type of the encoding.

/etc/minicom

'minicom' configuration files. 'minicom' is a communication program which somewhat resembles the shareware program TELIX but is free with source code and runs under most unices. Features include dialling directory with auto−redial, support for UUCP−style lock files on serial devices, a separate script language interpreter, capture to file, multiple users with individual configurations, and more.

/etc/modules

List of modules to be loaded at startup.

```
# /etc/modules: kernel modules to load at boot time.
#
# This file should contain the names of kernel modules that are
# to be loaded at boot time, one per line. Comments begin with
# a "#", and everything on the line after them are ignored.
unix
af_packet
via-rhine
cmpci
ne2k-pci
nvidia
```

/etc/modules.conf

```
### This file is automatically generated by update-modules"
#
# Please do not edit this file directly. If you want to change or add
# anything please take a look at the files in /etc/modutils and read
# the manpage for update-modules.
#
### update-modules: start processing /etc/modutils/0keep
# DO NOT MODIFY THIS FILE!
# This file is not marked as conffile to make sure if you upgrade modutils
# it will be restored in case some modifications have been made.
#
# The keep command is necessary to prevent insmod and friends from ignoring
# the builtin defaults of a path-statement is encountered. Until all other
# packages use the new `add path'-statement this keep-statement is essential
# to keep your system working
keep

### update-modules: end processing /etc/modutils/0keep

### update-modules: start processing /etc/modutils/actions
# Special actions that are needed for some modules

# The BTTV module does not load the tuner module automatically,
# so do that in here
post-install bttv insmod tuner
post-remove bttv rmmod tuner


### update-modules: end processing /etc/modutils/actions

### update-modules: start processing /etc/modutils/aliases
# Aliases to tell insmod/modprobe which modules to use

# Uncomment the network protocols you don't want loaded:
# alias net-pf-1 off          # Unix
# alias net-pf-2 off          # IPv4
# alias net-pf-3 off          # Amateur Radio AX.25
# alias net-pf-4 off          # IPX
# alias net-pf-5 off          # DDP / appletalk
```

```
# alias net-pf-6 off          # Amateur Radio NET/ROM
# alias net-pf-9 off          # X.25
# alias net-pf-10 off         # IPv6
# alias net-pf-11 off         # ROSE / Amateur Radio X.25 PLP
# alias net-pf-19 off         # Acorn Econet

alias char-major-10-175       agpgart
alias char-major-10-200       tun
alias char-major-81   bttv
alias char-major-108  ppp_generic
alias /dev/ppp                ppp_generic
alias tty-ldisc-3     ppp_async
alias tty-ldisc-14    ppp_synctty
alias ppp-compress-21 bsd_comp
alias ppp-compress-24 ppp_deflate
alias ppp-compress-26 ppp_deflate

# Crypto modules (see http://www.kerneli.org/)
alias loop-xfer-gen-0 loop_gen
alias loop-xfer-3     loop_fish2
alias loop-xfer-gen-10        loop_gen
alias cipher-2                des
alias cipher-3                fish2
alias cipher-4                blowfish
alias cipher-6                idea
alias cipher-7                serp6f
alias cipher-8                mars6
alias cipher-11               rc62
alias cipher-15               dfc2
alias cipher-16               rijndael
alias cipher-17               rc5


### update-modules: end processing /etc/modutils/aliases

### update-modules: start processing /etc/modutils/ltmodem-2.4.18
# lt_drivers: autoloading and insertion parameter usage
alias char-major-62 lt_serial
alias /dev/tts/LT0  lt_serial
alias /dev/modem lt_serial
# options lt_modem vendor_id=0x115d device_id=0x0420 Forced=3,0x130,0x2f8
# section for lt_drivers ends

### update-modules: end processing /etc/modutils/ltmodem-2.4.18

### update-modules: start processing /etc/modutils/paths
# This file contains a list of paths that modprobe should scan,
# beside the once that are compiled into the modutils tools
# themselves.


### update-modules: end processing /etc/modutils/paths

### update-modules: start processing /etc/modutils/ppp
alias /dev/ppp          ppp_generic
alias char-major-108    ppp_generic
alias tty-ldisc-3       ppp_async
alias tty-ldisc-14      ppp_synctty
alias ppp-compress-21   bsd_comp
alias ppp-compress-24   ppp_deflate
alias ppp-compress-26   ppp_deflate
```

```
### update-modules: end processing /etc/modutils/ppp

### update-modules: start processing /etc/modutils/setserial
#
# This is what I wanted to do, but logger is in /usr/bin, which isn't
# loaded when the module is first loaded into the kernel at boot time!
#
#post-install serial /etc/init.d/setserial start |
#logger -p daemon.info -t "setserial-module reload"
#pre-remove serial /etc/init.d/setserial stop |
#logger -p daemon.info -t "setserial-module uload"
#
alias /dev/tts          serial
alias /dev/tts/0        serial
alias /dev/tts/1        serial
alias /dev/tts/2        serial
alias /dev/tts/3        serial
post-install serial /etc/init.d/setserial modload > /dev/null 2> /dev/null
pre-remove serial /etc/init.d/setserial modsave  > /dev/null 2> /dev/null

### update-modules: end processing /etc/modutils/setserial

### update-modules: start processing /etc/modutils/arch/i386
alias parport_lowlevel parport_pc
alias char-major-10-144 nvram
alias binfmt-0064 binfmt_aout
alias char-major-10-135 rtc

### update-modules: end processing /etc/modutils/arch/i386
```

/etc/modutils
: These utilities are intended to make a Linux modular kernel manageable for all users, administrators and distribution maintainers.

/etc/mtools
: Debian default mtools configuration file. The mtools series of commands work with MS−DOS files and directories on floppy disks. This allows you to use Linux with MS−DOS formatted diskettes on DOS and Windows systems.

/etc/manpath.conf
: This file is used by the man_db package to configure the man and cat paths. It is also used to provide a manpath for those without one by examining their PATH environment variable. For details see the manpath(5) man page.

/etc/mediaprm
: Was formally named /etc/fdprm. See /etc/fdprm for further details.

/etc/motd
: The message of the day, automatically output after a successful login. Contents are up to the system administrator. Often used for getting information to every user, such as warnings about planned downtimes. Linux debian.localdomain.com 2.4.18 #1 Sat Mar 15 00:17:39 EST 2003 i686 unknown Most of the programs included with the Debian GNU/Linux system are freely redistributable; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

/etc/mtab
: List of currently mounted filesystems. Initially set up by the bootup scripts, and updated automatically by the mount command. Used when a list of mounted filesystems is needed, e.g., by the df command. This file is sometimes a symbolic link to /proc/mounts.

/etc/networks

List of networks that the system is currently located on. For example, 192.168.0.0.

/etc/nsswitch.conf
System Database/Name Service Switch configuration file.

/etc/oss.conf
OSS (Open Sound System) configuration file.

/etc/pam.d/
This directory is the home of the configuration files for PAMs, Pluggable Authentication Modules.

/etc/postfix/
Holds your postfix configuration files. Postfix is now the MTA of choice among Linux distributions. It is sendmail−compatible, offers improved speed over sendmail, ease of administration and security. It was originally developed by IBM and was called the IBM Secure Mailer and is used in many large commercial networks. It is now the de−facto standard.

/etc/ppp/
The place where your dial−up configuration files are placed. More than likely to be created by the text menu based pppconfig or other GUI based ppp configuration utilities such as kppp or gnome−ppp.

/etc/pam.conf
Most programs use a file under the /etc/pam.d/ directory to setup their PAM service modules. This file is and can be used, but is not recommended.

/etc/paper.config
Paper size configuration file.

/etc/papersize
Default papersize.

/etc/passwd
This is the 'old' password file, It is kept for compatibility and contains the user database, with fields giving the username, real name, home directory, encrypted password, and other information about each user. The format is documented in the passwd man(ual) page.

```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:100:sync:/bin:/bin/sync games:x:5:100:games:/usr/games:/bin/sh
man:x:6:100:man:/var/cache/man:/bin/sh lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh proxy:x:13:13:proxy:/bin:/bin/sh
postgres:x:31:32:postgres:/var/lib/postgres:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
operator:x:37:37:Operator:/var:/bin/sh
list:x:38:38:SmartList:/var/list:/bin/sh irc:x:39:39:ircd:/var:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/home:/bin/sh
binh:x:1000:1000:,,,:/home/binh:/bin/bash
identd:x:100:65534::/var/run/identd:/bin/false
sshd:x:101:65534::/var/run/sshd:/bin/false gdm:x:102:101:Gnome Display
Manager:/var/lib/gdm:/bin/false
telnetd:x:103:103::/usr/lib/telnetd:/bin/false
dummy:x:1001:1001:,,,:/home/dummy:/bin/bash
```

/etc/passwd−
Old /etc/passwd file.

/etc/printcap
Printer configuration (capabilities) file. The definition of all system printers, whether local or remote, is stored in this file. Its layout is similar to that of /etc/termcap but it uses a different syntax.

/etc/profile
Files and commands to be executed at login or startup time by the Bourne or C shells. These allow the system administrator to set global defaults for all users.

/etc/profile.d
: Shells scripts to be executed upon login to the Bourne or C shells. These scripts are normally called from the /etc/profile file.

/etc/protocols
: Protocols definitions file. It describes the various DARPA Internet protocols that are available from the TCP/IP subsystem. It should be consulted instead of using the numbers in the ARPA include files or resorting to guesstimation. This file should be left untouched since changes could result in incorrect IP packages.

/etc/pcmcia
: Configuration files for PCMCIA devices. Generally only useful to laptop users.

/etc/reportbug.conf
: Configuration file for reportbug. Reportbug is primarily designed to report bugs in the Debian distribution. By default it creates an e−mail to the Debian bug tracking system at mit@bugs.debian.org with information about the bug. Using the −bts option you can report bugs to other servers also using ddebbugs such as KDE.org. It is similar to bug but has far greater capabilities while still maintaining simplicity.

/etc/rc.boot or /etc/rc?.d
: These directories contain all the files necessary to control system services and configure runlevels. A skeleton file is provided in /etc/init.d/skeleton

/etc/rcS.d
: The scripts in this directory are executed once when booting the system, even when booting directly into single user mode. The files are all symbolic links, the real files are located in /etc/init.d/. For a more general discussion of this technique, see /etc/init.d/README.

/etc/resolv.conf
: Configuration of how DNS is to occur is defined in this file. It tells the name resolver libraries where they need to go to find information not found in the /etc/hosts file. This always has at least one nameserver line, but preferably three. The resolver uses each in turn. More than the first three can be included but anything beyond the first three will be ignored. Two lines that appear in the /etc/resolv.conf file are domain and search. Both of these are mutually exclusive options, and where both show up, the last one wins. Other entries beyond the three discussed here are listed in the man pages but aren't often used.

/etc/rmt
: This is not a mistake. This shell script (/etc/rmt) has been provided for compatibility with other Unix−like systems, some of which have utilities that expect to find (and execute) rmt in the /etc directory on remote systems.

/etc/rpc
: The rpc file contains user readable names that can be used in place of rpc program numbers. Each line has the following information: −name of server for the rpc program −rpc program number −aliases Items are separated by any number of blanks and/or tab characters. A ``#'' indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines which search the file.

```
# /etc/rpc:
# $Id: etc.xml,v 1.10 2004/02/03 21:42:57 binh Exp $
#
# rpc 88/08/01 4.0 RPCSRC; from 1.12   88/02/07 SMI

portmapper    100000  portmap sunrpc
rstatd                100001  rstat rstat_svc rup perfmeter
rusersd               100002  rusers
nfs           100003  nfsprog
ypserv                100004  ypprog
mountd                100005  mount showmount
ypbind                100007
```

```
walld          100008  rwall shutdown
yppasswdd      100009  yppasswd
etherstatd     100010  etherstat
rquotad               100011  rquotaprog quota rquota
sprayd                100012  spray
3270_mapper    100013
rje_mapper     100014
selection_svc  100015  selnsvc
database_svc   100016
rexd           100017  rex
alis           100018
sched          100019
llockmgr       100020
nlockmgr       100021
x25.inr               100022
statmon               100023
status                100024
bootparam      100026
ypupdated      100028  ypupdate
keyserv               100029  keyserver
tfsd           100037
nsed           100038
nsemntd               100039
pcnfsd                150001
amd            300019  amq
sgi_fam               391002
ugidd          545580417
bwnfsd           788585389
```

/etc/samba
: Samba configuration files. A 'LanManager' like file and printer server for Unix. The Samba software suite is a collection of programs that implements the SMB protocol for unix systems, allowing you to serve files and printers to Windows, NT, OS/2 and DOS clients. This protocol is sometimes also referred to as the LanManager or NetBIOS protocol.

/etc/sane.d
: Sane configuration files. SANE stands for "Scanner Access Now Easy" and is an application programming interface (API) that provides standardized access to any raster image scanner hardware (flatbed scanner, hand−held scanner, video− and still−cameras, frame−grabbers, etc.). The SANE API is public domain and its discussion and development is open to everybody. The current source code is written for UNIX (including GNU/Linux) and is available under the GNU General Public License (the SANE API is available to proprietary applications and backends as well, however).

SANE is a universal scanner interface. The value of such a universal interface is that it allows writing just one driver per image acquisition device rather than one driver for each device and application. So, if you have three applications and four devices, traditionally you'd have had to write 12 different programs. With SANE, this number is reduced to seven: the three applications plus the four drivers. Of course, the savings get even bigger as more and more drivers and/or applications are added.

Not only does SANE reduce development time and code duplication, it also raises the level at which applications can work. As such, it will enable applications that were previously unheard of in the UNIX world. While SANE is primarily targeted at a UNIX environment, the standard has been carefully designed to make it possible to implement the API on virtually any hardware or operating system.

While SANE is an acronym for ``Scanner Access Now Easy'' the hope is of course that SANE is indeed sane in the sense that it will allow easy implementation of the API while accommodating all features required by today's scanner hardware and applications. Specifically, SANE should be broad enough to accommodate devices such as scanners, digital still and video cameras, as well as virtual devices like image file filters.

If you're familiar with TWAIN, you may wonder why there is a need for SANE. Simply put, TWAIN does not separate the user−interface from the driver of a device. This, unfortunately, makes it difficult, if not impossible, to provide network transparent access to image acquisition devices (which is useful if you have a LAN full of machines, but scanners connected to only one or two machines; it's obviously also useful for remote−controlled cameras and such). It also means that any particular TWAIN driver is pretty much married to a particular GUI API (be it Win32 or the Mac API). In contrast, SANE cleanly separates device controls from their representation in a user−interface. As a result, SANE has no difficulty supporting command−line driven interfaces or network−transparent scanning. For these reasons, it is unlikely that there will ever be a SANE backend that can talk to a TWAIN driver. The converse is no problem though: it would be pretty straight forward to access SANE devices through a TWAIN source. In summary, if TWAIN had been just a little better designed, there would have been no reason for SANE to exist, but things being the way they are, TWAIN simply isn't SANE.

/etc/securetty
: Identifies secure terminals, i.e., the terminals from which root is allowed to log in. Typically only the virtual consoles are listed, so that it becomes impossible (or at least harder) to gain superuser privileges by breaking into a system over a modem or a network.

```
# /etc/securetty: list of terminals on which root is allowed to login.
# See securetty(5) and login(1).
console

# Standard consoles
tty1
tty2
tty3
tty4
tty5
tty6
tty7
tty8
tty9
tty10
tty11
tty12

# Same as above, but these only occur with devfs devices
vc/1
vc/2
vc/3
vc/4
vc/5
vc/6
vc/7
vc/8
vc/9
vc/10
vc/11
vc/12
```

/etc/sensors.conf

Configuration file for libsensors. A set of libraries designed to ascertain current hardware states via motherboard sensor chips. Useful statistics such as core voltages, CPU temperature can be determined through third party utilities that make user of these libraries such as 'gkrellm'. If you do not wish to install these packages you may also utilise the /proc filesystem real−time nature.

/etc/sudoers
: Sudoers file. This file must be edited with the 'visudo' command as root. The sudo command allows an authenticated user to execute an authorized command as root. Both the effective UID and GID are set to 0 (you are basically root). It determines which users are authorized and which commands they are authorized to use. Configuration of this command is via this file.

/etc/shadow
: Shadow password file on systems with shadow password software installed (PAMs). Shadow passwords move the encrypted password from /etc/passwd into /etc/shadow; the latter is not readable by anyone except root. This makes it more difficult to crack passwords.

/etc/shadow−
: Old /etc/shadow file.

/etc/sysctl.conf
: Configuration file for setting system variables, most notably kernel parameters. 'sysctl' is a means of configuring certain aspects of the kernel at run−time, and the /proc/sys/ directory is there so that you don't even need special tools to do it!

/etc/security
: Essential to security. This subdirectory allows administrators to impose quota limits, access limits and also to configure PAM environments.

/etc/serial.conf
: Serial port configuration. Changeable parameters include speed, baud rate, port, irq and type.

/etc/services
: A definition of the networks, services and the associated port for each protocol that are available on this system. For example, web services (http) are assigned to port 80 by default. # /etc/services: # $Id: etc.xml,v 1.10 2004/02/03 21:42:57 binh Exp $ # # Network services, Internet style # # Note that it is presently the policy of IANA to assign a single # well−known port number for both TCP and UDP; hence, most entries # here have two entries even if the protocol doesn't support UDP # operations. Updated from RFC 1700, ``Assigned Numbers'' (October # 1994). Not all ports are included, only the more common ones. echo 7/tcp echo 7/udp discard 9/tcp sink null discard 9/udp sink null systat 11/tcp users daytime 13/tcp daytime 13/udp netstat 15/tcp qotd 17/tcp quote msp 18/tcp # message send protocol msp 18/udp # message send protocol chargen 19/tcp ttytst source chargen 19/udp ttytst source ftp−data 20/tcp ftp 21/tcp fsp 21/udp fspd ssh 22/tcp # SSH Remote Login Protocol ssh 22/udp # SSH Remote Login Protocol telnet 23/tcp # 24 − private smtp 25/tcp mail # 26 − unassigned time 37/tcp timserver time 37/udp timserver rlp 39/udp resource # resource location nameserver 42/tcp name # IEN 116 whois 43/tcp nicname re−mail−ck 50/tcp # Remote Mail Checking Protocol re−mail−ck 50/udp # Remote Mail Checking Protocol domain 53/tcp nameserver # name−domain server domain 53/udp nameserver netbios−ns 137/tcp # NETBIOS Name Service netbios−ns 137/udp netbios−dgm 138/tcp # NETBIOS Datagram Service netbios−dgm 138/udp netbios−ssn 139/tcp # NETBIOS session service netbios−ssn 139/udp x11 6000/tcp x11−0 # X windows system x11 6000/udp x11−0 # X windows system

/etc/shells
: Lists trusted shells. The chsh command allows users to change their login shell only to shells listed in this file. ftpd, the server process that provides FTP services for a machine, will check that the user's shell is listed in /etc/shells and will not let people log in unless the shell is listed there. There are also some display managers that will passively or actively (dependent upon on distribution and display manager being used) refuse a user access to the system unless their shell is one of those listed here.

```
# /etc/shells: valid login shells
/bin/ash
/bin/bash
/bin/csh
/bin/sh
/usr/bin/es
/usr/bin/ksh
/bin/ksh
/usr/bin/rc
/usr/bin/tcsh
/bin/tcsh
/usr/bin/zsh
/bin/sash
/bin/zsh
/usr/bin/esh
```

/etc/skel/
: The default files for each new user are stored in this directory. Each time a new user is added, these skeleton files are copied into their home directory. An average system would have: .alias, .bash_profile, .bashrc and .cshrc files. Other files are left up to the system administrator.

