The Linux System Administrators' Guide

Version 0.6.2

Lar

The Linux System Administrators' Guide: Version 0.6.2
by Lars Wirzenius

by Joanna

Table of Contents

Dedication

This place is dedicated to a future dedication.

Source and pre-formatted versions available

Chapter 1. Introduction

> “In the beginning, the file was without form, and void; and emptiness was upon the face of the bits. And the Fingers of the Author moved upon the face of the keyboard. And the Author said, Let there be words, and there were words.”

This manual, the Linux System Administrators’ Guide, describes the system administration aspects of using Linux.

as such and can be skipped if you use a preconfigured system. Reading everything will, naturally, increase your understanding of the system and should make using and administering it more pleasant.

Like allringandvthe5 663.454815724wasbe aring4-257.327498724basis:(as)-241.634994434Iingand39.65.4796766749

Chapter

Chapter 2. Overview of a Lin6ux99 0 9.87n6 God.44 0

Figure 2-1 sho

Logins from terminals (via serial lines) and the console (when not running X) are provided by the **getty** program. **init** starts a separate instance of **getty** for each terminal for which logins are to be allowed. **getty** reads the username and runs the **login** program, which reads the password. If the username and password are correct, **login** runs the shell. When the shell terminates, i.e., the user logs out, or when **login** terminated because the username and password didn't match, **init** notices this and starts a new instance of **getty**. The

Mail

Electronic mail is usually the most important method for communicating via computer. An electronic letter is stored in a file using a special format, and special mail programs are used to send and read the letters.

Each user has an *incoming mailbox* (a file in the special format), where all new mail is stored. When someone sends mail, the mail program locates the receiver's mailbox and appends the letter to the mailbox file. If the receiver's mailbox is in another machine, the letter is sent to the other machine, which delivers it to the mailbox as it best sees fit.

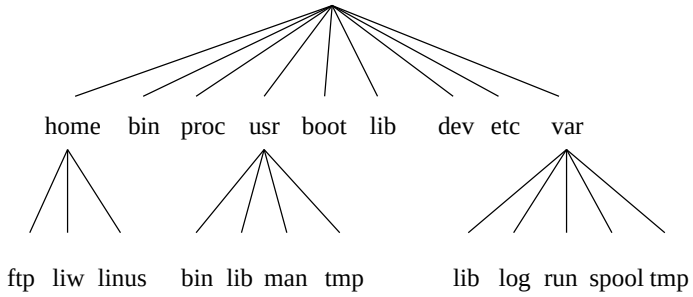
specia699 0 Td(),)Tj 8.2r(whn626

The mail system consists of many programs. The delivery of mail to local or remote mailboxes is done by one program (the *mail transfer agent* or *MTA*, e.g., **sendmail** or **smail**), while the programs

Notes

1. In fact, it is often mistakenly considered to be the operating system itself, but it is not. An operating system provides man2. .96s3T59 31.4399 [illegible] Tj 17.1ia7w.2.

Figure 3-1. Parts of a Unix directory tree. Dashed lines indicate partition limits.



`/home`, e.g., `/home/students` and `/home/staff`.

Although the different parts have been called filesystems above, there is no requirement that they actually be on separate filesystems. They could easily be kept in a single one if the system is a small single-user system and the

Configuration files specific to the machine.

`/root`

The home directory for user root.

`/lib`

Shared libraries needed by the programs on the root filesystem.

`/lib/modules`

Loadable kernel modules, especially those that are needed to boot the system when recovering from disasters (e.g., network and filesystem drivers).

`/dev`

Device files.

`/tmp`

Temporary files. Programs running after bootup should use `/var/tmp`, not `/tmp`, since the

`/proc`, `/usr`, `/var`, `/home`

Mount points for the other filesystems.

The `/etc` directory

The `/etc` directory contains a lot of files. Some of them are described below. For others, you should determine which program they belong to and read the manual page for that program. Many networking

filew

Similar to `/etc/passwd`, but describes groups instead of users. See the **group** manual page for more

Configuration file for the **login** command.

