2

NAVIGATION

The first thing we need to learn (besides just typing) is how to navigate the filesystem on our Linux system. In this chapter we will introduce the following commands:

- `pwd`—Print name of current working directory.
- `cd`—Change directory.
- `ls`—List directory contents.

Understanding the Filesystem Tree

Like Windows, a Unix-like operating system such as Linux organizes its files in what is called a *hierarchical directory structure*. This means that they are organized in a tree-like pattern of directories (sometimes called folders in other systems), which may contain files and other directories. The first directory in the filesystem is called the *root directory*. The root directory contains files and subdirectories, which contain more files and subdirectories, and so on.

Note that unlike Windows, which has a separate filesystem tree for each storage device, Unix-like systems such as Linux always have a single filesystem tree, regardless of how many drives or storage devices are attached to the computer. Storage devices are attached (or more correctly, *mounted*) at various points on the tree according to the whims of the *system administrator*, the person (or persons) responsible for the maintenance of the system.

The Current Working Directory

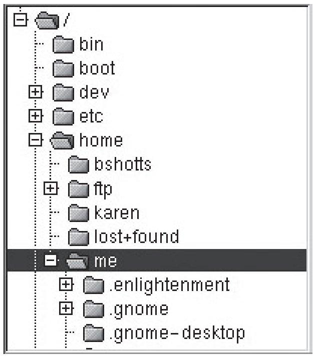

Figure 2-1: Filesystem tree as shown by a graphical file manager

Most of us are probably familiar with a graphical file manager, which represents the filesystem tree, as in Figure 2-1. Notice that the tree is usually shown upended, that is, with the root at the top and the various branches descending below.

However, the command line has no pictures, so to navigate the filesystem tree, we need to think of it in a different way.

Imagine that the filesystem is a maze shaped like an upside-down tree and we are able to stand in the middle of it. At any given time, we are inside a single directory and we can see the files contained in the directory and the pathway to the directory above us (called the *parent directory*) and any subdirectories below us. The directory we are standing in is called the *current working directory*. To display the current working directory, we use the `pwd` (print working directory) command:

```
[me@linuxbox ~]$ pwd
/home/me
```

When we first log in to our system (or start a terminal emulator session), our current working directory is set to our *home directory*. Each user account is given its own home directory, which is the only place the user is allowed to write files when operating as a regular user.

Listing the Contents of a Directory

To list the files and directories in the current working directory, we use the `ls` command:

```
[me@linuxbox ~]$ ls
Desktop  Documents  Music  Pictures  Public  Templates  Videos
```

Actually, we can use the `ls` command to list the contents of any directory, not just the current working directory, and it can do many other fun things as well. We'll spend more time with `ls` in Chapter 3.

Changing the Current Working Directory

To change your working directory (where we are standing in our tree-shaped maze) we use the `cd` command: Type **`cd`** followed by the pathname of the desired working directory. A *pathname* is the route we take along the branches of the tree to get to the directory we want. Pathnames can be specified in one of two ways, as absolute pathnames or as relative pathnames. Let's deal with absolute pathnames first.

Absolute Pathnames

An *absolute pathname* begins with the root directory and follows the tree branch by branch until the path to the desired directory or file is completed. For example, there is a directory on your system in which most of your system's programs are installed. The pathname of that directory is */usr/bin*. This means from the root directory (represented by the leading slash in the pathname) there is a directory called *usr* that contains a directory called *bin*.

```
[me@linuxbox ~]$ cd /usr/bin
[me@linuxbox bin]$ pwd
/usr/bin
[me@linuxbox bin]$ ls

...Listing of many, many files ...
```

Now we can see that we have changed the current working directory to */usr/bin* and that it is full of files. Notice how the shell prompt has changed? As a convenience, it is usually set up to automatically display the name of the working directory.

Relative Pathnames

Where an absolute pathname starts from the root directory and leads to its destination, a *relative pathname* starts from the working directory. To do this, it uses a couple of special symbols to represent relative positions in the filesystem tree. These special symbols are **.** (dot) and **..** (dot dot).

The **.** symbol refers to the working directory and the **..** symbol refers to the working directory's parent directory. Here is how it works. Let's change the working directory to */usr/bin* again:

```
[me@linuxbox ~]$ cd /usr/bin
[me@linuxbox bin]$ pwd
/usr/bin
```

Okay, now let's say that we wanted to change the working directory to the parent of */usr/bin*, which is */usr.* We could do that two different ways, either with an absolute pathname:

```
[me@linuxbox bin]$ cd /usr
[me@linuxbox usr]$ pwd
/usr
```

or with a relative pathname:

```
[me@linuxbox bin]$ cd ..
[me@linuxbox usr]$ pwd
/usr
```

Two different methods produce identical results. Which one should we use? The one that requires the least typing!

Likewise, we can change the working directory from */usr* to */usr/bin* in two different ways, either by using an absolute pathname:

```
[me@linuxbox usr]$ cd /usr/bin
[me@linuxbox bin]$ pwd
/usr/bin
```

or with a relative pathname:

```
[me@linuxbox usr]$ cd ./bin
[me@linuxbox bin]$ pwd
/usr/bin
```

Now, there is something important that I must point out here. In almost all cases, you can omit the `./` because it is implied. Typing

```
[me@linuxbox usr]$ cd bin
```

does the same thing. In general, if you do not specify a pathname to something, the working directory will be assumed.

Some Helpful Shortcuts

In Table 2-1 we see some useful ways the current working directory can be quickly changed.

Table 2-1: cd Shortcuts

Shortcut	Result
`cd`	Changes the working directory to your home directory.
`cd -`	Changes the working directory to the previous working directory.
`cd ~`*`username`*	Changes the working directory to the home directory of *`username`*. For example, `cd ~bob` changes the directory to the home directory of user *bob*.