/etc/sysconfig/
: This directory contains configuration files and subdirectories for the setup of system configuration specifics and for the boot process, like 'clock', which sets the timezone, or 'keyboard' which controls the keyboard map. The contents may vary drastically depending on which distribution and what utilities you have installed. For example, on a Redhat or Mandrake based system it is possible to alter an endless array of attributes from the default desktop to whether DMA should be enabled for your IDE devices. On our Debian reference system though this folder is almost expedient containing only two files hwconf and soundcard which are both configured by the Redhat utilities hwconf and sndconfig respectively.

/etc/slip
: Configuration files for the setup and operation of SLIP (serial line IP) interface. Generally unused nowadays. This protocol has been superceded by the faster and more efficient PPP protocol.

/etc/screenrc
: This is the system wide screenrc. You can use this file to change the default behavior of screen system wide or copy it to ~/.screenrc and use it as a starting point for your own settings. Commands in this file are used to set options, bind screen functions to keys, redefine terminal capabilities, and to automatically establish one or more windows at the beginning of your screen session. This is not a comprehensive list of options, look at the screen manual for details on everything that you can put in this file.

/etc/scrollkeeper.conf
: A free electronic cataloging system for documentation. It stores metadata specified by the http://www.ibiblio.org/osrt/omf/ (Open Source Metadata Framework) as well as certain metadata extracted directly from documents (such as the table of contents). It provides various functionality pertaining to this metadata to help browsers, such as sorting the registered documents or searching the metadata for documents which satisfy a set of criteria.

/etc/ssh
: 'ssh' configuration files. 'ssh' is a secure rlogin/rsh/rcp replacement (OpenSSH). This is the portable version of OpenSSH, a free implementation of the Secure Shell protocol as specified by the IETF secsh working group. 'ssh' (Secure Shell) is a program for logging into a remote machine and for executing commands on a remote machine. It provides secure encrypted communications between two untrusted hosts over an insecure network. X11 connections and arbitrary TCP/IP ports can also be forwarded over the secure channel. It is intended as a replacement for rlogin, rsh and rcp, and can be used to provide applications with a secure communication channel. It should be noted that in some

countries, particularly Iraq, and Pakistan, it may be illegal to use any encryption at all without a special permit.

/etc/syslog.conf
: Lists where log files should go, what messages are written to them and the level of verbosity. It is also now possible to filter based on message content, message integrity, message encryption (near future), portability and better network forwarding.

/etc/termcap
: The terminal capability database. Describes the "escape sequences" by which various terminals can be controlled. Programs are written so that instead of directly outputting an escape sequence that only works on a particular brand of terminal, they look up the correct sequence to do whatever it is they want to do in /etc/termcap. As a result most programs work with most kinds of terminals.

/etc/timezone
: local timezone.

/etc/updatedb.conf
: Sets environment variables that are used by updatedb which therefore configures the database for 'locate', a utility that locates a pattern in a database of filenames and returns the filenames that match.

```
# This file sets environment variables which are used by updatedb

# filesystems which are pruned from updatedb database
PRUNEFS="NFS nfs afs proc smbfs autofs auto iso9660 ncpfs coda devpts ftpfs"
export PRUNEFS
# paths which are pruned from updatedb database
PRUNEPATHS="/tmp /usr/tmp /var/tmp /afs /amd /alex /var/spool"
export PRUNEPATHS
# netpaths which are added
NETPATHS=""
export NETPATHS
```

/etc/vga
: The configuration file for the svgalib is stored in this directory. svgalib provides graphics capabilities to programs running on the system console, without going through the X Window System. It uses direct access to the video hardware to provide low−level access to the standard VGA and SVGA graphics modes. It only works with some video hardware; so use with caution.

/etc/vim
: Contains configuration files for both vim and its X based counterpart gvim. A wide range of options can be accessed though these two files such as automatic indentation, syntax highlighting, etc....

/etc/xinetd.d/
: The original 'inetd' daemon has now been superceded by the much improved 'xinetd'. 'inetd' should be run at boot time by /etc/init.d/inetd (or /etc/rc.local on some systems). It then listens for connections on certain Internet sockets. When a connection is found on one of its sockets, it decides what service the socket corresponds to, and invokes a program to service the request. After the program is finished, it continues to listen on the socket (except in some cases). Essentially, inetd allows running one daemon to invoke several others, reducing load on the system. Services controlled via xinetd put their configuration files here.

/etc/zlogin
: System−wide .zlogin file for zsh(1). This file is sourced only for login shells. It should contain commands that should be executed only in login shells. It should be used to set the terminal type and run a series of external commands (fortune, msgs, from, etc.)

/etc/zlogout
: Commands to be executed upon user exit from the zsh. Its control is system−wide but the .zlogout file for zsh(1) does override it in terms of importance.

/etc/zprofile

System−wide .zprofile file for zsh(1). This file is sourced only for login shells (i.e. Shells invoked with "−" as the first character of argv[0], and shells invoked with the −l flag.)

/etc/zshenv
: System−wide .zshenv file for zsh(1). This file is sourced on all invocations of the shell. If the −f flag is present or if the NO_RCS option is set within this file, all other initialization files are skipped. This file should contain commands to set the command search path, plus other important environment variables. This file should not contain commands that produce output or assume the shell is attached to a tty.

/etc/zshrc
: System−wide .zshrc file for zsh(1). This file is sourced only for interactive shells. It should contain commands to set up aliases, functions, options, key bindings, etc.

Compliance with the FSSTND requires that the following directories, or symbolic links to directories are required in /etc:

```
opt       Configuration for /opt
X11       Configuration for the X Window system (optional)
sgml      Configuration for SGML (optional)
xml       Configuration for XML (optional)

The following directories, or symbolic links to directories must be in /etc,
if the corresponding subsystem is installed:

opt       Configuration for /opt

The following files, or symbolic links to files, must be in /etc if the
corresponding subsystem is installed (it is recommended that files be
stored in subdirectories of /etc/ rather than directly in /etc:

csh.login   Systemwide initialization file for C shell logins (optional)
exports     NFS filesystem access control list (optional)
fstab       Static information about filesystems (optional)
ftpusers    FTP daemon user access control list (optional)
gateways    File which lists gateways for routed (optional)
gettydefs   Speed and terminal settings used by getty (optional)
group       User group file (optional)
host.conf   Resolver configuration file (optional)
hosts       Static information about host names (optional)
hosts.allow Host access file for TCP wrappers (optional)
hosts.deny  Host access file for TCP wrappers (optional)
hosts.equiv List of trusted hosts for rlogin, rsh, rcp (optional)
hosts.lpd   List of trusted hosts for lpd (optional)
inetd.conf  Configuration file for inetd (optional)
inittab     Configuration file for init (optional)
issue       Pre-login message and identification file (optional)
ld.so.conf  List of extra directories to search for shared libraries
            (optional)
motd        Post-login message of the day file (optional)
mtab        Dynamic information about filesystems (optional)
mtools.conf Configuration file for mtools (optional)
networks    Static information about network names (optional)
passwd      The password file (optional)
printcap    The lpd printer capability database (optional)
profile     Systemwide initialization file for sh shell logins (optional)
protocols   IP protocol listing (optional)
resolv.conf Resolver configuration file (optional)
rpc         RPC protocol listing (optional)
securetty   TTY access control for root login (optional)
services    Port names for network services (optional)
```

```
shells      Pathnames of valid login shells (optional)
syslog.conf Configuration file for syslogd (optional)

mtab does not fit the static nature of /etc: it is excepted for historical
reasons. On some Linux systems, this may be a symbolic link to /proc/mounts,
in which case this exception is not required.

/etc/opt : Configuration files for /opt
Host-specific configuration files for add-on application software packages
must be installed within the directory /etc/opt/&60;subdir&62;, where
&60;subdir&62; is the name of the subtree in /opt where the static data
from that package is stored.

No structure is imposed on the internal arrangement of /etc/opt/&60;subdir&62;.
If a configuration file must reside in a different location in order for the
package or system to function properly, it may be placed in a location other
than /etc/opt/&60;subdir&62;.

The rationale behind this subtree is best explained by refering to the
rationale for /opt.

/etc/X11 : Configuration for the X Window System (optional)
/etc/X11 is the location for all X11 host-specific configuration. This
directory is necessary to allow local control if /usr is mounted read only.

The following files, or symbolic links to files, must be in /etc/X11 if the
corresponding subsystem is installed:

Xconfig    The configuration file for early versions of XFree86 (optional)
XF86Config The configuration file for XFree86 versions 3 and 4 (optional)
Xmodmap    Global X11 keyboard modification file (optional)

Subdirectories of /etc/X11 may include those for xdm and for any other
programs (some window managers, for example) that need them.

/etc/X11/xdm holds the configuration files for xdm. These are most of the
files previously found in /usr/lib/X11/xdm. Some local variable data for
xdm is stored in /var/lib/xdm.

It is recommended that window managers with only one configuration file
which is a default .*wmrc file must name it system.*wmrc (unless there is
a widely-accepted alternative name) and not use a subdirectory. Any window
manager subdirectories must be identically named to the actual window
manager binary.

/etc/sgml : Configuration files for SGML (optional)
Generic configuration files defining high-level parameters of the SGML
systems are installed here. Files with names *.conf indicate generic
configuration files. File with names *.cat are the DTD-specific centralized
catalogs, containing references to all other catalogs needed to use the
given DTD. The super catalog file catalog references all the centralized
catalogs.

/etc/xml : Configuration files for XML (optional)
Generic configuration files defining high-level parameters of the XML
systems are installed here. Files with names *.conf indicate generic
configuration files. The super catalog file catalog references all the
centralized catalogs.
```

1.7. /home

Linux is a multi−user environment so each user is also assigned a specific directory that is accessible only to them and the system administrator. These are the user home directories, which can be found under '/home/$USER' (~/). It is your playground: everything is at your command, you can write files, delete them, install programs, etc.... Your home directory contains your personal configuration files, the so−called dot files (their name is preceded by a dot). Personal configuration files are usually 'hidden', if you want to see them, you either have to turn on the appropriate option in your file manager or run ls with the −a switch. If there is a conflict between personal and system wide configuration files, the settings in the personal file will prevail.

Dotfiles most likely to be altered by the end user are probably your .xsession and .bashrc files. The configuration files for X and Bash respectively. They allow you to be able to change the window manager to be startup upon login and also aliases, user−specified commands and environment variables respectively. Almost always when a user is created their dotfiles will be taken from the /etc/skel directory where system administrators place a sample file that user's can modify to their hearts content.

/home can get quite large and can be used for storing downloads, compiling, installing and running programs, your mail, your collection of image or sound files etc.

The FSSTND states that:

```
/home is a fairly standard concept, but it is clearly a site-specific
filesystem.

Different people prefer to place user accounts in a variety of places.
This section describes only a suggested placement for user home
directories; nevertheless we recommend that all FHS-compliant
distributions use this as the default location for home directories.
On small systems, each user's directory is typically one of the many
subdirectories of /home such as /home/smith, /home/torvalds,
/home/operator, etc. On large systems (especially when the /home
directories are shared amongst many hosts using NFS) it is useful
to subdivide user home directories. Subdivision may be accomplished by
using subdirectories such as /home/staff, /home/guests, /home/students,
etc.

The setup will differ from host to host. Therefore, no program
should rely on this location.

If you want to find out a user's home directory, you should use the
getpwent(3) library function rather than relying on /etc/passwd because
user information may be stored remotely using systems such as NIS.

User specific configuration files for applications are stored in the
user's home directory in a file that starts with the '.' character
(a "dot file"). If an application needs to create more than one dot
file then they should be placed in a subdirectory with a name starting
with a '.' character, (a "dot directory"). In this case the
configuration files should not start with the '.' character.

It is recommended that apart from autosave and lock files programs
should refrain from creating non dot files or directories in a home
directory without user intervention.
```

1.8. /initrd

```
initrd provides the capability to load a RAM disk by the boot
loader. This RAM disk can then be mounted as the root file
system and programs can be run from it. Afterwards, a new
root file system can be mounted from a different device. The
previous root (from initrd) is then moved to a directory and
can be subsequently unmounted.

initrd is mainly designed to allow system startup to occur
in two phases, where the kernel comes up with a minimum set
of compiled-in drivers, and where additional modules are
loaded from initrd.

Operation
---------

When using initrd, the system typically boots as follows:

 1) the boot loader loads the kernel and the initial RAM disk
 2) the kernel converts initrd into a "normal" RAM disk and
    frees the memory used by initrd
 3) initrd is mounted read-write as root
 4) /linuxrc is executed (this can be any valid executable,
 including shell scripts; it is run with uid 0 and can do
 basically everything init can do)
 5) linuxrc mounts the "real" root file system
 6) linuxrc places the root file system at the root directory
    using the pivot_root system call
 7) the usual boot sequence (e.g. invocation of /sbin/init) is
    performed on the root file system
 8) the initrd file system is removed

Note that changing the root directory does not involve unmounting
it. It is therefore possible to leave processes running on initrd
during that procedure. Also note that file systems mounted under
initrd continue to be accessible.

Usage scenarios
---------------

The main motivation for implementing initrd was to allow
for modular kernel configuration at system installation.
The procedure would work as follows:

 1) system boots from floppy or other media with a minimal kernel
    (e.g. support for RAM disks, initrd, a.out, and the Ext2 FS)
    and loads initrd
 2) /linuxrc determines what is needed to (1) mount the "real" root
    FS (i.e. device type, device drivers, file system) and (2) the
    distribution media (e.g. CD-ROM, network, tape, ...). This can
    be done by asking the user, by auto-probing, or by using a
    hybrid approach.
 3) /linuxrc loads the necessary kernel modules
 4) /linuxrc creates and populates the root file system (this
    doesn't have to be a very usable system yet)
 5) /linuxrc invokes pivot_root to change the root file system and
    execs - via chroot - a program that continues the installation
 6) the boot loader is installed
 7) the boot loader is configured to load an initrd with the set of
    modules that was used to bring up the system (e.g. /initrd can
```

```
    be modified, then unmounted, and finally, the image is written
    from /dev/ram0 or /dev/rd/0 to a file)
 8) now the system is bootable and additional installation tasks
    can be performed

The key role of initrd here is to re-use the configuration data
during normal system operation without requiring the use of a
bloated "generic" kernel or re-compiling or re-linking the kernel.

A second scenario is for installations where Linux runs on systems
with different hardware configurations in a single administrative
domain. In such cases, it is desirable to generate only a small set
of kernels (ideally only one) and to keep the system-specific part
of configuration information as small as possible. In this case, a
common initrd could be generated with all the necessary modules.
Then, only /linuxrc or a file read by it would have to be different.

A third scenario are more convenient recovery disks, because
information like the location of the root FS partition doesn't have
to be provided at boot time, but the system loaded from initrd can
invoke a user-friendly dialog and it can also perform some sanity
checks (or even some form of auto-detection).

Last not least, CD-ROM distributors may use it for better installation
from CD, e.g. by using a boot floppy and bootstrapping a bigger RAM disk
via initrd from CD; or by booting via a loader like LOADLIN or directly
from the CD-ROM, and loading the RAM disk from CD without need of floppies.
```

1.9. /lib

The /lib directory contains kernel modules and those shared library images (the C programming code library) needed to boot the system and run the commands in the root filesystem, ie. by binaries in /bin and /sbin. Libraries are readily identifiable through their filename extension of *.so. Windows equivalent to a shared library would be a DLL (dynamically linked library) file. They are essential for basic system functionality. Kernel modules (drivers) are in the subdirectory /lib/modules/'kernel−version'. To ensure proper module compilation you should ensure that /lib/modules/'kernel−version'/kernel/build points to /usr/src/'kernel−version' or ensure that the Makefile knows where the kernel source itself are located.

/lib/'machine−architecture'
: Contains platform/architecture dependent libraries.

/lib/iptables
: iptables shared library files.

/lib/kbd
: Contains various keymaps.

/lib/modules/'kernel−version'
: The home of all the kernel modules. The organisation of files here is reasonably clear so no requires no elaboration.

/lib/modules/'kernel−version'/isapnpmap.dep
: has details on ISA based cards, the modules that they require and various other attributes.

/lib/modules/'kernel−version'/modules.dep
: lists all modules dependencies. This file can be updated using the depmod command.

/lib/modules/'kernel−version'/pcimap
: is the PCI equivalent of the /lib/modules/'kernel−version'/isapnpmap.dep file.

/lib/modules/'kernel−version'/usbmap
: is the USB equivalent of the /lib/modules/'kernel−version'/isapnpmap.dep file.

/lib/oss
: All OSS (Open Sound System) files are installed here by default.