`/etc/printcap`

Like `/etc/termcap`, but intended for printers. Different syntax.

`/etc/profile`, `/etc/csh.login`, `/etc/csh.cshrc`

Files executed at login or startup time by the Bourne or C shells. These allow the system administrator to set global defaults for all users. See the manual pages for the respective shells.

`/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/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 `/dev` directory contains the special device files for all the devices. The device files are named using special conventions; these are described in the *Device list* (see XXX). The device files are created during installation, and later with the **/dev/MAKEDEV** script. The **/dev/MAKEDEV.local** is a script written by the system administrator that creates local-only device files or links (i.e., those that are not part of the standard **MAKEDEV**, such as device files for some non-standard device driver).

The `/usr` filesystem

The `/usr` filesystem is often lar

Manual pages, GNU Info documents, and miscellaneous other documentation files,

`/var/local`

Variable data for programs that are installed in `/usr/local` (i.e., programs that have been installed by the system administrator). Note that even locally installed programs should use the other `/var` directories if they are appropriate, e.g., `/var/lock`.

`/var/lock`

Lock files. Many programs follow a convention to create a lock file in `/var/lock` to indicate that

The `/proc` filesystem

The `/proc` filesystem contains a illusionary filesystem. It does not exist on a disk. Instead, the kernel creates it in memory. It is used to provide information about the system (originally about processes, hence the name). Some of the more important files and directories are explained below. The `/proc` filesystem is described in more detail in the `proc` manual page.

`/proc/stat`

Various statistics about the

Chapter 4. Using Disks and Other Storage Media

> “On a clear disk you can seek forever. ”

When you install or upgrade your system, you need to do a fair amount of work on your disks. You have to make filesystems on your disks so that files can be stored on them and reserve space for the different parts of your system.

This chapter explains all these ind(your)Tj012.4799 0 T,(e)Tj 1.79399 0 Td(pace)Tj124.3599 0 Td(you)Tj 11.4 0 Tgethe

Figure 4-1. A schematicOther91j 23.7599 0 Td(e)Tj 6.8399ofTd(4-0r)Tj 25.799Td(Mend)

Each surface is divided into tracks (and sectors) in the Other

Chapter

costs and large storage capacities.

Formatting

Formatting is the process of writing marks on the magnetic media that are used to mark tracks and sectors. Before a disk is formatted, its magnetic surface is a complete mess of magnetic signals. When it is formatted, some order is brought into the chaos by essentially drawing lines where the tracks go, and where they are divided into me di

Note that if you want to use an autodetecting device (e.g., `/dev/fd0`), you *must* set the parameters of the device with **setfdprm** first. To achieve the same effect as above, one would have to do the following:

```
$ setfdprm /dev/fd0 1440/1440
$ fdformat /dev/fd0
Double-sidedT rst.80 tracksT rst3.1227733201 (18)-599.3122463521 (sec/track. rst2.4350196722 (Tota
```

disk’s manual, if you’re curious if it is happening. Even such disks can fail, if the number of bad blocks grows too

```
   Device Boot  Begin   Start     End  Blocks   Id  System
/dev/hda1           1       1      24   10231+  82  Linux swap
/dev/hda2          25      25      48   10260   83  Linux native
/dev/hda3          49      49     408  153900   83  Linux native
/dev/hda4         409     409     790  163305    5  Extended
/dev/hda5         409     409     744  143611+  83  Linux native
/dev/hda6         745     745     790   19636+  83  Linux native
$
```

Extended and logical partitions

The original partitioning scheme for PC hard disks allowed only four partitions. This quickly turned out to be too little in real life, partly because some people want more than four operating systems (Linux, MS-DOS,

Figure

1	DOS 12-bit FAT	51	Novell?	a5	BSD/386
2	XENIX root	52	Microport	b7	BSDI fs
3	XENIX usr	63	GNU HURD	b8	BSDI swap
4	DOS 16-bitf <32M	64	Novell	c7	Syrinx
5	Extended	75	PC/IX	db	CP/M
6	DOS 16-bit >=32M	80	Old MINIX	e1	DOS access
7	OS/2 HPFS	81	Linux/MINIX	e3	DOS R/O

result in an problems,

Chapter

A modified version of the minix filesystem that lifts the limits on the filenames and filesystem sizes, but does not otherwise introduce new features. It is not very popular, but is reported to work very well.

ext2

The most featureful of the native Linux filesystems, currently also the most popular one. It is designed to be easily

Read the initial bad block list from the name file.