/lib/security
: PAM library files.

```
The FSSTND states that the /lib directory contains those shared library
images needed to boot the system and run the commands in the root filesystem,
ie. by binaries in /bin and /sbin.

Shared libraries that are only necessary for binaries in /usr (such as any
X Window binaries) must not be in /lib. Only the shared libraries required
to run binaries in /bin and /sbin may be here. In particular, the library
libm.so.* may also be placed in /usr/lib if it is not required by anything
in /bin or /sbin.

At least one of each of the following filename patterns are required (they
may be files, or symbolic links):

libc.so.* The dynamically-linked C library (optional)
ld*       The execution time linker/loader (optional)

If a C preprocessor is installed, /lib/cpp must be a reference to it, for
historical reasons. The usual placement of this binary is /usr/bin/cpp.

The following directories, or symbolic links to directories, must be in
/lib, if the corresponding subsystem is installed:

modules   Loadable kernel modules (optional)

/lib<qual> : Alternate format essential shared libraries (optional)

There may be one or more variants of the /lib directory on systems which
support more than one binary format requiring separate libraries.

This is commonly used for 64-bit or 32-bit support on systems which support
multiple binary formats, but require libraries of the same name. In this
case, /lib32 and /lib64 might be the library directories, and /lib a symlink
to one of them.

If one or more of these directories exist, the requirements for their contents
are the same as the normal /lib directory, except that /lib<qual>/cpp is
not required.

/lib<qual>/cpp is still permitted: this allows the case where /lib and
/lib<qual> are the same (one is a symbolic link to the other).
```

1.10. /lost+found

As was explained earlier during the overview of the FSSTND, Linux should always go through a proper shutdown. Sometimes your system might crash or a power failure might take the machine down. Either way, at the next boot, a lengthy filesystem check (the speed of this check is dependent on the type of filesystem that you actually use. ie. ext3 is faster than ext2 because it is a journalled filesystem) using fsck will be done. Fsck will go through the system and try to recover any corrupt files that it finds. The result of this recovery operation will be placed in this directory. The files recovered are not likely to be complete or make much sense but there always is a chance that something worthwhile is recovered. Each partition has its own lost+found directory. If you find files in there, try to move them back to their original location. If you find something like a broken symbolic link to 'file', you have to reinstall the file/s from the corresponding RPM, since your file system got damaged so badly that the files were mutilated beyond recognition. Below is an example of a /lost+found directory. As you can see, the vast majority of files contained here are in actual fact sockets. As for the rest of the other files they were found to be damaged system files and personal files. These files were not able to be recovered.

```
      total 368
      -rw-r--r-- 1 root root 110891 Oct 5 14:14 #388200
      -rw-r--r-- 1 root root 215 Oct 5 14:14 #388201
      -rw-r--r-- 1 root root 110303 Oct 6 23:09 #388813
      -rw-r--r-- 1 root root 141 Oct 6 23:09 #388814
      -rw-r--r-- 1 root root 110604 Oct 6 23:09 #388815a
      -rw-r--r-- 1 root root 194 Oct 6 23:09 #388816
      srwxr-xr-x 1 root root 0 Oct 6 13:00 #51430
      srwxr-xr-x 1 root root 0 Oct 6 00:23 #51433
      -rw------- 1 root root 63 Oct 6 00:23 #51434
      srwxr-xr-x 1 root root 0 Oct 6 13:00 #51436
      srwxrwxrwx 1 root root 0 Oct 6 00:23 #51437
      srwx------ 1 root root 0 Oct 6 00:23 #51438
      -rw------- 1 root root 63 Oct 6 13:00 #51439
      srwxrwxrwx 1 root root 0 Oct 6 13:00 #51440
      srwx------ 1 root root 0 Oct 6 13:00 #51442
      -rw------- 1 root root 63 Oct 6 23:09 #51443
      srwx------ 1 root root 0 Oct 6 10:40 #51445
      srwxrwxrwx 1 root root 0 Oct 6 23:09 #51446
      srwx------ 1 root root 0 Oct 6 23:09 #51448
```

1.11. /media

Amid much controversy and consternation on the part of system and network administrators a directory containing mount points for removable media has now been created. Funnily enough, it has been named /media.

```
This directory contains subdirectories which are used as mount points for
removeable media such as floppy disks, cdroms and zip disks.

The motivation for the creation of this directory has been that historically
there have been a number of other different places used to mount removeable
media such as /cdrom, /mnt or /mnt/cdrom. Placing the mount points for all
removeable media directly in the root directory would potentially result in
a large number of extra directories in /. Although the use of subdirectories
in /mnt as a mount point has recently been common, it conflicts with a much
older tradition of using /mnt directly as a temporary mount point.

The following directories, or symbolic links to directories, must be in /media,
if the corresponding subsystem is installed:

floppy     Floppy drive (optional)
cdrom      CD-ROM drive (optional)
cdrecorder CD writer (optional)
zip        Zip drive (optional)

On systems where more than one device exists for mounting a certain type of
media, mount directories can be created by appending a digit to the name of
those available above starting with '0', but the unqualified name must also
exist.

A compliant implementation with two CDROM drives might have /media/cdrom0
```

```
and /media/cdrom1 with /media/cdrom a symlink to either of these.
```

Please see the section on the /mnt directory to achieve a better understanding of the process on mounting and unmounting filesystems.

1.12. /mnt

This is a generic mount point under which you mount your filesystems or devices. Mounting is the process by which you make a filesystem available to the system. After mounting your files will be accessible under the mount−point. This directory usually contains mount points or sub−directories where you mount your floppy and your CD. You can also create additional mount−points here if you wish. Standard mount points would include /mnt/cdrom and /mnt/floppy. There is no limitation to creating a mount−point anywhere on your system but by convention and for sheer practicality do not litter your file system with mount−points. It should be noted that some distributions like Debian allocate /floppy and /cdrom as mount points while Redhat and Mandrake puts them in /mnt/floppy and /mnt/cdrom respectively.

However, it should be noted that as of FSSTND version 2.3 the purpose of this directory has changed.

```
This directory is provided so that the system administrator may temporarily
mount a filesystem as needed. The content of this directory is a local issue
and should not affect the manner in which any program is run.

This directory must not be used by installation programs: a suitable temporary
directory not in use by the system must be used instead.
```

1.12.1. Mounting and unmounting

Before one can use a filesystem, it has to be *mounted*. The operating system then does various bookkeeping things to make sure that everything works. Since all files in UNIX are in a single directory tree, the mount operation will make it look like the contents of the new filesystem are the contents of an existing subdirectory in some already mounted filesystem.

The mounts could be done as in the following example:

```
$ mount /dev/hda2 /home
$ mount /dev/hda3 /usr
$
```

The **mount** command takes two arguments. The first one is the device file corresponding to the disk or partition containing the filesystem. The second one is the directory below which it will be mounted. After these commands the contents of the two filesystems look just like the contents of the `/home` and `/usr` directories, respectively. One would then say that ``` `` ```/dev/hda2 *is mounted on* `/home`'', and similarly for `/usr`. To look at either filesystem, one would look at the contents of the directory on which it has been mounted, just as if it were any other directory. Note the difference between the device file, `/dev/hda2`, and the mounted−on directory, `/home`. The device file gives access to the raw contents of the disk, the mounted−on directory gives access to the files on the disk. The mounted−on directory is called the *mount point*.

Linux supports many filesystem types. **mount** tries to guess the type of the filesystem. You can also use the `−t fstype` option to specify the type directly; this is sometimes necessary, since the heuristics **mount** uses do not always work. For example, to mount an MS−DOS floppy, you could use the following command:

```
$ mount -t msdos /dev/fd0 /floppy
   $
```

The mounted−on directory need not be empty, although it must exist. Any files in it, however, will be inaccessible by name while the filesystem is mounted. (Any files that have already been opened will still be accessible. Files that have hard links from other directories can be accessed using those names.) There is no harm done with this, and it can even be useful. For instance, some people like to have `/tmp` and `/var/tmp` synonymous, and make `/tmp` be a symbolic link to `/var/tmp`. When the system is booted, before the `/var` filesystem is mounted, a `/var/tmp` directory residing on the root filesystem is used instead. When `/var` is mounted, it will make the `/var/tmp` directory on the root filesystem inaccessible. If `/var/tmp` didn't exist on the root filesystem, it would be impossible to use temporary files before mounting `/var`.

If you don't intend to write anything to the filesystem, use the `−r` switch for **mount** to do a *read−only mount*. This will make the kernel stop any attempts at writing to the filesystem, and will also stop the kernel from updating file access times in the inodes. Read−only mounts are necessary for unwritable media, e.g., CD−ROMs.

The alert reader has already noticed a slight logistical problem. How is the first filesystem (called the *root filesystem*, because it contains the root directory) mounted, since it obviously can't be mounted on another filesystem? Well, the answer is that it is done by magic.

For more information, see the kernel source or the Kernel Hackers' Guide.

The root filesystem is magically mounted at boot time, and one can rely on it to always be mounted. If the root filesystem can't be mounted, the system does not boot. The name of the filesystem that is magically mounted as root is either compiled into the kernel, or set using LILO or **rdev**.

The root filesystem is usually first mounted read−only. The startup scripts will then run **fsck** to verify its validity, and if there are no problems, they will *re−mount* it so that writes will also be allowed. **fsck** must not be run on a mounted filesystem, since any changes to the filesystem while **fsck** is running *will* cause trouble. Since the root filesystem is mounted read−only while it is being checked, **fsck** can fix any problems without worry, since the remount operation will flush any metadata that the filesystem keeps in memory.

On many systems there are other filesystems that should also be mounted automatically at boot time. These are specified in the `/etc/fstab` file; see the fstab man page for details on the format. The details of exactly when the extra filesystems are mounted depend on many factors, and can be configured by each administrator if need be.

When a filesystem no longer needs to be mounted, it can be unmounted with **umount**.

It should of course be **unmount**, but the n mysteriously disappeared in the 70s, and hasn't been seen since. Please return it to Bell Labs, NJ, if you find it.

umount takes one argument: either the device file or the mount point. For example, to unmount the directories of the previous example, one could use the commands

```
$ umount /dev/hda2
  $ umount /usr
  $
```

See the man page for further instructions on how to use the command. It is imperative that you always unmount a mounted floppy. *Don't just pop the floppy out of the drive!* Because of disk caching, the data is not necessarily written to the floppy until you unmount it, so removing the floppy from the drive too early might cause the contents to become garbled. If you only read from the floppy, this is not very likely, but if you write, even accidentally, the result may be catastrophic.

Mounting and unmounting requires super user privileges, i.e., only root can do it. The reason for this is that if any user can mount a floppy on any directory, then it is rather easy to create a floppy with, say, a Trojan horse disguised as `/bin/sh`, or any other often used program. However, it is often necessary to allow users to use floppies, and there are several ways to do this:

- Give the users the root password. This is obviously bad security, but is the easiest solution. It works well if there is no need for security anyway, which is the case on many non−networked, personal systems.
- Use a program such as **sudo** to allow users to use mount. This is still bad security, but doesn't directly give super user privileges to everyone. [1]
- Make the users use **mtools**, a package for manipulating MS−DOS filesystems, without mounting them. This works well if MS−DOS floppies are all that is needed, but is rather awkward otherwise.
- List the floppy devices and their allowable mount points together with the suitable options in `/etc/fstab`.

The last alternative can be implemented by adding a line like the following to the `/etc/fstab` file:

```
/dev/fd0 /floppy
msdos user,noauto 0 0
```

The columns are: device file to mount, directory to mount on, filesystem type, options, backup frequency (used by **dump**), and **fsck** pass number (to specify the order in which filesystems should be checked upon boot; 0 means no check).

The `noauto` option stops this mount to be done automatically when the system is started (i.e., it stops **mount −a** from mounting it). The `user` option allows any user to mount the filesystem, and, because of security reasons, disallows execution of programs (normal or setuid) and interpretation of device files from the mounted filesystem. After this, any user can mount a floppy with an msdos filesystem with the following command:

```
$ mount /floppy
$
```

The floppy can (and needs to, of course) be unmounted with the corresponding **umount** command.

If you want to provide access to several types of floppies, you need to give several mount points. The settings can be different for each mount point. For example, to give access to both MS−DOS and ext2 floppies, you could have the following to lines in `/etc/fstab`:

```
/dev/fd0 /dosfloppy msdos user,noauto 0 0 /dev/fd0
/ext2floppy ext2 user,noauto 0 0
```

For MS−DOS filesystems (not just floppies), you probably want to restrict access to it by using the `uid`, `gid`, and `umask` filesystem options, described in detail on the **mount** manual page. If you aren't careful, mounting an MS−DOS filesystem gives everyone at least read access to the files in it, which is not a good idea.

1.13. /opt

This directory is reserved for all the software and add−on packages that are not part of the default installation. For example, StarOffice, Kylix, Netscape Communicator and WordPerfect packages are normally found here. To comply with the FSSTND, all third party applications should be installed in this directory. Any package to be installed here must locate its static files (ie. extra fonts, clipart, database files) must locate its static files in a separate /opt/'package' or /opt/'provider' directory tree (similar to the way in which Windows will install new software to its own directory tree C:\Windows\Progam Files\"Program Name"), where 'package' is a name that describes the software package and 'provider' is the provider's LANANA registered name.

Although most distributions neglect to create the directories /opt/bin, /opt/doc, /opt/include, /opt/info, /opt/lib, and /opt/man they are reserved for local system administrator use. Packages may provide "front−end" files intended to be placed in (by linking or copying) these reserved directories by the system administrator, but must function normally in the absence of these reserved directories. Programs to be invoked by users are located in the directory /opt/'package'/bin. If the package includes UNIX manual pages, they are located in /opt/'package'/man and the same substructure as /usr/share/man must be used. Package files that are variable must be installed in /var/opt. Host−specific configuration files are installed in /etc/opt.

Under no circumstances are other package files to exist outside the /opt, /var/opt, and /etc/opt hierarchies except for those package files that must reside in specific locations within the filesystem tree in order to function properly. For example, device lock files in /var/lock and devices in /dev. Distributions may install software in /opt, but must not modify or delete software installed by the local system administrator without the assent of the local system administrator.

The use of /opt for add−on software is a well−established practice in the UNIX community. The System V Application Binary Interface [AT&T 1990], based on the System V Interface Definition (Third Edition) and the Intel Binary Compatibility Standard v. 2 (iBCS2) provides for an /opt structure very similar to the one defined here.

Generally, all data required to support a package on a system must be present within /opt/'package', including files intended to be copied into /etc/opt/'package' and /var/opt/'package' as well as reserved directories in /opt. The minor restrictions on distributions using /opt are necessary because conflicts are possible between distribution installed and locally installed software, especially in the case of fixed pathnames found in some binary software.

The structure of the directories below /opt/'provider' is left up to the packager of the software, though it is recommended that packages are installed in /opt/'provider'/'package' and follow a similar structure to the guidelines for /opt/package. A valid reason for diverging from this structure is for support packages which may have files installed in /opt/ 'provider'/lib or /opt/'provider'/bin.

1.14. /proc

/proc is very special in that it is also a virtual filesystem. It's sometimes referred to as a process information pseudo−file system. It doesn't contain 'real' files but runtime system information (e.g. system memory, devices mounted, hardware configuration, etc). For this reason it can be regarded as a control and information centre for the kernel. In fact, quite a lot of system utilities are simply calls to files in this directory. For example, 'lsmod' is the same as 'cat /proc/modules' while 'lspci' is a synonym for 'cat /proc/pci'. By altering files located in this directory you can even read/change kernel parameters (sysctl) while the system is running.

The most distinctive thing about files in this directory is the fact that all of them have a file size of 0, with the exception of kcore, mtrr and self. A directory listing looks similar to the following:

```
total 525256
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 1
dr-xr-xr-x    3 daemon   root            0 Jan 19 15:00 109
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 170
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 173
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 178
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 2
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 3
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 4
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 421
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 425
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 433
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 439
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 444
dr-xr-xr-x    3 daemon   daemon          0 Jan 19 15:00 446
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 449
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 453
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 456
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 458
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 462
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 463
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 464
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 465
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 466
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 467
dr-xr-xr-x    3 gdm      gdm             0 Jan 19 15:00 472
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 483
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 5
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 6
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 7
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 8
-r--r--r--    1 root     root            0 Jan 19 15:00 apm
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 bus
-r--r--r--    1 root     root            0 Jan 19 15:00 cmdline
-r--r--r--    1 root     root            0 Jan 19 15:00 cpuinfo
-r--r--r--    1 root     root            0 Jan 19 15:00 devices
-r--r--r--    1 root     root            0 Jan 19 15:00 dma
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 driver
-r--r--r--    1 root     root            0 Jan 19 15:00 execdomains
-r--r--r--    1 root     root            0 Jan 19 15:00 fb
-r--r--r--    1 root     root            0 Jan 19 15:00 filesystems
dr-xr-xr-x    2 root     root            0 Jan 19 15:00 fs
dr-xr-xr-x    4 root     root            0 Jan 19 15:00 ide
-r--r--r--    1 root     root            0 Jan 19 15:00 interrupts
-r--r--r--    1 root     root            0 Jan 19 15:00 iomem
-r--r--r--    1 root     root            0 Jan 19 15:00 ioports
dr-xr-xr-x   18 root     root            0 Jan 19 15:00 irq
-r--------    1 root     root     536809472 Jan 19 15:00 kcore
-r--------    1 root     root            0 Jan 19 14:58 kmsg
-r--r--r--    1 root     root            0 Jan 19 15:00 ksyms
-r--r--r--    1 root     root            0 Jan 19 15:00 loadavg
-r--r--r--    1 root     root            0 Jan 19 15:00 locks
-r--r--r--    1 root     root            0 Jan 19 15:00 mdstat
-r--r--r--    1 root     root            0 Jan 19 15:00 meminfo
-r--r--r--    1 root     root            0 Jan 19 15:00 misc
-r--r--r--    1 root     root            0 Jan 19 15:00 modules
-r--r--r--    1 root     root            0 Jan 19 15:00 mounts
-rw-r--r--    1 root     root          137 Jan 19 14:59 mtrr
```

```
dr-xr-xr-x    3 root     root            0 Jan 19 15:00 net
dr-xr-xr-x    2 root     root            0 Jan 19 15:00 nv
-r--r--r--    1 root     root            0 Jan 19 15:00 partitions
-r--r--r--    1 root     root            0 Jan 19 15:00 pci
dr-xr-xr-x    4 root     root            0 Jan 19 15:00 scsi
lrwxrwxrwx    1 root     root           64 Jan 19 14:58 self -> 483
-rw-r--r--    1 root     root            0 Jan 19 15:00 slabinfo
-r--r--r--    1 root     root            0 Jan 19 15:00 stat
-r--r--r--    1 root     root            0 Jan 19 15:00 swaps
dr-xr-xr-x   10 root     root            0 Jan 19 15:00 sys
dr-xr-xr-x    2 root     root            0 Jan 19 15:00 sysvipc
dr-xr-xr-x    4 root     root            0 Jan 19 15:00 tty
-r--r--r--    1 root     root            0 Jan 19 15:00 uptime
-r--r--r--    1 root     root            0 Jan 19 15:00 version
```

Each of the numbered directories corresponds to an actual process ID. Looking at the process table, you can match processes with the associated process ID. For example, the process table might indicate the following for the secure shell server:

```
# ps ax | grep sshd
439 ? S 0:00 /usr/sbin/sshd
```

Details of this process can be obtained by looking at the associated files in the directory for this process, /proc/460. You might wonder how you can see details of a process that has a file size of 0. It makes more sense if you think of it as a window into the kernel. The file doesn't actually contain any data; it just acts as a pointer to where the actual process information resides. For example, a listing of the files in the /proc/460 directory looks similar to the following:

```
total 0
-r--r--r--    1 root     root            0 Jan 19 15:02 cmdline
lrwxrwxrwx    1 root     root            0 Jan 19 15:02 cwd -> /
-r--------    1 root     root            0 Jan 19 15:02 environ
lrwxrwxrwx    1 root     root            0 Jan 19 15:02 exe -> /usr/sbin/sshd
dr-x------    2 root     root            0 Jan 19 15:02 fd
-r--r--r--    1 root     root            0 Jan 19 15:02 maps
-rw-------    1 root     root            0 Jan 19 15:02 mem
lrwxrwxrwx    1 root     root            0 Jan 19 15:02 root -> /
-r--r--r--    1 root     root            0 Jan 19 15:02 stat
-r--r--r--    1 root     root            0 Jan 19 15:02 statm
-r--r--r--    1 root     root            0 Jan 19 15:02 status
```

The purpose and contents of each of these files is explained below:

/proc/PID/cmdline
: Command line arguments.

/proc/PID/cpu
: Current and last cpu in which it was executed.

/proc/PID/cwd
: Link to the current working directory.

/proc/PID/environ
: Values of environment variables.

/proc/PID/exe
: Link to the executable of this process.

/proc/PID/fd
: Directory, which contains all file descriptors.

/proc/PID/maps
: Memory maps to executables and library files.

/proc/PID/mem

Memory held by this process.

/proc/PID/root

Link to the root directory of this process.

/proc/PID/stat

Process status.

/proc/PID/statm

Process memory status information.

/proc/PID/status

Process status in human readable form.

Should you wish to know more, the man page for proc describes each of the files associated with a running process ID in far greater detail.

Even though files appear to be of size 0, examining their contents reveals otherwise:

```
# cat status
```

```
Name: sshd
State: S (sleeping)
Tgid: 439
Pid: 439
PPid: 1
TracerPid: 0
Uid: 0 0 0 0
Gid: 0 0 0 0
FDSize: 32
Groups:
VmSize:     2788 kB
VmLck:         0 kB
VmRSS:      1280 kB
VmData:      252 kB
VmStk:        16 kB
VmExe:       268 kB
VmLib:      2132 kB
SigPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: 8000000000001000
SigCgt: 0000000000014005
CapInh: 0000000000000000
CapPrm: 00000000fffffeff
CapEff: 00000000fffffeff
```

The files in the /proc directory act very similar to the process ID subdirectory files. For example, examining the contents of the /proc/interrupts file displays something like the following:

```
# cat interrupts
```

```
           CPU0
  0:      32657          XT-PIC  timer
  1:       1063          XT-PIC  keyboard
  2:          0          XT-PIC  cascade
  8:          3          XT-PIC  rtc
  9:          0          XT-PIC  cmpci
 11:        332          XT-PIC  nvidia
 14:       5289          XT-PIC  ide0
 15:         13          XT-PIC  ide1
NMI:          0
ERR:          0
```

Each of the numbers down the left−hand column represents the interrupt that is in use. Examining the contents of the file dynamically gathers the associated data and displays it to the screen. Most of the /proc file system is read−only; however, some files allow kernel variable to be changed. This provides a mechanism to actually tune the kernel without recompiling and rebooting.

The procinfo utility summarizes /proc file system information into a display similar to the following:

```
# /usr/bin/procinfo
```

```
Linux 2.4.18 (root@DEB) (gcc 2.95.4 20011002 ) #2 1CPU [DEB.(none)]

Memory:      Total        Used        Free      Shared     Buffers      Cached
Mem:        513908      107404      406504           0        2832       82180
Swap:       265032           0      265032

Bootup: Sun Jan 19 14:58:27 2003    Load average: 0.29 0.13 0.05 1/30 566

user  :       0:00:10.26   2.3%  page in :    74545  disk 1:     6459r     796w
nice  :       0:00:00.00   0.0%  page out:     9416  disk 2:       19r       0w
system:       0:00:19.55   4.5%  swap in :        1
idle  :       0:06:48.30  93.2%  swap out:        0
uptime:       0:07:18.11         context :    22059

irq  0:     43811 timer                irq  9:         0 cmpci
irq  1:      1427 keyboard             irq 11:       332 nvidia
irq  2:         0 cascade [4]          irq 12:         2
irq  6:         2                      irq 14:      7251 ide0
irq  8:         3 rtc                  irq 15:        83 ide1
```

/proc/apm
: Advanced power management info.

/proc/bus
: Directory containing bus specific information.

/proc/cmdline
: Kernel command line.

/proc/cpuinfo
: Information about the processor, such as its type, make, model, and performance.

/proc/devices
: List of device drivers configured into the currently running kernel (block and character).

/proc/dma
: Shows which DMA channels are being used at the moment.

/proc/driver
: Various drivers grouped here, currently rtc

/proc/execdomains
: Execdomains, related to security.

/proc/fb
: Frame Buffer devices.

/proc/filesystems
: Filesystems configured/supported into/by the kernel.

/proc/fs
: File system parameters, currently nfs/exports.

/proc/ide
: This subdirectory contains information about all IDE devices of which the kernel is aware. There is one subdirectory for each IDE controller, the file drivers and a link for each IDE device, pointing to

the device directory in the controller−specific subtree. The file drivers contains general information about the drivers used for the IDE devices. More detailed information can be found in the controller−specific subdirectories. These are named ide0, ide1 and so on. Each of these directories contains the files shown here:

/proc/ide/ide?/channel
: IDE channel (0 or 1)

/proc/ide/ide?/config
: Configuration (only for PCI/IDE bridge)

/proc/ide/ide?/mate
: Mate name (onchip partnered controller)

/proc/ide/ide?/model
: Type/Chipset of IDE controller
: Each device connected to a controller has a separate subdirectory in the controllers directory. The following files listed are contained in these directories:

/proc/ide/ide?/model/cache
: The cache.

/proc/ide/ide?/model/capacity
: Capacity of the medium (in 512Byte blocks)

/proc/ide/ide?/model/driver
: driver and version

/proc/ide/ide?/model/geometry
: physical and logical geometry

/proc/ide/ide?/model/identify
: device identify block

/proc/ide/ide?/model/media
: media type

/proc/ide/ide?/model/model
: device identifier

/proc/ide/ide?/model/settings
: device setup

/proc/ide/ide?/model/smart_thresholds
: IDE disk management thresholds

/proc/ide/ide?/model/smart_values
: IDE disk management values

/proc/interrupts
: Shows which interrupts are in use, and how many of each there have been.
: You can, for example, check which interrupts are currently in use and what they are used for by looking in the file /proc/interrupts:

```
# cat /proc/interrupts
```

```
  CPU0 0: 8728810
  XT-PIC timer 1: 895
  XT-PIC keyboard 2:
  0 XT-PIC cascade 3: 531695
  XT-PIC aha152x 4: 2014133
  XT-PIC serial 5: 44401
  XT-PIC pcnet_cs 8: 2
  XT-PIC rtc 11: 8
  XT-PIC i82365 12: 182918
  XT-PIC PS/2 Mouse 13: 1
  XT-PIC fpu 14: 1232265
```

```
XT−PIC ide0 15: 7
XT−PIC ide1 NMI: 0
```

In 2.4 based kernels a couple of lines were added to this file LOC & ERR (this is the output of an SMP machine):

```
# cat /proc/interrupts
```

```
CPU0 CPU1
0: 1243498 1214548 IO−APIC−edge timer
1: 8949 8958 IO−APIC−edge keyboard
2: 0 0 XT−PIC cascade
5: 11286 10161 IO−APIC−edge soundblaster
8: 1 0 IO−APIC−edge rtc
9: 27422 27407 IO−APIC−edge 3c503
12: 113645 113873 IO−APIC−edge PS/2 Mouse
13: 0 0 XT−PIC fpu 14: 22491 24012 IO−APIC−edge ide0
15: 2183 2415 IO−APIC−edge ide1
17: 30564 30414 IO−APIC−level eth0
18: 177 164 IO−APIC−level bttv NMI: 2457961 2457959
            LOC: 2457882 2457881 ERR: 2155
```

NMI is incremented in this case because every timer interrupt generates a NMI (Non Maskable Interrupt) which is used by the NMI Watchdog to detect lookups.
LOC is the local interrupt counter of the internal APIC of every CPU.
ERR is incremented in the case of errors in the IO−APIC bus (the bus that connects the CPUs in an SMP system. This means that an error has been detected, the IO−APIC automatically retries the transmission, so it should not be a big problem, but you should read the SMP−FAQ.
In this context it could be interesting to note the new irq directory in 2.4. It could be used to set IRQ to CPU affinity, this means that you can "hook" an IRQ to only one CPU, or to exclude a CPU from handling IRQs. The contents of the irq subdir is one subdir for each IRQ, and one file; prof_cpu_mask. For example,

```
# ls /proc/irq/ 0 10 12 14 16 18 2 4 6 8 prof_cpu_mask
                1 11 13 15 17 19 3 5 7 9
```

```
# ls /proc/irq/0/ smp_affinity
```

The contents of the prof_cpu_mask file and each smp_affinity file for each IRQ is the same by default:

```
# cat /proc/irq/0/smp_affinity
ffffffff
```

It's a bitmask, in which you can specify which CPUs can handle the IRQ, you can set it by doing:

```
# echo 1 > /proc/irq/prof_cpu_mask
```

This means that only the first CPU will handle the IRQ, but you can also echo 5 which means that only the first and fourth CPU can handle the IRQ. The way IRQs are routed is handled by the IO−APIC, and its Round Robin between all the CPUs which are allowed to handle it. As usual the kernel has more info than you and does a better job than you, so the defaults are the best choice for almost everyone.

/proc/iomem
: Memory map.

/proc/ioports
: Which I/O ports are in use at the moment.

/proc/irq
: Masks for irq to cpu affinity.

/proc/isapnp
: ISA PnP (Plug&Play) Info.

/proc/kcore
: An image of the physical memory of the system (can be ELF or A.OUT (deprecated in 2.4)). This is exactly the same size as your physical memory, but does not really take up that much memory; it is generated on the fly as programs access it. (Remember: unless you copy it elsewhere, nothing under /proc takes up any disk space at all.)

/proc/kmsg
: Messages output by the kernel. These are also routed to syslog.

/proc/ksyms
: Kernel symbol table.

/proc/loadavg
: The 'load average' of the system; three indicators of how much work the system has done during the last 1, 5 & 15 minutes.

/proc/locks
: Kernel locks.

/proc/meminfo
: Information about memory usage, both physical and swap. Concatenating this file produces similar results to using 'free' or the first few lines of 'top'.

/proc/misc
: Miscellaneous pieces of information. This is for information that has no real place within the rest of the proc filesystem.

/proc/modules
: Kernel modules currently loaded. Typically its output is the same as that given by the 'lsmod' command.

/proc/mounts
: Mounted filesystems

/proc/mtrr
: Information regarding mtrrs. (On Intel P6 family processors (Pentium Pro, Pentium II and later) the Memory Type Range Registers (MTRRs) may be used to control processor access to memory ranges. This is most useful when you have a video (VGA) card on a PCI or AGP bus. Enabling write−combining allows bus write transfers to be combined into a larger transfer before bursting over the PCI/AGP bus. This can increase performance of image write operations 2.5 times or more. The Cyrix 6x86, 6x86MX and M II processors have Address Range Registers (ARRs) which provide a similar functionality to MTRRs. For these, the ARRs are used to emulate the MTRRs. The AMD K6−2 (stepping 8 and above) and K6−3 processors have two MTRRs. These are supported. The AMD Athlon family provide 8 Intel style MTRRs. The Centaur C6 (WinChip) has 8 MCRs, allowing write−combining. These are also supported. The VIA Cyrix III and VIA C3 CPUs offer 8 Intel style MTRRs.) For more details regarding mtrr technology see /usr/src/linux/Documentation/mtrr.txt.

/proc/net
: Status information about network protocols.

IPv6 information

/proc/net/udp6

UDP sockets (IPv6).

/proc/net/tcp6
: TCP sockets (IPv6).

/proc/net/raw6
: Raw device statistics (IPv6).

/proc/net/igmp6
: IP multicast addresses, which this host joined (IPv6).

/proc/net/if_inet6
: List of IPv6 interface addresses.

/proc/net/ipv6_route
: Kernel routing table for IPv6.

/proc/net/rt6_stats
: Global IPv6 routing tables statistics.

/proc/net/sockstat6
: Socket statistics (IPv6).

/proc/net/snmp6
: Snmp data (IPv6).

General Network information

/proc/net/arp
: Kernel ARP table.

/proc/net/dev
: network devices with statistics.

/proc/net/dev_mcast
: the Layer2 multicast groups which a device is listening to (interface index, label, number of references, number of bound addresses).

/proc/net/dev_stat
: network device status.

/proc/net/ip_fwchains
: Firewall chain linkage.

/proc/net/ip_fwnames
: Firewall chain names.

/proc/net/ip_masq
: Directory containing the masquerading tables.

/proc/net/ip_masquerade
: Major masquerading table.

/proc/net/netstat
: Network statistics.

/proc/net/raw
: raw device statistics.

/proc/net/route
: Kernel routing table.

/proc/net/rpc
: Directory containing rpc info.

/proc/net/rt_cache
: Routing cache.

/proc/net/snmp
: SNMP data.

/proc/net/sockstat
: Socket statistics.

/proc/net/tcp

TCP sockets.

/proc/net/tr_rif
: Token ring RIF routing table.

/proc/net/udp
: UDP sockets.

/proc/net/unix
: UNIX domain sockets.

/proc/net/wireless
: Wireless interface data (Wavelan etc).

/proc/net/igmp
: IP multicast addresses, which this host joined.

/proc/net/psched
: Global packet scheduler parameters.

/proc/net/netlink
: List of PF_NETLINK sockets.

/proc/net/ip_mr_vifs
: List of multicast virtual interfaces.

/proc/net/ip_mr_cache
: List of multicast routing cache.

You can use this information to see which network devices are available in your system and how much traffic was routed over those devices. In addition, each Channel Bond interface has its own directory. For example, the bond0 device will have a directory called /proc/net/bond0/. It will contain information that is specific to that bond, such as the current slaves of the bond, the link status of the slaves, and how many times the slaves link has failed.

/proc/parport
: The directory /proc/parport contains information about the parallel ports of your system. It has one subdirectory for each port, named after the port number (0,1,2,...).

/proc/parport/autoprobe
: Any IEEE−1284 device ID information that has been acquired.

/proc/parport/devices
: list of the device drivers using that port. A + will appear by the name of the device currently using the port (it might not appear against any).

/proc/parport/hardware
: Parallel port's base address, IRQ line and DMA channel.

/proc/parport/irq
: IRQ that parport is using for that port. This is in a separate file to allow you to alter it by writing a new value in (IRQ number or none).

/proc/partitions
: Table of partitions known to the system

/proc/pci, /proc/bus/pci
: Depreciated info of PCI bus.

/proc/rtc
: Real time clock

/proc/scsi
: If you have a SCSI host adapter in your system, you'll find a subdirectory named after the driver for this adapter in /proc/scsi. You'll also see a list of all recognized SCSI devices in /proc/scsi. The directory named after the driver has one file for each adapter found in the system. These files contain information about the controller, including the used IRQ and the IO address range. The amount of information shown is dependent on the adapter you use.

/proc/self

A symbolic link to the process directory of the program that is looking at /proc. When two processes look at /proc, they get different links. This is mainly a convenience to make it easier for programs to get at their process directory.

/proc/slabinfo

The slabinfo file gives information about memory usage at the slab level. Linux uses slab pools for memory management above page level in version 2.2. Commonly used objects have their own slab pool (such as network buffers, directory cache, and so on).

/proc/stat

Overall/various statistics about the system, such as the number of page faults since the system was booted.

/proc/swaps

Swap space utilization

/proc/sys

This is not only a source of information, it also allows you to change parameters within the kernel without the need for recompilation or even a system reboot. Take care when attempting this as it can both optimize your system and also crash it. It is advisable to read both documentation and source before actually making adjustments. The entries in /proc may change slightly between kernel versions, so if there is any doubt review the kernel documentation in the directory /usr/src/linux/Documentation. Under some circumstances, you may have no alternative but to reboot the machine once an error occurs. To change a value, simply echo the new value into the file. An example is given below in the section on the file system data. Of course, you need to be 'root' to do any of this. You can create your own boot script to perform this every time your system boots.

/proc/sys/fs

Contains file system data. This subdirectory contains specific file system, file handle, inode, dentry and quota information.

dentry−state

Status of the directory cache. Since directory entries are dynamically allocated and deallocated, this file indicates the current status. It holds six values, in which the last two are not used and are always zero. The others are listed below:

```
File      Content
nr_dentry Almost always zero
nr_unused Number of unused cache entries
age_limit in seconds after the entry may be
          reclaimed, when memory is short want_pages internally
```

dquot−max

The file dquot−max shows the maximum number of cached disk quota entries.

dquot−nr

shows the number of allocated disk quota entries and the number of free disk quota entries. If the number of available cached disk quotas is very low and you have a large number of simultaneous system users, you might want to raise the limit.

file−nr and file−max

The kernel allocates file handles dynamically, but doesn't free them again at this time. The value in file−max denotes the maximum number of file handles that the Linux kernel will allocate. When you get a lot of error messages about running out of file handles, you might want to raise this limit. The default value is 4096. To change it, just write the new number into the file:

```
# cat /proc/sys/fs/file-max
4096
# echo 8192 > /proc/sys/fs/file-max
# cat /proc/sys/fs/file-max
8192
```

This method of revision is useful for all customizable parameters of the kernel − simply echo the new value to the corresponding file.
The three values in file−nr denote the number of allocated file handles, the number of used file handles, and the maximum number of file handles. When the allocated file handles come close to the maximum, but the number of actually used handles is far behind, you've encountered a peak in your usage of file handles and you don't need to increase the maximum.

inode−state, inode−nr and inode−max
: As with file handles, the kernel allocates the inode structures dynamically, but can't free them yet.
The value in inode−max denotes the maximum number of inode handlers. This value should be 3 to 4 times larger than the value in file−max, since stdin, stdout, and network sockets also need an inode struct to handle them. If you regularly run out of inodes, you should increase this value.
The file inode−nr contains the first two items from inode−state, so we'll skip to that file...
inode−state contains three actual numbers and four dummy values. The numbers are nr_inodes, nr_free_inodes, and preshrink (in order of appearance).

nr_inodes
: Denotes the number of inodes the system has allocated. This can be slightly more than inode−max because Linux allocates them one pageful at a time.

nr_free_inodes
: Represents the number of free inodes and preshrink is nonzero when nr_inodes is greater than inode−max and the system needs to prune the inode list instead of allocating more.

super−nr and super−max
: Again, super block structures are allocated by the kernel, but not freed. The file super−max contains the maximum number of super block handlers, where super−nr shows the number of currently allocated ones. Every mounted file system needs a super block, so if you plan to mount lots of file systems, you may want to increase these numbers.

binfmt_misc
: This handles the kernel support for miscellaneous binary formats. binfmt_misc provides the ability to register additional binary formats to the kernel without compiling an additional module/kernel. Therefore, binfmt_misc needs to know magic numbers at the beginning or the filename extension of the binary. It works by maintaining a linked list of structs that contain a description of a binary format, including a magic with size (or the filename extension), offset and mask, and the interpreter name. On request it invokes the given interpreter with the original program as argument, as binfmt_java and binfmt_em86 and binfmt_mz do. Since binfmt_misc does not define any default binary−formats, you have to register an additional binary−format. There are two general files in binfmt_misc and one file per registered format. The two general files are register and status. To register a new binary format you have to issue the command echo :name:type:offset:magic:mask:interpreter: > /proc/sys/fs/binfmt_misc/register with appropriate name (the name for the /proc−dir entry), offset (defaults to 0, if omitted), magic, mask (which can be omitted, defaults to all 0xff) and last but not least, the interpreter that is to be invoked (for example and testing /bin/echo). Type can be M for usual magic matching or E for filename extension matching (give extension in place of magic). If you do a cat on the file /proc/sys/fs/binfmt_misc/status, you will get the current status (enabled/disabled) of binfmt_misc. Change the status by echoing 0 (disables) or 1 (enables) or −1 (caution: this clears all previously registered binary formats) to

status. For example echo 0 > status to disable binfmt_misc (temporarily). Each registered handler has an entry in /proc/sys/fs/binfmt_misc. These files perform the same function as status, but their scope is limited to the actual binary format. By 'cating' this file, you also receive all related information about the interpreter/magic of the binfmt. An example of the usage of binfmt_misc (emulate binfmt_java) follows:

```
cd /proc/sys/fs/binfmt_misc
echo ':Java:M::\xca\xfe\xba\xbe::/usr/local/java/bin/javawrapper:'
> register
echo ':HTML:E::html::/usr/local/java/bin/appletviewer:'
> register
echo ':Applet:M::<!--applet::/usr/local/java/bin/appletviewer:' >
register
echo ':DEXE:M::\x0eDEX::/usr/bin/dosexec:' < register
```

These four lines add support for Java executables and Java applets (like binfmt_java, additionally recognizing the .html extension with no need to put <!−−applet> to every applet file). You have to install the JDK and the shell−script /usr/local/java/bin/javawrapper too. It works around the brokenness of the Java filename handling. To add a Java binary, just create a link to the class−file somewhere in the path.