`/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,

Mounting and unmounting requires super

mounting an MS-DOS filesystem gives everyone

Checking for disk errors with badblocks

It can be a good idea to periodically check for bad blocks. This is done with the **badblocks** command. It outputs a list of the numbers of all bad blocks it can find. This list can be fed to **fsck** to be recorded in the filesystem data structures so that the operating system won't try to use the bad blocks for storing data. The following example will show how this could be done.

```
$ badblocks
```

```
Inode count:              360
Block count:              1440
Reserved block count:     72
```

Chapter

by using NFS), thereby reducing the total disk space required by several tens or hundreds of megabytes times the number of machines.

The problem with having many partitions is that it splits the total amount of free disk space into many

Chapter

A swap file is an ordinary file; it is in no way special to the kernel. scondleingTj 23.2399 0 Td(5tha)Tj 717.399 0 Td(nma

Chapter 5.

- Add some security to the estimate in step 1. This is because estimates of program sizes will probably be wrong, because you'll probably forget some programs you want to run, and to make certain that you have some extra space just in case. A couple of megabytes should be fine. (It is better to allocate too much than too little

not need to worry about it.orr43

Chapter 6.

floppy, then, when the system is otherwise installed and working well, will install LILO and start booting from the hard disk.

After the Linux ker,l,

- After this, the kernel starts the program **init** (located in `/sbin`) that starts

Chapter 6.

Chapter 7. init

> “Uuno on numero yksi” (Slogan for a series of Finnish movies.)

This chapter describes the **init** process, which is the first

Configuring init to start getty: the `/etc/inittab` file

When it starts up, **init** reads the `/etc/inittab` configuration file. While the system is running, it will re-read it, if sent the HUP signal; [2] this feature makes it unnecessary 02.24 the system 0 Tc 0 Tj 23.5199 0 Td (the)Tj /R16 configuration

Booting in single user mode

An important run level is *single user mode*

Note that the only new process is the one created by **init** (using the `fork` system call); **getty** and **login** only replace the program running in the process (using the `exec` system call).

A separate program, for noticing the user, is needed for serial lines, since it can be (and traditionally was) complicated to notice when a terminal becomes active. **getty** also adapts to the speed and other settings of the connection, which is important especially for dial-in connections, where these parameters may change from call to call.

There are several versions of **getty** and **init** in use, all with their good and bad points. It is a good idea to learn about the versions on your system, and also about the other versions (you could use the Linux Software Map to search them). If you don’t have dial-in’s, you probably don’t have to worry about **getty**, but **init** is still important.

Logins via the network

Two computers in the same netw

Chapter

Chapter 9. Managing user accounts

“The similarities of sysadmins and drug dealers: both measure stuff in

Initial

first), you need to create an account called ftp. In such cases, it is usually not necessary to set

Chapter 10. Backups

> Hardware is indeterministically reliable.
> Software is deterministically unreliable.
> People are indeterministically unreliable.
> Nature is deterministically reliable.

This chapter explains about why, how, and when to make backups, and how

dump is different in that it reads the filesystem directly and not via the filesystem. It is also written specifically for backups; **tar** and **cpio** are really for

```
usr/src/linux-1.2.10-includes/include/asm-m68k/
usr/src/linux-1.2.10-includes/include/asm-sparc/
usr/src/patch-1.2.11.gz
#
```

Unfortunately, **tar** can’t notice when a file’s inode information has changed, for example, that it’s permission bits hav

Chapter 11. Keeping Time

“Time

That time is Sunday,

If you need to adjust the hardware clock, it is usually simplest to reboot, go

Glossary (DRAFT)

“The Librarian of the Unseen Univ

Part