/proc/sys/kernel

This directory reflects general kernel behaviors and the contents will be dependent upon your configuration. Here you'll find the most important files, along with descriptions of what they mean and how to use them.

/proc/sys/kernel/acct

The file contains three values; highwater, lowwater, and frequency. It exists only when BSD−style process accounting is enabled. These values control its behavior. If the free space on the file system where the log lives goes below lowwater percentage, accounting suspends. If it goes above highwater percentage, accounting resumes. Frequency determines how often you check the amount of free space (value is in seconds). Default settings are: 4, 2, and 30. That is, suspend accounting if there is less than 2 percent free; resume it if we have a value of 3 or more percent; consider information about the amount of free space valid for 30 seconds

/proc/sys/kernel/ctrl−alt−del

When the value in this file is 0, ctrl−alt−del is trapped and sent to the init program to handle a graceful restart. However, when the value is greater that zero, Linux's reaction to this key combination will be an immediate reboot, without syncing its dirty buffers. It should be noted that when a program (like dosemu) has the keyboard in raw mode, the ctrl−alt−del is intercepted by the program before it ever reaches the kernel tty layer, and it is up to the program to decide what to do with it.

/proc/sys/kernel/domainname, /proc/sys/kernel/hostname

These files can be controlled to set the NIS domainname and hostname of your box. For the classic darkstar.frop.org a simple: # echo "darkstar" > /proc/sys/kernel/hostname # echo "frop.org" > /proc/sys/kernel/domainname would suffice to set your hostname and NIS domainname. /proc/sys/kernel/osrelease, /proc/sys/kernel/ostype, /proc/sys/kernel/version The names make it pretty obvious what these fields contain: # cat /proc/sys/kernel/osrelease 2.2.12 # cat /proc/sys/kernel/ostype Linux # cat /proc/sys/kernel/version #4 Fri Oct 1 12:41:14 PDT 1999 The files osrelease and ostype should be clear enough. Version needs a little more clarification. The #4 means that this is the 4th kernel built from this source base and the date after it indicates the time the kernel was built. The only way to tune these values is to rebuild the kernel.

/proc/sys/kernel/panic

The value in this file represents the number of seconds the kernel waits before rebooting on a panic. When you use the software watchdog, the recommended setting is 60. If set to 0, the auto reboot after a kernel panic is disabled, which is the default setting.

/proc/sys/kernel/printk
: The four values in printk denote * console_loglevel, * default_message_loglevel, * minimum_console_level and * default_console_loglevel respectively. These values influence printk() behavior when printing or logging error messages, which come from inside the kernel. See syslog(2) for more information on the different log levels.

/proc/sys/kernel/console_loglevel
: Messages with a higher priority than this will be printed to the console.

/proc/sys/kernel/default_message_level
: Messages without an explicit priority will be printed with this priority.

/proc/sys/kernel/minimum_console_loglevel
: Minimum (highest) value to which the console_loglevel can be set.

/proc/sys/kernel/default_console_loglevel
: Default value for console_loglevel.

/proc/sys/kernel/sg−big−buff
: This file shows the size of the generic SCSI (sg) buffer. At this point, you can't tune it yet, but you can change it at compile time by editing include/scsi/sg.h and changing the value of SG_BIG_BUFF. If you use a scanner with SANE (Scanner Access Now Easy) you might want to set this to a higher value. Refer to the SANE documentation on this issue.

/proc/sys/kernel/modprobe
: The location where the modprobe binary is located. The kernel uses this program to load modules on demand.

/proc/sys/vm
: The files in this directory can be used to tune the operation of the virtual memory (VM) subsystem of the Linux kernel. In addition, one of the files (bdflush) has some influence on disk usage.

nfract
: This parameter governs the maximum number of dirty buffers in the buffer cache. Dirty means that the contents of the buffer still have to be written to disk (as opposed to a clean buffer, which can just be forgotten about). Setting this to a higher value means that Linux can delay disk writes for a long time, but it also means that it will have to do a lot of I/O at once when memory becomes short. A lower value will spread out disk I/O more evenly.

ndirty
: Ndirty gives the maximum number of dirty buffers that bdflush can write to the disk at one time. A high value will mean delayed, bursty I/O, while a small value can lead to memory shortage when bdflush isn't woken up often enough.

nrefill
: This is the number of buffers that bdflush will add to the list of free buffers when refill_freelist() is called. It is necessary to allocate free buffers beforehand, since the buffers are often different sizes than the memory pages and some bookkeeping needs to be done beforehand. The higher the number, the more memory will be wasted and the less often refill_freelist() will need to run.

nref_dirt
: When refill_freelist() comes across more than nref_dirt dirty buffers, it will wake up bdflush.

age_buffer, age_super
: Finally, the age_buffer and age_super parameters govern the maximum time Linux waits before writing out a dirty buffer to disk. The value is expressed in jiffies (clockticks), the number of jiffies per second is 100. Age_buffer is the maximum age for data blocks, while age_super is for filesystems meta data.

buffermem

The three values in this file control how much memory should be used for buffer memory. The percentage is calculated as a percentage of total system memory.

The values are:

min_percent
: This is the minimum percentage of memory that should be spent on buffer memory.

borrow_percent
: When Linux is short on memory, and the buffer cache uses more than it has been allotted, the memory management (MM) subsystem will prune the buffer cache more heavily than other memory to compensate.

max_percent
: This is the maximum amount of memory that can be used for buffer memory.

freepages

This file contains three values: min, low and high:

min
: When the number of free pages in the system reaches this number, only the kernel can allocate more memory.

low
: If the number of free pages falls below this point, the kernel starts swapping aggressively.

high
: The kernel tries to keep up to this amount of memory free; if memory falls below this point, the kernel starts gently swapping in the hopes that it never has to do really aggressive swapping.

kswapd

Kswapd is the kernel swap out daemon. That is, kswapd is that piece of the kernel that frees memory when it gets fragmented or full. Since every system is different, you'll probably want some control over this piece of the system.

The file contains three numbers:

tries_base
: The maximum number of pages kswapd tries to free in one round is calculated from this number. Usually this number will be divided by 4 or 8 (see mm/vmscan.c), so it isn't as big as it looks. When you need to increase the bandwidth to/from swap, you'll want to increase this number.

tries_min
: This is the minimum number of times kswapd tries to free a page each time it is called. Basically it's just there to make sure that kswapd frees some pages even when it's being called with minimum priority.

swap_cluster
: This is probably the greatest influence on system performance. swap_cluster is the number of pages kswapd writes in one turn. You'll want this value to be large so that kswapd does its I/O in large chunks and the disk doesn't have to seek as often, but you don't want it to be too large since that would flood the request queue.

overcommit_memory

This file contains one value. The following algorithm is used to decide if there's enough memory: if the value of overcommit_memory is positive, then there's always enough memory. This is a useful feature, since programs often malloc() huge amounts of memory 'just in case', while they only use a small part of it. Leaving this value at 0 will lead to the failure of such a huge malloc(), when in fact the system has enough memory for the program to run. On the other hand, enabling this feature can cause you to run out of memory and thrash the system to death, so large and/or important servers will want to set this value to 0.

pagecache

This file does exactly the same job as buffermem, only this file controls the amount of memory allowed for memory mapping and generic caching of files. You don't want the minimum level to be too low, otherwise your system might thrash when memory is tight or fragmentation is high.

pagetable_cache

The kernel keeps a number of page tables in a per−processor cache (this helps a lot on SMP systems). The cache size for each processor will be between the low and the high value. On a low−memory, single CPU system, you can safely set these values to 0 so you don't waste memory. It is used on SMP systems so that the system can perform fast pagetable allocations without having to acquire the kernel memory lock. For large systems, the settings are probably fine. For normal systems they won't hurt a bit. For small systems (less than 16MB ram) it might be advantageous to set both values to 0.

swapctl

This file contains no less than 8 variables. All of these values are used by kswapd. The first four variables sc_max_page_age, sc_page_advance, sc_page_decline and sc_page_initial_age are used to keep track of Linux's page aging. Page ageing is a bookkeeping method to track which pages of memory are often used, and which pages can be swapped out without consequences.

When a page is swapped in, it starts at sc_page_initial_age (default 3) and when the page is scanned by kswapd, its age is adjusted according to the following scheme.

If the page was used since the last time we scanned, its age is increased by sc_page_advance (default 3). Where the maximum value is given by sc_max_page_age (default 20). Otherwise (meaning it wasn't used) its age is decreased by sc_page_decline (default 1).

When a page reaches age 0, it's ready to be swapped out.

The variables sc_age_cluster_fract, sc_age_cluster_min, sc_pageout_weight and sc_bufferout_weight, can be used to control kswapd's aggressiveness in swapping out pages.

Sc_age_cluster_fract is used to calculate how many pages from a process are to be scanned by kswapd. The formula used is

(sc_age_cluster_fract divided by 1024) times resident set size

So if you want kswapd to scan the whole process, sc_age_cluster_fract needs to have a value of 1024. The minimum number of pages kswapd will scan is represented by sc_age_cluster_min, which is done so that kswapd will also scan small processes. The values of sc_pageout_weight and sc_bufferout_weight are used to control how many tries kswapd will make in order to swap out one page/buffer. These values can be used to fine−tune the ratio between user pages and buffer/cache memory. When you find that your Linux system is swapping out too many process pages in order to satisfy buffer memory demands, you may want to either increase sc_bufferout_weight, or decrease the value of sc_pageout_weight.

/proc/sys/dev

Device specific parameters. Currently there is only support for CDROM drives, and for those, there is only one read−only file containing information about the CD−ROM drives attached to the system: >cat /proc/sys/dev/cdrom/info CD−ROM information, Id: cdrom.c 2.55 1999/04/25 drive name: sr0 hdb drive speed: 32 40 drive # of slots: 1 0 Can close tray: 1 1 Can open tray: 1 1 Can lock tray: 1 1 Can change speed: 1 1 Can select disk: 0 1 Can read multisession: 1 1 Can read MCN: 1 1 Reports media changed: 1 1 Can play audio: 1 1 You see two drives, sr0 and hdb, along with a list of their features.

SUNRPC

/proc/sys/sunrpc

This directory contains four files, which enable or disable debugging for the RPC functions NFS, NFS−daemon, RPC and NLM. The default values are 0. They can be set to one to turn debugging on. (The default value is 0 for each)

/proc/sys/net

The interface to the networking parts of the kernel is located in /proc/sys/net. The following table shows all possible subdirectories. You may see only some of them, depending on your kernel's configuration. Our main focus will be on IP networking since AX15, X.25, and DEC Net are only minor players in the Linux world. Should you wish review the online documentation and the kernel source to get a detailed view of the parameters for those protocols not covered here. In this section we'll discuss the subdirectories listed above. As default values are suitable for most needs, there is no need to change these values.

GENERAL PARAMETERS

/proc/sys/net/core
: Network core options

rmem_default
: The default setting of the socket receive buffer in bytes.

rmem_max
: The maximum receive socket buffer size in bytes.

wmem_default
: The default setting (in bytes) of the socket send buffer.

wmem_max
: The maximum send socket buffer size in bytes.

message_burst and message_cost
: These parameters are used to limit the warning messages written to the kernel log from the networking code. They enforce a rate limit to make a denial−of−service attack impossible. A higher message_cost factor, results in fewer messages that will be written. Message_burst controls when messages will be dropped. The default settings limit warning messages to one every five seconds.

netdev_max_backlog
: Maximum number of packets, queued on the INPUT side, when the interface receives packets faster than kernel can process them.

optmem_max
: Maximum ancillary buffer size allowed per socket. Ancillary data is a sequence of struct cmsghdr structures with appended data.

UNIX DOMAIN SOCKETS

/proc/sys/net/unix
: Parameters for Unix domain sockets
: There are only two files in this subdirectory. They control the delays for deleting and destroying socket descriptors.

IPv4

/proc/sys/net/ipv4
: IPV4 settings. IP version 4 is still the most used protocol in Unix networking. It will be replaced by IP version 6 in the next couple of years, but for the moment it's the de facto standard for the internet and is used in most networking environments around the world. Because of the importance of this protocol, we'll have a deeper look into the subtree controlling the behavior of the Ipv4 subsystem of the Linux kernel.
: Let's start with the entries in /proc/sys/net/ipv4.

ICMP settings

icmp_echo_ignore_all and icmp_echo_ignore_broadcasts
: Turn on (1) or off (0), if the kernel should ignore all ICMP ECHO requests, or just those to broadcast and multicast addresses.

Please note that if you accept ICMP echo requests with a broadcast/multi\−cast destination address your network may be used as an exploder for denial of service packet flooding attacks to other hosts.

icmp_destunreach_rate, icmp_echoreply_rate, icmp_paramprob_rate and icmp_timeexeed_rate
: Sets limits for sending ICMP packets to specific targets. A value of zero disables all limiting. Any positive value sets the maximum package rate in hundredth of a second (on Intel systems).

IP settings

ip_autoconfig
: This file contains the number one if the host received its IP configuration by RARP, BOOTP, DHCP or a similar mechanism. Otherwise it is zero.

ip_default_ttl
: TTL (Time To Live) for IPv4 interfaces. This is simply the maximum number of hops a packet may travel.

ip_dynaddr
: Enable dynamic socket address rewriting on interface address change. This is useful for dialup interface with changing IP addresses.

ip_forward
: Enable or disable forwarding of IP packages between interfaces. Changing this value resets all other parameters to their default values. They differ if the kernel is configured as host or router.

ip_local_port_range
: Range of ports used by TCP and UDP to choose the local port. Contains two numbers, the first number is the lowest port, the second number the highest local port. Default is 1024−4999. Should be changed to 32768−61000 for high−usage systems.

ip_no_pmtu_disc
: Global switch to turn path MTU discovery off. It can also be set on a per socket basis by the applications or on a per route basis.

ip_masq_debug
: Enable/disable debugging of IP masquerading.

IP fragmentation settings

ipfrag_high_trash and ipfrag_low_trash
: Maximum memory used to reassemble IP fragments. When ipfrag_high_thrash bytes of memory is allocated for this purpose, the fragment handler will toss packets until ipfrag_low_thrash is reached.

ipfrag_time
: Time in seconds to keep an IP fragment in memory.

TCP settings

tcp_ecn
: This file controls the use of the ECN bit in the IPv4 headers, this is a new feature about Explicit Congestion Notification, but some routers and firewalls block traffic that has this bit set, so it could be necessary to echo 0 to /proc/sys/net/ipv4/tcp_ecn, if you want to talk to this sites. For more info you could read RFC2481.

tcp_retrans_collapse
: Bug−to−bug compatibility with some broken printers. On retransmit, try to send larger packets to work around bugs in certain TCP stacks. Can be turned off by setting it to zero.

tcp_keepalive_probes
: Number of keep alive probes TCP sends out, until it decides that the connection is broken.

tcp_keepalive_time
: How often TCP sends out keep alive messages, when keep alive is enabled. The default is 2 hours.

tcp_syn_retries
: Number of times initial SYNs for a TCP connection attempt will be retransmitted. Should not be higher than 255. This is only the timeout for outgoing connections, for incoming connections the number of retransmits is defined by tcp_retries1.

tcp_sack
: Enable select acknowledgments after RFC2018.

tcp_timestamps
: Enable timestamps as defined in RFC1323.

tcp_stdurg
: Enable the strict RFC793 interpretation of the TCP urgent pointer field. The default is to use the BSD compatible interpretation of the urgent pointer pointing to the first byte after the urgent data. The RFC793 interpretation is to have it point to the last byte of urgent data. Enabling this option may lead to interoperability problems. Disabled by default.

tcp_syncookies
: Only valid when the kernel was compiled with CONFIG_SYNCOOKIES. Send out syncookies when the syn backlog queue of a socket overflows. This is to ward off the common 'syn flood attack'. Disabled by default. Note that the concept of a socket backlog is abandoned. This means the peer may not receive reliable error messages from an over loaded server with syncookies enabled.

tcp_window_scaling
: Enable window scaling as defined in RFC1323.

tcp_fin_timeout
: The length of time in seconds it takes to receive a final FIN before the socket is always closed. This is strictly a violation of the TCP specification, but required to prevent denial−of−service attacks.

tcp_max_ka_probes
: Indicates how many keep alive probes are sent per slow timer run. Should not be set too high to prevent bursts.

tcp_max_syn_backlog
: Length of the per socket backlog queue. Since Linux 2.2 the backlog specified in listen(2) only specifies the length of the backlog queue of already established sockets. When more connection requests arrive Linux starts to drop packets. When syncookies are enabled the packets are still answered and the maximum queue is effectively ignored.

tcp_retries1
: Defines how often an answer to a TCP connection request is retransmitted before giving up.

tcp_retries2
: Defines how often a TCP packet is retransmitted before giving up.

/proc/sys/net/ipv4/conf
: Here you'll find one subdirectory for each interface the system knows about and one directory called all. Changes in the all subdirectory affect all interfaces, whereas changes in the other subdirectories affect only one interface. All directories have the same entries:

accept_redirects
: This switch decides if the kernel accepts ICMP redirect messages or not. The default is 'yes' if the kernel is configured for a regular host and 'no' for a router configuration.

accept_source_route
: Should source routed packages be accepted or declined. The default is dependent on the kernel configuration. It's 'yes' for routers and 'no' for hosts.

bootp_relay

Accept packets with source address 0.b.c.d with destinations not to this host as local ones. It is supposed that a BOOTP relay daemon will catch and forward such packets. The default is 0.

forwarding
: Enable or disable IP forwarding on this interface.

log_martians
: Log packets with source addresses with no known route to kernel log.

mc_forwarding
: Do multicast routing. The kernel needs to be compiled with CONFIG_MROUTE and a multicast routing daemon is required.

proxy_arp
: Does (1) or does not (0) perform proxy ARP.

rp_filter
: Integer value determines if a source validation should be made. 1 means yes, 0 means no. Disabled by default, but local/broadcast address spoofing is always on. If you set this to 1 on a router that is the only connection for a network to the net, it will prevent spoofing attacks against your internal networks (external addresses can still be spoofed), without the need for additional firewall rules.

secure_redirects
: Accept ICMP redirect messages only for gateways, listed in default gateway list. Enabled by default.

shared_media
: If it is not set the kernel does not assume that different subnets on this device can communicate directly. Default setting is 'yes'.

send_redirects
: Determines whether to send ICMP redirects to other hosts.

Routing settings
: The directory /proc/sys/net/ipv4/route contains several file to control routing issues.

error_burst and error_cost
: These parameters are used to limit the warning messages written to the kernel log from the routing code. The higher the error_cost factor is, the fewer messages will be written. Error_burst controls when messages will be dropped. The default settings limit warning messages to one every five seconds.

flush
: Writing to this file results in a flush of the routing cache.

gc_elastic, gc_interval, gc_min_interval, gc_tresh, gc_timeout
: Values to control the frequency and behavior of the garbage collection algorithm for the routing cache.

max_size
: Maximum size of the routing cache. Old entries will be purged once the cache reached has this size.

max_delay, min_delay
: Delays for flushing the routing cache.

redirect_load, redirect_number
: Factors which determine if more ICPM redirects should be sent to a specific host. No redirects will be sent once the load limit or the maximum number of redirects has been reached.

redirect_silence
: Timeout for redirects. After this period redirects will be sent again, even if this has been stopped, because the load or number limit has been reached.

/proc/sys/net/ipv4/neigh

Network Neighbor handling. It contains settings about how to handle connections with direct neighbors (nodes attached to the same link). As we saw it in the conf directory, there is a default subdirectory which holds the default values, and one directory for each interface. The contents of the directories are identical, with the single exception that the default settings contain additional options to set garbage collection parameters.

In the interface directories you'll find the following entries:

base_reachable_time
: A base value used for computing the random reachable time value as specified in RFC2461.

retrans_time
: The time, expressed in jiffies (1/100 sec), between retransmitted Neighbor Solicitation messages. Used for address resolution and to determine if a neighbor is unreachable.

unres_qlen
: Maximum queue length for a pending arp request − the number of packets which are accepted from other layers while the ARP address is still resolved.

anycast_delay
: Maximum for random delay of answers to neighbor solicitation messages in jiffies (1/100 sec). Not yet implemented (Linux does not have anycast support yet).

ucast_solicit
: Maximum number of retries for unicast solicitation.

mcast_solicit
: Maximum number of retries for multicast solicitation.

delay_first_probe_time
: Delay for the first time probe if the neighbor is reachable. (see gc_stale_time)

locktime
: An ARP/neighbor entry is only replaced with a new one if the old is at least locktime old. This prevents ARP cache thrashing.

proxy_delay
: Maximum time (real time is random [0..proxytime]) before answering to an ARP request for which we have an proxy ARP entry. In some cases, this is used to prevent network flooding.

proxy_qlen
: Maximum queue length of the delayed proxy arp timer. (see proxy_delay).

app_solcit
: Determines the number of requests to send to the user level ARP daemon. Use 0 to turn off.

gc_stale_time
: Determines how often to check for stale ARP entries. After an ARP entry is stale it will be resolved again (which is useful when an IP address migrates to another machine). When ucast_solicit is greater than 0 it first tries to send an ARP packet directly to the known host When that fails and mcast_solicit is greater than 0, an ARP request is broadcasted.

APPLETALK

/proc/sys/net/appletalk
: Holds the Appletalk configuration data when Appletalk is loaded. The configurable parameters are:

aarp−expiry−time
: The amount of time we keep an ARP entry before expiring it. Used to age out old hosts.

aarp−resolve−time
: The amount of time we will spend trying to resolve an Appletalk address.

aarp−retransmit−limit
: The number of times we will retransmit a query before giving up.

aarp−tick−time
: Controls the rate at which expires are checked.

/proc/net/appletalk
: Holds the list of active Appletalk sockets on a machine. The fields indicate the DDP type, the local address (in network:node format) the remote address, the size of the transmit pending queue, the size of the received queue (bytes waiting for applications to read) the state and the uid owning the socket.

/proc/net/atalk_iface
: lists all the interfaces configured for appletalk. It shows the name of the interface, its Appletalk address, the network range on that address (or network number for phase 1 networks), and the status of the interface.

/proc/net/atalk_route
: lists each known network route. It lists the target (network) that the route leads to, the router (may be directly connected), the route flags, and the device the route is using.

IPX

The IPX protocol has no tunable values in proc/sys/net, it does, however, provide proc/net/ipx. This lists each IPX socket giving the local and remote addresses in Novell format (that is network:node:port). In accordance with the strange Novell tradition, everything but the port is in hex. Not_Connected is displayed for sockets that are not tied to a specific remote address. The Tx and Rx queue sizes indicate the number of bytes pending for transmission and reception. The state indicates the state the socket is in and the uid is the owning uid of the socket.

ipx_interface
: Lists all IPX interfaces. For each interface it gives the network number, the node number, and indicates if the network is the primary network. It also indicates which device it is bound to (or Internal for internal networks) and the Frame Type if appropriate. Linux supports 802.3, 802.2, 802.2 SNAP and DIX (Blue Book) ethernet framing for IPX.

ipx_route
: Table holding a list of IPX routes. For each route it gives the destination network, the router node (or Directly) and the network address of the router (or Connected) for internal networks.

/proc/sysvipc
: Info of SysVIPC Resources (msg, sem, shm) (2.4)

/proc/tty
: Information about the available and actually used tty's can be found in the directory /proc/tty. You'll find entries for drivers and line disciplines in this directory.

/proc/tty/drivers
: list of drivers and their usage.

/proc/tty/ldiscs
: registered line disciplines.

/proc/tty/driver/serial
: usage statistic and status of single tty lines.

To see which tty's are currently in use, you can simply look into the file /proc/tty/drivers:

```
# cat /proc/tty/drivers
serial               /dev/cua        5   64-127 serial:callout
serial               /dev/ttyS       4   64-127 serial
pty_slave            /dev/pts      143    0-255 pty:slave
pty_master           /dev/ptm      135    0-255 pty:master
pty_slave            /dev/pts      142    0-255 pty:slave
pty_master           /dev/ptm      134    0-255 pty:master
pty_slave            /dev/pts      141    0-255 pty:slave
pty_master           /dev/ptm      133    0-255 pty:master
pty_slave            /dev/pts      140    0-255 pty:slave
pty_master           /dev/ptm      132    0-255 pty:master
pty_slave            /dev/pts      139    0-255 pty:slave
```

```
pty_master          /dev/ptm      131   0-255 pty:master
pty_slave           /dev/pts      138   0-255 pty:slave
pty_master          /dev/ptm      130   0-255 pty:master
pty_slave           /dev/pts      137   0-255 pty:slave
pty_master          /dev/ptm      129   0-255 pty:master
pty_slave           /dev/pts      136   0-255 pty:slave
pty_master          /dev/ptm      128   0-255 pty:master
pty_slave           /dev/ttyp       3   0-255 pty:slave
pty_master          /dev/pty        2   0-255 pty:master
/dev/vc/0           /dev/vc/0       4       0 system:vtmaster
/dev/ptmx           /dev/ptmx       5       2 system
/dev/console        /dev/console    5       1 system:console
/dev/tty            /dev/tty        5       0 system:/dev/tty
unknown             /dev/vc/%d      4    1-63 console
```

Note that while the above files tend to be easily readable text files, they can sometimes be formatted in a way that is not easily digestible. There are many commands that do little more than read the above files and format them for easier understanding. For example, the free program reads /proc/meminfo and converts the amounts given in bytes to kilobytes (and adds a little more information, as well).

/proc/uptime
: The time the system has been up.

/proc/version
: The kernel version.

/proc/video
: BTTV info of video resources.

1.15. /root

This is the home directory of the System Administrator, 'root'. This may be somewhat confusing ('root on root') but in former days, '/' was root's home directory (hence the name of the Administrator account). To keep things tidier, 'root' got his own home directory. Why not in '/home'? Because '/home' is often located on a different partition or even on another system and would thus be inaccessible to 'root' when – for some reason – only '/' is mounted.

The FSSTND merely states that this is the recommended location for the home directory of 'root'. It is left up to the end user to determine the home directory of 'root'. However, the FSSTND also says that:

```
If the home directory of the root account is not stored on the root
partition it will be necessary to make certain it will default to
/ if it can not be located.

We recommend against using the root account for tasks that can be
performed as an unprivileged user, and that it be used solely for
system administration. For this reason, we recommend that subdirectories
for mail and other applications not appear in the root account's home
directory, and that mail for administration roles such as root, postmaster,
and webmaster be forwarded to an appropriate user.
```

1.16. /sbin

Linux discriminates between 'normal' executables and those used for system maintenance and/or administrative tasks. The latter reside either here or − the less important ones − in /usr/sbin. Locally installed system administration programs should be placed into /usr/local/sbin.

Programs executed after /usr is known to be mounted (when there are no problems) are generally placed into /usr/sbin. This directory contains binaries that are essential to the working of the system. These include system administration as well as maintenance and hardware configuration programs. You may find lilo, fdisk, init, ifconfig, etc.... here.

Another directory that contains system binaries is /usr/sbin. This directory contains other binaries of use to the system administrator. This is where you will find the network daemons for your system along with other binaries that (generally) only the system administrator has access to, but which are not required for system maintenance and repair. Normally, these directories are never part of normal user's $PATHs, only of roots (PATH is an environment variable that controls the sequence of locations that the system will attempt to look in for commands).

The FSSTND states that:

```
/sbin should contain only binaries essential for booting, restoring,
recovering, and/or repairing the system in addition to the binaries
in /bin.
```

A particular eccentricity of the Linux filesystem hierarchy is that originally /sbin binaries were kept in /etc.

```
Deciding what things go into "sbin" directories is simple: if a normal
(not a system administrator) user will ever run it directly, then it
must be placed in one of the "bin" directories. Ordinary users should
not have to place any of the sbin directories in their path.

For example, files such as chfn which users only occasionally use must
still be placed in /usr/bin. ping, although it is absolutely necessary
for root (network recovery and diagnosis) is often used by users and
must live in /bin for that reason.

We recommend that users have read and execute permission for everything
in /sbin except, perhaps, certain setuid and setgid programs. The
division between /bin and /sbin was not created for security reasons or
to prevent users from seeing the operating system, but to provide a
good partition between binaries that everyone uses and ones that are
primarily used for administration tasks. There is no inherent security
advantage in making /sbin off-limits for users.
```

FSSTND compliance requires that the following commands, or symbolic links to commands, are required in /sbin.

```
     shutdown Command to bring the system down.
```

The following files, or symbolic links to files, must be in /sbin if the corresponding subsystem is installed:

```
     fastboot   Reboot the system without checking the disks (optional)
     fasthalt   Stop the system without checking the disks (optional)
     fdisk      Partition table manipulator (optional)
     fsck       File system check and repair utility (optional)
```

```
       fsck.*     File system check and repair utility for a specific filesystem (optional)
       getty      The getty program (optional)
       halt       Command to stop the system (optional)
       ifconfig   Configure a network interface (optional)
       init       Initial process (optional)
       mkfs       Command to build a filesystem (optional)
       mkfs.*     Command to build a specific filesystem (optional)
       mkswap     Command to set up a swap area (optional)
       reboot     Command to reboot the system (optional)
       route      IP routing table utility (optional)
       swapon     Enable paging and swapping (optional)
       swapoff    Disable paging and swapping (optional)
       update     Daemon to periodically flush filesystem buffers (optional)
```

1.17. /usr

/usr usually contains by far the largest share of data on a system. Hence, this is one of the most important directories in the system as it contains all the user binaries, their documentation, libraries, header files, etc.... X and its supporting libraries can be found here. User programs like telnet, ftp, etc.... are also placed here. In the original Unix implementations, /usr was where the home directories of the users were placed (that is to say, /usr/someone was then the directory now known as /home/someone). In current Unices, /usr is where user–land programs and data (as opposed to 'system land' programs and data) are. The name hasn't changed, but it's meaning has narrowed and lengthened from "everything user related" to "user usable programs and data". As such, some people may now refer to this directory as meaning 'User System Resources' and not 'user' as was originally intended.

```
/usr is shareable, read-only data. That means that /usr should
be shareable between various FHS-compliant hosts and must not be written to.
Any information that is host-specific or varies with time is stored elsewhere.

Large software packages must not use a direct subdirectory under the /usr
hierarchy.
```

/usr/X11R6

Another large subdirectory structure begins here, containing libraries, executables, docs, fonts and much more concerning the X Window System. Its inclusion here is somewhat inconsistent and so is the difference between '/usr' and '/usr/X11R6' directories. One would assume that programs that run on X only have their files in the '/usr/X11R6' hierarchy, while the others use '/usr'. Regrettably, it isn't so. KDE and GNOME put their files in the '/usr' hierarchy, whereas the window manager Window Maker uses '/usr/X11R6'. Documentation files for X11R6 are not in '/usr/X11R6/doc', but primarily in '/usr/X11R6/lib/X11/doc'. This mess is due to the fact that in contrast to other operating systems, the graphical desktop isn't an integral part of the system. Linux is still primarily used on servers, where graphical systems don't make sense.
This hierarchy is reserved for the X Window System, version 11 release 6, and related files. To simplify matters and make XFree86 more compatible with the X Window System on other systems, the following symbolic links must be present if /usr/X11R6 exists:

```
  /usr/bin/X11 -> /usr/X11R6/bin
  /usr/lib/X11 -> /usr/X11R6/lib/X11
  /usr/include/X11 -> /usr/X11R6/include/X11
```

In general, software must not be installed or managed via the above symbolic links. They are intended for utilization by users only. The difficulty is related to the release version of the X

Window System − in transitional periods, it is impossible to know what release of X11 is in use.

/usr/X11R6/bin
: XFree86 system binaries. These are necessary for the initialisation, configuration and running of the X windowing system. X, xf86config, xauth, xmodmap and even xpenguin are located here.

/usr/X11R6/include
: XFree86 system header files. There are required for the compilation of some applications that utilise the X toolkit.

/usr/X11R6/lib
: XFree86 system libraries.

/usr/X11R6/lib/modules
: XFree86 system modules. These are the modules that X loads upon startup. Without these modules video4linux, DRI and GLX extensions and drivers for certain input devices would cease to function.

/usr/X11R6/lib/X11/fonts
: XFree86 system fonts. Fonts that are utilised by 'xfs' (the X Font Server) and programs of that ilk.

/usr/bin
: This directory contains the vast majority of binaries on your system. Executables in this directory vary widely. For instance vi, gcc, gnome−session and mozilla and are all found here.

/usr/doc
: The central documentation directory. Documentation is actually located in /usr/share/doc and linked from here.

/usr/etc
: Theoretically, that's another directory for configuration files. Virtually unused now.

/usr/games
: Once upon a time, this directory contained network games files. Rarely used now.

/usr/include
: The directory for 'header files', needed for compiling user space source code.

/usr/include/'package−name'
: Application specific header files.

/usr/info
: This directory used to contain the files for the info documentation system. Now they are in '/usr/share/info'.

/usr/lib
: This directory contains program libraries. Libraries are collections of frequently used program routines.

/usr/local
: The original idea behind '/usr/local' was to have a separate ('local') '/usr' directory on every machine besides '/usr', which might be just mounted read−only from somewhere else. It copies the structure of '/usr'. These days, '/usr/local' is widely regarded as a good place in which to keep self−compiled or third−party programs. The /usr/local hierarchy is for use by the system administrator when installing software locally. It needs to be safe from being overwritten when the system software is updated. It may be used for programs and data that are shareable amongst a group of hosts, but not found in /usr. Locally installed software must be placed within /usr/local rather than /usr unless it is being installed to replace or upgrade software in /usr.

/usr/man
: It once held the man pages. It has been moved to /usr/share/man.

/usr/sbin
: This directory contains programs for administering a system, meant to be run by 'root'. Like '/sbin', it's not part of a user's $PATH. Examples of included binaries here are chroot, useradd, in.tftpd and pppconfig.

/usr/share

This directory contains 'shareable', architecture–independent files (docs, icons, fonts etc). Note, however, that '/usr/share' is generally not intended to be shared by different operating systems or by different releases of the same operating system. Any program or package which contains or requires data that doesn't need to be modified should store that data in '/usr/share' (or '/usr/local/share', if installed locally). It is recommended that a subdirectory be used in /usr/share for this purpose."

/usr/share/doc
: Location of package specific documentation files. These directories often contain useful information that might not be in the man pages. They may also contain templates and configuration files for certain utilities making configuration that much easier.

/usr/share/info
: Location of 'info' pages. This style of documentation seems to be largely ignored now. Manual pages are in far greater favour.

/usr/share/man
: Manual pages. They are organised into 8 sections, which are explained below.

```
man1: User programs
Manual pages that describe publicly accessible commands are contained
in this chapter. Most program documentation that a user will need to
use is located here.

man2: System calls
This section describes all of the system calls (requests for the kernel
to perform operations).

man3: Library functions and subroutines
Section 3 describes program library routines that are not direct calls
to kernel services. This and chapter 2 are only really of interest to
programmers.

man4: Special files
Section 4 describes the special files, related driver functions, and
networking support available in the system. Typically, this includes
the device files found in /dev and the kernel interface to networking
protocol support.

man5: File formats
The formats for many data files are documented in the section 5. This
includes various include files, program output files, and system files.

man6: Games
This chapter documents games, demos, and generally trivial programs.
Different people have various notions about how essential this is.

man7: Miscellaneous Manual pages that are difficult to classify are
designated as being section 7. The troff and other text processing
macro packages are found here.

man8: System administration Programs used by system administrators
for system operation and maintenance are documented here. Some of
these programs are also occasionally useful for normal users.
```

/usr/src
: The 'linux' sub–directory holds the Linux kernel sources, header–files and documentation.

/usr/src/RPM
: RPM provides a substructure for building RPMs from SRPMs. Organisation of this branch is fairly logical with packages being organised according to a package's architecture.

/usr/src/RPM/BUILD
: A temporary store for RPM binary files that are being built from source code.

/usr/src/RPM/RPMS/athlon, /usr/src/RPM/RPMS/i386, /usr/src/RPM/RPMS/i486, /usr/src/RPM/RPMS/i586, /usr/src/RPM/RPMS/i686, /usr/src/RPM/RPMS/noarch
: These directories contain architecture dependant RPM source files.

/usr/src/RPM/SOURCES
: This directory contains the source TAR files, patches, and icon files for software to be packaged.

/usr/src/RPM/SPECS
: RPM SPEC files. A SPEC file is a file that contains information as well as scripts that are necessary to build a package.

/usr/src/RPM/SRPMS
: Contains the source RPM files which result from a build.

/usr/src/linux
: Contains the source code for the Linux kernel.

/usr/src/linux/.config
: The last kernel source configuration. This file is normally created through the 'make config', 'make menuconfig' or 'make xconfig' steps during kernel compilation.

/usr/src/linux/.depend, /usr/src/linux/.hdepend
: 'make dep' checks the dependencies of the selections you made when you created your .config file. It ensures that the required files can be found and it builds a list that is to be used during compilation. Should this process be successful these two files are created.

/usr/src/linux/COPYING
: GNU License

/usr/src/linux/CREDITS
: A partial credits−file of people that have contributed to the Linux project. It is sorted by name and formatted to allow easy grepping and beautification by scripts. The fields are: name (N), email (E), web−address (W), PGP key ID and fingerprint (P), description (D), and snail−mail address (S).

/usr/src/linux/MAINTAINERS
: List of maintainers and details on how to submit kernel changes.

/usr/src/linux/Makefile
: Contains data necessary for compilation of a working kernel. It allows developers and end−users to compile a kernel with a few simple steps (ie. make dep, make clean, make bzImage, make modules, make modules_install) and also not have to worry about re−compiling everything from scratch if parts of it have already been done so and are up to date.

/usr/src/linux/README
: These are the release notes for Linux version 2.4. Read them carefully, as they tell you what this is all about, explain how to install the kernel, and what to do if something goes wrong.

/usr/src/linux/REPORTING−BUGS
: A suggested procedure for reporting Linux bugs. You aren't obliged to use the bug reporting format, it is provided as a guide to the kind of information that can be useful to developers − no more.

/usr/src/linux/Rules.make
: This file contains rules which are shared between multiple Makefiles.

/usr/src/linux/Documentation
: Contains documentation that may be necessary in order to re−compile a kernel. However, it also provides quite a lot of information about your Linux system in general as well. For those who wish to seek further information on the contents of this directory you may consult the /usr/src/linux/Documentation/00−INDEX file. Further, more detailed documentation may be found in /usr/src/linux/Documentation/Docbook. Of course, the contents of this directory is written in Docbook but may be converted to pdf, ps or html using the make targets of 'pdfdocs', 'psdocs' and 'htmldocs' respectively.

/usr/tmp
: User space temporary files. This directory is not found on modern distributions at all and was most likely created as a consequence of Linux's UNIX heritage.

1.18. /var

Contains variable data like system logging files, mail and printer spool directories, and transient and temporary files. Some portions of /var are not shareable between different systems. For instance, /var/log, /var/lock, and /var/run. Other portions may be shared, notably /var/mail, /var/cache/man, /var/cache/fonts, and /var/spool/news. Why not put it into /usr? Because there might be circumstances when you may want to mount /usr as read−only, e.g. if it is on a CD or on another computer. '/var' contains variable data, i.e. files and directories the system must be able to write to during operation, whereas /usr should only contain static data. Some directories can be put onto separate partitions or systems, e.g. for easier backups, due to network topology or security concerns. Other directories have to be on the root partition, because they are vital for the boot process. 'Mountable' directories are: '/home', '/mnt', '/tmp', '/usr' and '/var'. Essential for booting are: '/bin', '/boot', '/dev', '/etc', '/lib', '/proc' and '/sbin'.

```
If /var cannot be made a separate partition, it is often preferable to move /
var out of the root partition and into the /usr partition. (This is sometimes
done to reduce the size of the root partition or when space runs low in the
root partition.) However, /var must not be linked to /usr because this makes
separation of /usr and /var more difficult and is likely to create a naming
conflict. Instead, link /var to /usr/var.

Applications must generally not add directories to the top level of /var. Such
directories should only be added if they have some system−wide implication, and
in consultation with the FHS mailing list.
```

/var/backups
: Directory containing backups of various key system files such as /etc/shadow, /etc/group, /etc/inetd.conf and dpkg.status. They are normally renamed to something like dpkg.status.0, group.bak, gshadow.bak, inetd.conf.bak, passwd.bak, shadow.bak

/var/cache
: Is intended for cached data from applications. Such data is locally generated as a result of time−consuming I/O or calculation. This data can generally be regenerated or be restored. Unlike /var/spool, files here can be deleted without data loss. This data remains valid between invocations of the application and rebooting of the system. The existence of a separate directory for cached data allows system administrators to set different disk and backup policies from other directories in /var.

/var/cache/fonts
: Locally−generated fonts. In particular, all of the fonts which are automatically generated by mktexpk must be located in appropriately−named subdirectories of /var/cache/ fonts.

/var/cache/man
: A cache for man pages that are formatted on demand. The source for manual pages is usually stored in /usr/share/man/; some manual pages might come with a pre−formatted version, which is stored in /usr/share/man/cat* (this is fairly rare now). Other manual pages need to be formatted when they are first viewed; the formatted version is then stored in /var/man so that the next person to view the same page won't have to wait for it to be formatted (/var/catman is often cleaned in the same way temporary directories are cleaned).

/var/cache/'package−name'
: Package specific cache data.

/var/cache/www
: WWW proxy or cache data.

/var/crash

This directory will eventually hold system crash dumps. Currently, system crash dumps are not supported under Linux. However, development is already complete and is awaiting unification into the Linux kernel.

/var/db

Data bank store.

/var/games

Any variable data relating to games in /usr is placed here. It holds variable data that was previously found in /usr. Static data, such as help text, level descriptions, and so on, remains elsewhere though, such as in /usr/share/games. The separation of /var/games and /var/lib as in release FSSTND 1.2 allows local control of backup strategies, permissions, and disk usage, as well as allowing inter−host sharing and reducing clutter in /var/lib. Additionally, /var/games is the path traditionally used by BSD.

/var/lib

Holds dynamic data libraries/files like the rpm/dpkg database and game scores. Furthermore, this hierarchy holds state information pertaining to an application or the system. State information is data that programs modify while they run, and that pertains to one specific host. Users shouldn't ever need to modify files in /var/lib to configure a package's operation. State information is generally used to preserve the condition of an application (or a group of inter−related applications) between invocations and between different instances of the same application. An application (or a group of inter−related applications) use a subdirectory of /var/lib for their data. There is one subdirectory, /var/lib/misc, which is intended for state files that don't need a subdirectory; the other subdirectories should only be present if the application in question is included in the distribution. /var/lib/'name' is the location that must be used for all distribution packaging support. Different distributions may use different names, of course.

/var/local

Variable data for local programs (i.e., programs that have been installed by the system administrator) that are installed in /usr/local (as opposed to a remotely mounted '/var' partition). Note that even locally installed programs should use the other /var directories if they are appropriate, e.g., /var/lock.

/var/lock

Many programs follow a convention to create a lock file in /var/lock to indicate that they are using a particular device or file. This directory holds those lock files (for some devices) and hopefully other programs will notice the lock file and won't attempt to use the device or file.

Lock files should be stored within the /var/lock directory structure. Lock files for devices and other resources shared by multiple applications, such as the serial device lock files that were originally found in either /usr/spool/locks or /usr/spool/uucp, must now be stored in /var/lock. The naming convention which must be used is LCK.. followed by the base name of the device file. For example, to lock /dev/ttyS0 the file LCK..ttyS0 would be created. The format used for the contents of such lock files must be the HDB UUCP lock file format. The HDB format is to store the process identifier (PID) as a ten byte ASCII decimal number, with a trailing newline. For example, if process 1230 holds a lock file, it would contain the eleven characters: space, space, space, space, space, space, one, two, three, zero, and newline.

/var/log

Log files from the system and various programs/services, especially login (/var/log/wtmp, which logs all logins and logouts into the system) and syslog (/var/log/messages, where all kernel and system program message are usually stored). Files in /var/log can often grow indefinitely, and may require cleaning at regular intervals. Something that is now normally managed via log rotation utilities such as 'logrotate'. This utility also allows for the automatic rotation compression, removal and mailing of log files. Logrotate can be set to handle a log file daily, weekly, monthly or when the log file gets to a certain size. Normally, logrotate runs as a daily cron job. This is a good place to start troubleshooting general technical problems.

/var/log/auth.log
: Record of all logins and logouts by normal users and system processes.

/var/log/btmp
: Log of all attempted bad logins to the system. Accessed via the lastb command.

/var/log/debug
: Debugging output from various packages.

/var/log/dmesg
: Kernel ring buffer. The content of this file is referred to by the dmesg command.

/var/log/gdm/
: GDM log files. Normally a subset of the last X log file. See /var/log/xdm.log for mode details.

/var/log/kdm.log
: KDM log file. Normally a subset of the last X log file. See /var/log/xdm.log for more details.

/var/log/messages
: System logs.

/var/log/pacct
: Process accounting is the bookkeeping of process activity. The raw data of process activity is maintained here. Three commands can be used to access the contents of this file dump−acct, sa (summary of process accounting) and lastcomm (list the commands executed on the system).

/var/log/utmp
: Active user sessions. This is a data file and as such it can not be viewed normally. A human−readable form can be created via the dump−utmp command or through the w, who or users commands.

/var/log/wtmp
: Log of all users who have logged into and out of the system. The last command can be used to access a human readable form of this file. It also lists every connection and run−level change.

/var/log/xdm.log
: XDM log file. Normally subset of the last X startup log and pretty much useless in light of the details the X logs is able to provide us with. Only consult this file if you have XDM specific issues otherwise just use the X logfile.

/var/log/XFree86.0.log, /var/log/XFree86.?.log
: X startup logfile. An excellent resource for uncovering problems with X configuration. Log files are numbered according to when they were last used. For example, the last log file would be stored in /var/log/XFree86.0.log, the next /var/log/XFree86.9.log, so on and so forth.

/var/log/syslog
: The 'system' log file. The contents of this file is managed via the syslogd daemon which more often than not takes care of all log manipulation on most systems.

/var/mail
: Contains user mailbox files. The mail files take the form /var/mail/'username' (Note that /var/mail may be a symbolic link to another directory). User mailbox files in this location are stored in the standard UNIX mailbox format. The reason for the location of this directory was to bring the FHS inline with nearly every UNIX implementation (it was previously located in /var/spool/mail). This change is important for inter−operability since a single /var/mail is often shared between multiple hosts and multiple UNIX implementations (despite NFS locking issues).

/var/opt
: Variable data of the program packages in /opt must be installed in /var/opt/'package−name', where 'package−name' is the name of the subtree in /opt where the static data from an add−on software package is stored, except where superceded by another file in /etc. No structure is imposed on the internal arrangement of /var/opt/'package−name'.

/var/run
: Contains the process identification files (PIDs) of system services and other information about the system that is valid until the system is next booted. For example, /var/run/utmp contains information about users currently logged in.

/var/spool
: Holds spool files, for instance for mail, news, and printing (lpd) and other queued work. Spool files store data to be processed after the job currently occupying a device is finished or the appropriate cron job is started. Each different spool has its own subdirectory below /var/spool, e.g., the cron tables are stored in /var/spool/cron/crontabs.

/var/tmp
: Temporary files that are large or that need to exist for a longer time than what is allowed for /tmp. (Although the system administrator might not allow very old files in /var/tmp either.)

/var/named
: Database for BIND. The Berkeley Internet Name Domain (BIND) implements an Internet domain name server. BIND is the most widely used name server software on the Internet, and is supported by the Internet Software Consortium, www.isc.org.

/var/yp
: Database for NIS (Network Information Services). NIS is mostly used to let several machines in a network share the same account information (eg. /etc/passwd). NIS was formerly called Yellow Pages (YP).

The following directories, or symbolic links to directories, are required in /var for FSSTND compliance:

```
/var/cache    Application cache data
/var/lib      Variable state information
/var/local    Variable data for /usr/local
/var/lock     Lock files
/var/log      Log files and directories
/var/opt      Variable data for /opt
/var/run      Data relevant to running processes
/var/spool    Application spool data
/var/tmp      Temporary files preserved between system reboots
```

Several directories are 'reserved' in the sense that they must not be used arbitrarily by some new application, since they would conflict with historical and/or local practice. They are:

```
/var/backups
/var/cron
/var/msgs
/var/preserve
```

The following directories, or symbolic links to directories, must be in /var, if the corresponding subsystem is installed:

```
account   Process accounting logs (optional)
crash     System crash dumps (optional)
games     Variable game data (optional)
mail      User mailbox files (optional)
yp        Network Information Service (NIS) database files (optional)
```

1.19. /srv

```
/srv contains site-specific data which is served by this system.

This main purpose of specifying this is so that users may find
the location of the data files for particular service, and so that
services which require a single tree for readonly data, writable data
```

```
and scripts (such as cgi scripts) can be reasonably placed. Data that
is only of interest to a specific user should go in that users'
home directory.

The methodology used to name subdirectories of /srv is unspecified as there
is currently no consensus on how this should be done. One method for
structuring data under /srv is by protocol, eg. ftp, rsync, www, and cvs.
On large systems it can be useful to structure /srv by administrative
context, such as /srv/physics/www, /srv/compsci/cvs, etc. This setup will
differ from host to host. Therefore, no program should rely on a specific
subdirectory structure of /srv existing or data necessarily being stored in
/srv. However /srv should always exist on FHS compliant systems and should
be used as the default location for such data.

Distributions must take care not to remove locally placed files in these
directories without administrator permission.

This is particularly important as these areas will often contain both
files initially installed by the distributor, and those added by the
administrator.
```

1.20. /tmp

This directory contains mostly files that are required temporarily. Many programs use this to create lock files and for temporary storage of data. Do not remove files from this directory unless you know exactly what you are doing! Many of these files are important for currently running programs and deleting them may result in a system crash. Usually it won't contain more than a few KB anyway. On most systems, this directory is cleared out at boot or at shutdown by the local system. The basis for this was historical precedent and common practice. However, it was not made a requirement because system administration is not within the scope of the FSSTND. For this reason people and programs must not assume that any files or directories in /tmp are preserved between invocations of the program. The reasoning behind this is for compliance with IEEE standard P1003.2 (POSIX, part 2).

Glossary

ARPA
: The Advanced Research and Projects Agency of the United States Department of Defense. Also known as DARPA (the "D" stands for "Defense"), it originated in the late 1960s and early 1970s the proposal and standards for the Internet. For this reason, the Internet was initially referred to as ARPANet, and connected the military with the various centers of research around the United States in a way that was intended to have a high degree of survivability against a nuclear attack.

BASH
: The Bourne Again Shell and is based on the Bourne shell, sh, the original command interpreter.

Bourne Shell
: The Bourne shell is the original Unix shell (command execution program, often called a command interpreter) that was developed at AT&T. Named for its developer, Stephen Bourne, the Bourne shell is also known by its program name, sh. The shell prompt (character displayed to indicate readiness for input) used is the $ symbol. The Bourne shell family includes the Bourne, Korn shell, bash, and zsh shells. Bourne Again Shell (bash) is the free version of the Bourne shell distributed with Linux systems. Bash is similar to the original, but has added features such as command line editing. Its name is sometimes spelled as Bourne Again SHell, the capitalized Hell referring to the difficulty some people have with it.

CLI
: A CLI (command line interface) is a user interface to a computer's operating system or an application in which the user responds to a visual prompt by typing in a command on a specified line, receives a response back from the system, and then enters another command, and so forth. The MS−DOS Prompt application in a Windows operating system is an example of the provision of a command line interface. Today, most users prefer the graphical user interface (GUI) offered by Windows, Mac OS, BeOS, and others. Typically, most of today's Unix−based systems offer both a command line interface and a graphical user interface.

core
: A core file is created when a program terminates unexpectedly, due to a bug, or a violation of the operating system's or hardware's protection mechanisms. The operating system kills the program and creates a core file that programmers can use to figure out what went wrong. It contains a detailed description of the state that the program was in when it died. If would like to determine what program a core file came from, use the file command, like this: $ file core That will tell you the name of the program that produced the core dump. You may want to write the maintainer(s) of the program, telling them that their program dumped core. To Enable or Disable Core Dumps you must use the ulimit command in bash, the limit command in tcsh, or the rlimit command in ksh. See the appropriate manual page for details. This setting affects all programs run from the shell (directly or indirectly), not the whole system. If you wish to enable or disable core dumping for all processes by default, you can change the default setting in /usr/include/linux/sched.h. Refer to definition of INIT_TASK, and look also in /usr/include/linux/resource.h. PAM support optimizes the system's environment, including the amount of memory a user is allowed. In some distributions this parameter is configurable in the /etc/security/limits.conf file. For more information, refer to the Linux Administrator's Security Guide.

daemon
: A process lurking in the background, usually unnoticed, until something triggers it into action. For example, the \cmd{update} daemon wakes up every thirty seconds or so to flush the buffer cache, and the \cmd{sendmail} daemon awakes whenever someone sends mail.

DARPA
: The Defense Advanced Research Projects Agency is the central research and development organization for the Department of Defense (DoD). It manages and directs selected basic and applied

research and development projects for DoD, and pursues research and technology where risk and payoff are both very high and where success may provide dramatic advances for traditional military roles and missions.

DHCP
: Dynamic Host Control Protocol, is a protocol like BOOTP (actually dhcpd includes much of the functionality of BOOTPD). It assigns IP addresses to clients based on lease times. DHCP is used extensively by Microsoft and more recently also by Apple. It is probably essential in any multi−platform environment.

DNS
: Domain Name System translates Internet domain and host names to IP addresses. DNS implements a distributed database to store name and address information for all public hosts on the Net. DNS assumes IP addresses do not change (i.e., are statically assigned rather than dynamically assigned). The DNS database resides on a hierarchy of special−purpose servers. When visiting a Web site or other device on the Net, a piece of software called the DNS resolver (usually built into the network operating system) first contacts a DNS server to determine the server's IP address. If the DNS server does not contain the needed mapping, it will in turn forward the request to a DNS server at the next higher level in the hierarchy. After potentially several forwarding and delegation messages are sent within the DNS hierarchy, the IP address for the given host eventually is delivered to the resolver. DNS also includes support for caching requests and for redundancy. Most network operating systems allow one to enter the IP addresses of primary, secondary, and tertiary DNS servers, each of which can service initial requests from clients. Many ISPs maintain their own DNS servers and use DHCP to automatically assign the addresses of these servers to dial−in clients, so most home users need not be aware of the details behind DNS configuration. Registered domain names and addresses must be renewed periodically, and should a dispute occur between two parties over ownership of a given name, such as in trademarking, ICANN's Uniform Domain−Name Dispute−Resolution Policy can be invoked. Also known as Domain Name System, Domain Name Service, Domain Name Server.

environment variable
: A variable that is available to any program that is started by the shell.

ESD
: Enlightened Sound Daemon. This program is designed to mix together several digitized audio streams for playback by a single device.

filesystem
: The methods and data structures that an operating system uses to keep track of files on a disk or partition; the way the files are organized on the disk. Also used to describe a partition or disk that is used to store the files or the type of the filesystem.

FSSTND
: Often the group, which creates the Linux File System Structure document, or the document itself, is referred to as the 'FSSTND'. This is short for "file system standard". This document has helped to standardize the layout of file systems on Linux systems everywhere. Since the original release of the standard, most distributors have adopted it in whole or in part, much to the benefit of all Linux users. It is now often refered to as the FHS (Filesystem Hierarchy Standard) document though since its incorporation into the LSB (Linux Standards Base) Project.

GUI
: Graphical User Interface. The use of pictures rather than just words to represent the input and output of a program. A program with a GUI runs under some windowing system (e.g. The X Window System, Microsoft Windows, Acorn RISC OS, NEXTSTEP). The program displays certain icons, buttons, dialogue boxes etc. in its windows on the screen and the user controls it mainly by moving a pointer on the screen (typically controlled by a mouse) and selecting certain objects by pressing buttons on the mouse while the pointer is pointing at them. Though Apple Computer would like to claim they invented the GUI with their Macintosh operating system, the concept originated in the early 1970s at Xerox's PARC laboratory.

hard link
: A directory entry, which maps a filename to an inode, number. A file may have multiple names or hard links. The link count gives the number of names by which a file is accessible. Hard links do not allow multiple names for directories and do not allow multiple names in different filesystems.

init
: 'init' process is the first user level process started by the kernel. init has many important duties, such as starting getty (so that users can log in), implementing run levels, and taking care of orphaned processes. This chapter explains how init is configured and how you can make use of the different run levels. init is one of those programs that are absolutely essential to the operation of a Linux system, but that you still can mostly ignore. Usually, you only need to worry about init if you hook up serial terminals, dial−in (not dial−out) modems, or if you want to change the default run level. When the kernel has started (has been loaded into memory, has started running, and has initialized all device drivers and data structures and such), it finishes its own part of the boot process by starting a user level program, init. Thus, init is always the first process (its process number is always 1). The kernel looks for init in a few locations that have been historically used for it, but the proper location for it is /sbin/init. If the kernel can't find init, it tries to run /bin/sh, and if that also fails, the startup of the system fails. When init starts, it completes the boot process by doing a number of administrative tasks, such as checking filesystems, cleaning up /tmp, starting various services, and starting a getty for each terminal and virtual console where users should be able to log in. After the system is properly up, init restarts getty for each terminal after a user has logged out (so that the next user can log in). init also adopts orphan processes: when a process starts a child process and dies before its child, the child immediately becomes a child of init. This is important for various technical reasons, but it is good to know it, since it makes it easier to understand process lists and process tree graphs. init itself is not allowed to die. You can't kill init even with SIGKILL. There are a few variants of init available. Most Linux distributions use sysvinit (written by Miquel van Smoorenburg), which is based on the System V init design. The BSD versions of Unix have a different init. The primary difference is run levels: System V has them, BSD doesn't.

inode
: An inode is the address of a disk block. When you see the inode information through ls, ls prints the address of the first block in the file. You can use this information to tell if two files are really the same file with different names (links). A file has several components: a name, contents, and administrative information such as permissions and modification times. The administrative information is stored in the inode (over the years, the hyphen fell out of "i−node"), along with essential system data such as how long it is, where on the disc the contents of the file are stored, and so on. There are three times in the inode: the time that the contents of the file were last modified (written); the time that the file was last used (read or executed); and the time that the inode itself was last changed, for example to set the permissions. Altering the contents of the file does not affect its usage time and changing the permissions affects only the inode change time. It is important to understand inodes, not only to appreciate the options on ls, but because in a strong sense the inodes are the files. All the directory hierarchy does is provide convenient names for files. The system's internal name for the file is its i−number: the number of the inode holding the file's information.

kernel
: Part of an operating system that implements the interaction with hardware and the sharing of resources.

libraries
: Executables should have no undefined symbols, only useful symbols; all useful programs refer to symbols they do not define (eg. printf or write). These references are resolved by pulling object files from libraries into the executable.

link
: A symbolic link (alias in MacOS and shortcut under Windows) is a file that points to another file; this is a commonly used tool. A hard−link rarely created by the user, is a filename that points to a block of

data that has several other filenames as well.

man page
: Every version of UNIX comes with an extensive collection of online help pages called man pages (short for manual pages). The man pages are the authoritative documentation about your UNIX system. They contain complete information about both the kernel and all the utilities.

MTA
: Mail Transfer Agents. Alongside the web, mail is the top reason for the popularity of the Internet. E−mail is an inexpensive and fast method of time−shifted messaging which, much like the Web, is actually based around sending and receiving plain text files. The protocol used is called the Simple Mail Transfer Protocol (SMTP). The server programs that implement SMTP to move mail from one server to another are called MTAs. Once upon a time users would have to Telnet into an SMTP server and use a command line mail program like 'mutt' or 'pine' to check their mail. Now, GUI based e−mail clients like Mozilla, Kmail and Outlook allow users to check their email off of a local SMTP sever. Additional protocols like POP3 and IMAP4 are used between the SMTP server and desktop mail client to allow clients to manipulate files on, and download from, their local mail server. The programs that implement POP3 and IMAP4 are called Mail Delivery Agents (MDAs). They are generally separate from MTAs.

NFS
: Network File System, is the UNIX equivalent of Server Message Block (SMB). It is a way through which different machines can import and export local files between each other. Like SMB though, NFS sends information including user passwords unencrypted, so it's best to limit its usage to within your local network.

operating system
: Software that shares a computer system's resources (processor, memory, disk space, network bandwidth, and so on) between users and the application programs they run. Controls access to the system to provide security.

PAM
: Pluggable Authentication Modules. A suite of shared libraries that determine how a user will be authenticated. For example, conventionally UNIX users authenticate themselves by supplying a password at the password prompt after they have typed their name at the login prompt. In many circumstances, such as internal access to workstations, this simple form of authentication is considered sufficient. In other cases, more information is warranted. If a user wants to log in to an internal system from an external source, like the Internet, more or alternative information may be required − perhaps a one−time password. PAM provides this type of capability and much more. Most important, PAM modules allow you to configure your environment with the necessary level of security.

PATH
: The shell looks for commands and programs in a list of file paths stored in the PATH environment variable. An environment variable stores information in a place where other programs and commands can access it. Environment variables store information such as the shell that you are using, your login name, and your current working directory. To see a list of all the environment variables currently defined; type 'set' at the prompt. When you type a command at the shell prompt, the shell will look for that command's program file in each directory listed in the PATH variable, in order. The first program found matching the command you typed will be run. If the command's program file is not in a directory listed in you PATH environment variable, the shell returns a "commands not found" error. By default, the shell does not look in your current working directory or your home directory for commands This is really a security mechanism so that you don't execute programs by accident. What if a malicious user put a harmful program called ls in your home directory? If you typed ls and the shell looked for the fake program in your home directory before the real program in the /bin directory, what do you think would happen? If you thought bad things, you are on the right track. Since your PATH doesn't have the current directory as one of its search locations, programs in your current

directory must be called with an absolute path of a relative path specified as './program–name'. To see what directories are part of your PATH enter this command: # echo $PATH /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11

pipes and sockets
: Special files that programs use to communicate with one another. They are rarely seen, but you might be able to see a socket or two in the /dev/ directory.

process identifier
: Shown in the heading of the ps command as PID. The unique number assigned to every process running in the system.

rpc
: Remote Procedure Calls. It enables a system to make calls to programs such as NFS across the network transparently, enabling each system to interpret the calls as if they were local. In this case, it would make exported filesystems appear as thought they were local.

set group ID (SGID)
: The SGID permission causes a script to run with its group set to the group of the script, rather than the group of the user who started it. It is normally considered extremely bad practice to run a program in this way as it can pose many security problems. Later versions of the Linux kernel will even prohibit the running of shell scripts that have this attribute set.

set user ID (SUID)
: The SUID permission causes a script to run as the user who is the owner of the script, rather than the user who started it. It is normally considered extremely bad practice to run a program in this way as it can pose many security problems. Later versions of the Linux kernel will even prohibit the running of shell scripts that have this attribute set.

signal
: Software interrupts sent to a program to indicate that an important event has occurred. The events can vary from user requests to illegal memory access errors. Some signals, like the interrupt signal, indicate that a user has asked the program to do something that is not in the usual flow of control.

SSH
: The Secure Shell, or SSH, provides a way of running command line and graphical applications, and transferring files, over an encrypted connection, all that will be seen is junk. It is both a protocol and a suite of small command line applications, which can be used for various functions. SSH replaces the old Telnet application, and can be used for secure remote administration of machines across the Internet. However, it also has other features. SSH increases the ease of running applications remotely by setting up X permissions automatically. If you can log into a machine, it allows you to run a graphical application on it, unlike Telnet, which requires users to have an understanding of the X authentication mechanisms that are manipulated through the xauth and xhost commands. SSH also has inbuilt compression, which allows your graphic applications to run much faster over the network. SCP (Secure Copy) and SFTP (Secure FTP) allow transfer of files over the remote link, either via SSH's own command line utilities or graphical tools like Gnome's GFTP. Like Telnet, SSH is cross–platform. You can find SSH server and clients for Linux, Unix and all flavours of Windows, BeOS, PalmOS, Java and embedded Oses used in routers.

STDERR
: Standard error. A special type of output used for error messages. The file descriptor for STDERR is 2.

STDIN
: Standard input. User input is read from STDIN. The file descriptor for STDIN is 0.

STDOUT
: Standard output. The output of scripts is usually to STDOUT. The file descriptor for STDOUT is 1.

symbol table
: The part of an object table that gives the value of each symbol (usually as a section name and an offset) is called the symbol table. Executables may also have a symbol table, with this one giving the final values of the symbols. Debuggers use the symbol table to present addresses to the user in a

symbolic, rather than a numeric form. It is possible to strip the symbol table from executables resulting in a smaller sized executable but this prevents meaningful debugging.

symbolic link or soft link

A special filetype, which is a small pointer file, allowing multiple names for the same file. Unlike hard links, symbolic links can be made for directories and can be made across filesystems. Commands that access the file being pointed to are said to follow the symbolic link. Commands that access the link itself do not follow the symbolic link.

system call

The services provided by the kernel to application programs, and the way in which they are invoked. See section 2 of the manual pages.

system program

Programs that implement high level functionality of an operating system, i.e., things that aren't directly dependent on the hardware. May sometimes require special privileges to run (e.g., for delivering electronic mail), but often just commonly thought of as part of the system (e.g., a compiler).

tcp−wrappers

Almost all of the services provided through inetd are invoked through tcp−wrappers by way of the tcp−wrappers daemon, tcpd. The tcp−wrappers mechanism provides access control list restrictions and logging for all service requests to the service it wraps. It may be used for either TCP or TCP services as long as the services are invoked through a central daemon process such as inetd. These programs log the client host name of incoming telnet, ftp, rsh, rlogin, finger etc.... requests. Security options are access control per host, domain and/or service; detection of host name spoofing or host address spoofing; booby traps to implement an early−warning system.

ZSH

Zsh was developed by Paul Falstad as a replacement for both the Bourne and C shell. It incorporates features of all the other shells (such as file name completion and a history mechanism) as well as new capabilities. Zsh is considered similar to the Korn shell. Falstad intended to create in zsh a shell that would do whatever a programmer might reasonably hope it would do. Zsh is popular with advanced users. Along with the Korn shell and the C shell, the Bourne shell remains among the three most widely used and is included with all UNIX systems. The Bourne shell is often considered the best shell for developing scripts.

Appendix A. UNIX System V Signals

Symbol	*Number*	*Action*	*Meaning*
SIGHUP	1	exit	Hangs up.
SIGINT	2	exit	Interrupts.
SIGQUIT	3	core dump	Quits.
SIGILL	4	core dump	Illegal instruction.
SIGTRAP	5	core dump	Trace trap.
SIGIOT	6	core dump	IOT instruction.
SIGEMT	7	core dump	MT instruction.
SIGFPE	8	core dump	Floating point exception.
SIGKILL	9	exit	Kills (cannot be caught or ignored).
SIGBUS	10	core dump	Bus error.
SIGSEGV	11	core dump	Segmentation violation.
SIGSYS	12	core dump	Bad argument to system call.
SIGPIPE	13	exit	Writes on a pipe with no one to read it.
SIGALRM	14	exit	Alarm clock.
SIGTERM	15	exit	Software termination signal.

Appendix B. Sources

- The UNIX programming environment, Brian W. Kernighan, Rob Pike, Prentice Hall, New Jersey, 1984.
- Newnes UNIX Pocket Book, Steve Heath, Butterworth−Heinemann, Great Britain, 1998.
- Suse Linux Installation and Configuration, Nazeeh Amin El−Dirghami & Youssef A. Abu Kwaik, QUE Corporation, USA, 2000.
- Inside Linux, Michael J. Tobler, New Riders Publishing, USA, 2001.
- Linux in a Nutshell 2nd Edition, Ellen Siever, O'Reilly & Associates Inc., CA, USA, 1999
- Using Caldera OpenLinux Special Edition, Allan Smart, Erik Ratcliffe, Tim Bird, David Bandel, QUE Corporation, USA, 1999.
- Linux System Security (The Administrator's Guide to Open Source Security Tools), Scott Mann & Ellen L. Mitchell, Prentice−Hall, New−Jersey, 2000.
- XFree86 For Linux (Uncommon Solutions for the Technical Professional), Aron Hsiao, QUE Corporation, USA, 1999.
- Complete Idiot's Guide to Linux Second Edition, Manuel Alberto Ricart, QUE Corporation, USA, 1999.
- Lions' Commentary on UNIX 6th Edition with Source Code, John Lions, Peer−to−Peer Communications Incorporated, USA, 1996.
- The Linux System Administrators' Guide Version 0.6.1, Lars Wirzenius, liw@iki.fi, Finland, 1998.
- SAMS Teach Yourself Shell Programming in 24 Hours, Sriranga Veeraraghavan, SAMS Publishing, USA, 1999.
- 433−252 Software Development: Principles and Tools, Zoltan Somogyi, Les Kitchen, The University of Melbourne, Department of Computer Science and Software Engineering, Australia, 2002.
- The Advanced Linux Pocketbook, Ashton Mills, ashtonmills@bigpond.com, ACP Publishing Pty Ltd, Australia, 2001.
- http://www.pathname.com/fhs
- http://www.atnf.csiro.au/people/rgooch/linux/docs/devfs.html
- http://freeos.com/articles/3102/
- http://freeos.com/articles/2879/
- http://www.linuxjournal.com/article.php?sid=1104

- http://www.mlinux.org/projects/present/filesys/slide01.html
- http://www.mil.ufl.edu/linuxhelp/linuxfilesystem.pdf
- http://www.nsa.gov/selinux/doc/slinux/node57.html
- http://www.linuxnow.com/tutorial/fs/fs.html
- http://info.cqu.edu.au/courses/aut2001/85321/resources/study_guide/chapter_4
- http://lwn.net/2001/features/ols/pdf/pdf/devfs.pdf
- /usr/src/linux/Documentation/filesystems/proc.txt, Terrehon Bowden <terrehon@pacbell.net>, Bodo Bauer <bb@ricochet.net>, Jorge Nerin <comandante@zaralinux.com>
- /usr/share/doc/FAQ/Linux−FAQ/index.html, ftp://rtfm.mit.edu/pub/usenet−by−hierarchy/comp/os/linux/misc/, David Merrill, david −AT− lupercalia.net, Robert Kiesling, rkiesling@mainmatter.com, Linux Frequently Asked Questions with Answers, 2001−12−04.
- /usr/share/doc/sysvinit/README.runlevels.gz
- /usr/src/linux/Documentation/initrd.txt, Werner Almesberger <werner.almesberger@epfl.ch> and Hans Lermen <lermen@fgan.de>, 2000.
- http://www.opussoftware.com/tutorial/TutMakefile.htm
- http://www.cc.gatech.edu/people/home/oxblood/Linux_Doc.txt
- http://users.erols.com/mmmcd/lilo/boot_sequence.html
- hints.linuxfromscratch.org/hints/grub−howto.txt
- http://www.fifi.org/cgi−bin/man2html?initrd+4
- http://www.linuxsa.org.au/meetings/1997−06/fsstnd/fsstnd.html
- http://public.csusm.edu/public/guests/history/netinfo/arpa.html
- http://whatis.techtarget.com/definition/0,,sid9_gci214635,00.html
- http://whatis.techtarget.com/definition/0,,sid9_gci213627,00.html
- http://www.darpa.mil/
- http://hostingworks.com/support/dict.phtml?foldoc=Defense%2BAdvanced%2BResearch%2BProjects%2BAgenc
- http://www.linuxsa.org.au/meetings/1997−06/fsstnd/fsstnd.html
- http://compnetworking.about.com/library/glossary/bldef−dns.htm

- http://www.scala.com/definition/graphical−user−interface.html
- http://hostingworks.com/support/dict.phtml?foldoc=Graphical+User+Interface
- http://www.mostang.com/sane/intro.html

Appendix C. About the Author

Binh Nguyen was born, 26 March 1983, Melbourne, Victoria. He studied at St Josephs Marist Brothers College, North Fitzroy until receiving a scholarship to St Kevin's College, Toorak in 1998.

He is currently a university undergraduate studying computer science and physics at the University of Melbourne. His main interests in each area are in Operating Systems and Quantum Mechanics respectively.

His background is strongly biased towards science and mathematics. Nonetheless, he does have an appreciation for the arts, humanities and sport. A reasonably proficient musician (flute), he is currently pondering whether he should complete his musical studies to obtain a diploma so that he can teach. In high school, he was a member of the athletics, basketball, football, cricket and swimming squads. He speaks English predominately but is also able to communicate in Vietnamese and French. When younger he also possessed the ability to communicate in Chinese and Italian.

Although brought up as a Buddhist and studying at Catholic/Christian schools all his life as well as studying scripture during his final year of high school he considers himself an atheist.

At this moment in time, he works part–time as a (commercial) researcher/developer on Linux related projects with his current focus being on software distribution mechanisms.

Two of his technical documents have been incorporated into the Linux Documentation Project ("Linux Dictionary" and "Linux Filesystem Hierarchy", www.tldp.org/guides.html). Furthermore, they are being used as reference books in at least ten universities around the world (University of Southern Queensland (Australia), Universidad Michoacana (Mexico), Hong Kong Polytechnic University (Hong Kong), Universidade de Sao Paolo (Brazil), University of Southern California (United States of America), University of Wales Swansea (United Kingdom), University of Ulster (Ireland), Universität Duisburg–Essen (Germany), Universidad Rey Juan Carlos (Spain), Insituto Superior Miguel Torga (Portugal), and Universiti Sains Malaysia (Malaysia)). As well as this, he is also a Development Lead and Project Administrator of the "Computer Dictionary Project" http://computerdictionary.tsf.org.za/dictionary/index.html which is being supported by the Shuttleworth Foundation.

He has published articles at linux.com, linmagau.org, desktoplinux.com, newsforge.com, linuxtoday.com, linux.org, pclinuxonline.com, tuxreports.com, etc.... and has a keen interest in the latest innovations in science and technology.

The author's homepage is at, http://geocities.com/linuxfilesystem/

Appendix D. Contributors

The author would like to thank a group of members (who wish to remain anonymous) from MLUG http://www.mlug.org.au for having the time and patience to help proof read this document before going to publication.

- Peter Yellman.
- Clyde Forrester.
- Geoff Farrell.
- Nick Spence.
- Christopher Priest.

Appendix E. Disclaimer

No liability for the contents of this document can be accepted. Use the concepts, examples and other content at your own risk. As this is a new edition of this document, there may be errors and inaccuracies that may of course be damaging to your system. Proceed with caution, and although this is highly unlikely, the author does not and can not take any responsibility for any damage to your system that may occur as a direct or indirect result of information that is contained within this document.

Naming of particular products or brands should not be seen as endorsements. You are strongly recommended to make a backup of your system before major installation and adhere to the practice of backing up at regular intervals.

Appendix F. Donations

If you would like to make a donations towards this project please use Paypal, https://www.paypal.com.

Appendix G. Feedback

Further revisions of this document will be dependent upon user response. Any feedback on the content of this document is welcome. Every attempt has been made to ensure that the instructions and information herein are accurate and reliable. Send comments, corrections, suggestions and questions to the author Binh Nguyen, linuxfilesystem(at)yahoo(dot)com(dot)au with the subject heading of LHFS. There is however, no guarantee of response.

All trademarks and copyrights are the property of their owners, unless otherwise indicated. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

The author would appreciate and consider it courteous, notification of any and all modifications, translations, and printed versions.

Appendix H. GNU Free Documentation License

Version 1.2, November 2002

H.1. 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.

H.2. 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.

H.3. 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.

H.4. 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.

H.5. 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.

H.6. 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".

H.7. 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.

H.8. 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.

H.9. 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.

H.10. 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.

H.11. 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.

H.12. 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.

Notes

[1] It requires several seconds of hard thinking on the users' behalf. Furthermore **sudo** can be configured to only allow users to execute certain commands. See the sudo(8), sudoers(5), and visudo(8) manual pages.